297-8403-901

DMS-100 Family

Operator Services System Advanced Intelligent Network (OSSAIN)

User's Guide

SN09 and up Standard 13.01 July 2005

NORTEL NETWORKS™

DMS-100 Family

OSSAIN

User's Guide

Publication number: 297-8403-901
Product release: SN09 and up
Document release: Standard 13.01
Date: July 2005

Printed in the United States of America

Publication history

May 2005

Version 13.01 Standard release for TOPS22 (SN09), which adds the following OSSAIN functions:

- changes to table OAFUNDEF
  - new subfields USESERV and DARECALL
  - changes to subfield AUTOSYS
- ICMP echo replaced by OAP Node Connectivity Test messaging
- new log OAIN 306
- changed log OAIN 301
-

March 2005

Version 12.01 Standard release for TOPS21 (SN08), which adds the following OSSAIN functions:

- Table HOBICDEV removed for TOPS EOL program.

July 2004

Version 11.01 Standard release for TOPS20 (SN07), which adds the following OSSAIN functions:

- short message service (SMS)
- wireless RLT for DACC
- DAS-only function
- enhancements to simultaneous interactions
  - alternate hosting for passive operators
  - speech path change requests for passive operator connection
  - passive operator messaging optimization
- authorization codes

- extended account codes
- use of XA-Core Ethernet interface as an alternative to EIU for increased messaging capacity
- improved functionality for draining OSAC session pools
- OAP-level node connectivity test
- Session pool audits never more frequent than 60 seconds regardless of datafill for node audit frequency

This release also adds clarification of the following existing functionality:

- pre-SN07 functionality of DRAIN command for OSAC session pools
- implications of AABS datafill in OSSAIN tables (since AABS is no longer supported)

Finally, this release removes some obsolete material from Appendix A, moves some information from Appendix A to the Engineering chapter, and adds messaging capacity information to the Engineering chapter.

December 2003

Version 10.03 Standard maintenance release for TOPS15, which adds the following OSSAIN functions:

- adds clarification on datafilling table IPHOSTS when using an EIU

July 2002

Version 10.02 Standard maintenance release for TOPS15, which adds the following OSSAIN functions:

- adds a reason code to log OAIN606

March 2001

Version 10.01 Standard release for TOPS15, which adds the following OSSAIN functions:

- changes to option code references based on SOC consolidation
  - new SOC OSAN0101 includes former SOC codes OSAN0001, OSAN0002, OSAN0003, and OSAN004
  - new SOC includes former SOC codes OSAN0005, OSAN0006, and OSAN0007
  - new SOC OSAN0103 includes former SOC code OSAN0100
- new SOC-enabling table TOPSFTR

August 2000

Version 09.01 Standard release for TOPS14, which adds the following OSSAIN functions:

- removed the Display and Clear commands from OSSANCI tool
- OM group TOPPACT4
- new parameter SEND_BILLSPID_W_CLASSCHG
- deleted logs OAIN202 and OAIN204
- log OAIN615 no longer generated

Note: As of this release, all designations are labeled as TOPSxx instead of a LETxxxx. Previous references remain as their former LETxxxx designations.

March 2000

Version 08.01 Standard release for LET0013, which adds the following OSSAIN functions:

- OSSAIN support for DA automation
- log OAIN210
- changes to existing OSAN0007 SOC ordering code
- changes to existing OM OAPCALP9

August 1999

Version 07.01 Standard release for LET0012, which adds the following OSSAIN functions:

- STR digit detection
- Special Location Routing Number
- OSAN0007 SOC ordering code
- OM group OAPCP10
- log TOPS610
- enhancements to existing OSSAIN functions:
  - single method of assigning a trigger profile index
  - call timing
  - attributes for trunk group type TOPSVL
- new chapter covering OSSAIN enhancements
- OSSAIN QMS MIS information moved to enhancements chapter
- changes to fields in these tables:
  - OAINCTLA
  - CT4QNAMS
  - OATLKPRF
  - OAVLMAP

- OAFUNDEF
- OAINPARM

- changes to existing log OAIN204

May 1999

Version 06.02 Standard release for LET0011, clarified the OSSAIN call timing function.

March 1999

Version 06.01 Standard release for LET0011, which adds the following OSSAIN functions:

- OSSAIN alternate routing
- Commercial Credit Card Report
- automatic carrier selection
- estimate of charges
- new tables OAINRTE and SNVLGRP
- OSAN0006 SOC ordering code
- OM group OAPCALP9
- logs TOPS132, OAIN208, OAIN209, and OAIN625
- feature interactions with toll-free services
- enhancements to existing OSSAIN functions:
  - voice connections
  - QMS CT4Q refinements
  - disposition routing
  - conferencing
- changes to fields in these tables:
  - OAFUNDEF
  - OAFNDISP
  - OACNNPRF
  - OATLKPRF
  - OADTFPRF
  - OACAUPRF
  - OAINPARM
- changes to existing logs
- changes to SESNPOOLDIR DRAIN command

- datafill examples for provisioning Billing and Access Services software at the TOPS switch
- datafill examples for implementing DA Sequence Calling service
- information on TOPS Call Detail Recording (TDR) AMA billing format
- new arrangement of chapters to align with the seven Service Implementation Guide (SIG) parts:
    - Introduction
    - Functional description
    - Interactions
    - Planning and engineering
    - Provisioning
    - Billing
    - Operation, administration, and maintenance (OA&M)

August 1998

Version 05.01 Standard release for LET0010, which adds the following OSSAIN functions:

- OSSAIN Queue Management System Management Information System (QMS MIS) application
- OSSAIN country direct
- OSAN0005 SOC ordering code
- logs QMIS101 and OAIN624
- OM groups OAPNMIS and QMSMIS
- changes to interactions with release link trunking
- changes to fields in tables OAINPARM and QMSMIS
- changes to existing logs and OMs

April 1998

Version 04.01 Standard release for LET0009, which adds the following OSSAIN functions:

- voice link broadcasting (VLB)
- OSSAIN conferencing
- triggering directly to an operator
- changing the base service of a call
- new SOC ordering codes
- changes to fields in tables OANODINV, OAVLMAP, and OAFUNDEF
- streamlined description of OSSAIN maintenance

July 1997

Version 03.01 Standard release for LET0007, which adds the following OSSAIN functions:

- centralized OSSAIN (OSAC)
- simultaneous connection of two nodes to a call
- OSSAIN preprocessing
- enhancements to maintenance user interface
- new tables OANODNAM, OAINPRE, and OSCVLGRP
- ENSV0020 SOC ordering code
- changes to fields in tables OASESNPL and OATLKPRF
- changes to node types supported in table OANODINV
- changes to existing logs, OMs, and data schema tables
- feature interactions with local number portability (LNP), release link trunking (RLT), and service provider identifier (SPID)

November 1996

Version 02.02 Standard release for LETB006 with updated technical content

September 1996

Version 02.01 Standard release for LETB006

May 1996

Version 01.03 Preliminary release with updated technical content

February 1996

Version 01.02 Preliminary release

Contents

About this document 25
Chapters in this document 25
Related OSSAIN software and documentation 27
Related OSSAIN training 27
References in this document 28

Part 1: Introduction 31

Chapter 1: OSSAIN product overview 33
System architecture 34
Standalone OSSAIN 34
OSAC 36
Types of OSSAIN nodes 38
OSAC node type 38
OSNM node type 39
OSN node type 39
OSSAIN node datafill 39
OSSAIN base call processing overview 41
OSSAIN call flow concepts 42
Function 42
Control list 43
Session 43
Transition 43
Recall 44
Standalone OSSAIN call flow scenario 44
OSSAIN information road map 46

Part 2: Functional description 47

Chapter 2: OSSAIN software functionality 49
Call processing 49
Call model 49
OSSAIN call model terminology 50
OSSAIN base call processing architecture 55
Routing and queuing 55
Routing and queuing components 56
Overview 56
Routing and queuing configurations 57
Functions and session pools 61
Routing and queuing table relationships 65

Datafilling functions and session pools 66
Datafilling the QMS setup for routing and queuing 69
Voice link connections between the switch and SN 75
SN originated calls 78
Trigger processing 79
Call model 79
TDPs and trigger events 81
Initial call setup TDP and datafill 82
Call floated TDP and datafill 85
Trigger table relationships 97
Trigger call flow scenarios 99
Float processing with TOPS operator and automated systems 101
Trigger processing parameters 102
Notification timer 102
Transitions 103
Direct transfer to control list 103
Call flow 103
Function provider selection 105
Transitions between functions 111
OAP overview 118
OAP definition 118
Functional characteristics of OAP 123
Sanity timers 124
Session begin timer operation 125
On-hook sanity timer operation 125
Call sanity timer operation 127
Queued sanity timer operation 128
Hierarchy of OSSAIN sanity timers 128

Chapter 3: OSAC call processing 129
OSAC call distribution 131
Types of sessions 131
OSAC call distribution flow 131
OSAC call distribution datafill 134
SN originations to the OSAC remote switch 137
Releasing an SN session 137
Trigger event inform 137
Session throttling 138
Error handling 138
OSAC voice connections 139
OSAC voice connection flow 139
Voice connections with a centralized SN 139
OSAC voice link datafill 140
Releasing voice connections 141
Two-way voice link maintenance 141
Auto voice link selection and connection capability 142
Error handling 142
OSAC parallel datafill requirements 143
Table OANODNAM 143
Table OANODINV 145
Table OASESNPL 148
Table OAFUNDEF 150
Table OAVLMAP 151

Chapter 4: OSSAIN enhancements 153
OSSAIN preprocessing 154
Simultaneous connections 156
Sample interactions with two service providers 157
Agents in a simultaneous connection 157
Pass-through messaging 158
Simultaneous connection flow 159
Operator handoff 160
ALT_BILL_HANDOFF_METHOD parameter 161
VSNOPT parameters 161
Service provider identifier (SPID) 161
Obtaining the SPID through OAP 162
TOPS Local Number Portability (LNP) 163
Release link trunking (RLT) 163
RLT phased deployment 163
Eligibility checks 164
Types of RLT 165
ISUP messages supporting RLT 165
Call timing 167
GEN AMA with called party connected 167
Example OAP message flow 167
Outgoing ISUP 169
Sample scenario 169
Voice link broadcasting (VLB) 170
Example VLB scenarios 170
Datafilling broadcast voice links 173
Maintenance of broadcast voice links 175
Provisioning broadcast voice links 175
OSSAIN conferencing 176
Method 1 176
Method 2 177
Four-way conference 181
Country direct 185
TOPS country direct 185
OSSAIN country direct 187
OSSAIN QMS MIS 189
MIS node connectivity 189
OSSAIN QMS MIS application 191
MIS node maintenance 194
OSSAIN alternate routing 196
Two methods of alternate routing 196
Using a routing index—OAINRTE 197
Clearing and replacing RIs and RNs 198
Additional QMS CT4Q refinements 199
OSSAIN calls routed to operators—OAFUNDEF 199
Auto voice link selection and connection 201
Connecting the voice link—OAFUNDEF 202
Selecting the voice link—SNVLGRP 202
Commercial credit card sales report 203
Authorization Codes 203
Automatic carrier selection 204
Estimate of charges 204

Global support 204
Called DNSCRN screening 205
Special location routing number (SLRN) 205
  Optional answer on voice link connection capability 205
OSSAIN support for DA automation 206
Short Message Service 206

Part 3: Interactions 207

Chapter 5: OSSAIN interactions 209

Additional QMS CT4Q refinements 210
  CT4QAUTO refinement 210
  ASST refinement 210
  Invalid CT4Q refinement 210
Auto voice link selection and connection 211
  OSAC 211
  Simultaneous connections 211
  OSSAIN preprocessing 211
  SN initiated calls and recalls 212
  Trigger events 212
  Broadcast voice links 212
Branding 212
Call timing 212
Changing the base service 213
Country direct 213
Equal access 214
  COMFGD and EAOSS signaling types 215
Global support 215
OSSAIN alternate routing 215
  Equal access 216
  Local number portability (LNP) 216
  TOPS translations group (XLAGRP) scheme 216
OSSAIN conferencing 217
OSSAIN preprocessing 218
OSSAIN QMS MIS 219
OSSAIN support for DA automation 220
  Initial call routing 220
  Call arrival tone and caller join tone 220
  Call context block 220
  Text to operator text 221
  BILTYPES field in table ADACCOPT 221
OSSAIN support for toll-free calls 221
  Enhanced 800 features 221
  800 Plus features 222
  800 AMA 223
Outgoing ISUP 224
Release link trunking 224
  OSSAIN call processing 224
  OSSAIN trigger support 225
  Three-way calling 225
Routing and queuing 225
Simultaneous connections 226
Special location routing number 226

Transitions 226
SN transfer to DA operator 227
SN transfer to operator position 227
SN transfer to AABS 227
Operator transfer to control list 228
Operator transfer to SN 229
DAS transfer to SN 229
Blocking of OPP requests 229
Triggers 229
Call float interactions with an SN 230
Call float interactions with an operator 230
Triggering directly to an operator 230
Call float restrictions 231
Voice link broadcasting 232

Part 4: Planning and engineering 235

Chapter 6: OSSAIN engineering 237
OSSAIN system limitations 237
Hardware resources 238
Conference three ports 238
Specialized tone receivers 240
DTMF receivers 240
Voice links 240
Table OFCENG parameters 241
OSSAIN_NUM_RU 241
OSAC_NUM_RU 243
CRS_SUBRU_POOL*_SIZE parameters 245
CRS_SUBRU_POOL1_SIZE 245
CRS_SUBRU_POOL2_SIZE 245
CRS_SUBRU_POOL4_SIZE 246
Memory 247
IPL data store impact 247
Maximum datafill data store impact 247
Messaging and data network—introduction 248
EIU engineering and provisioning 249
IP address restrictions 250
Routing restrictions 250
Capacity restrictions 251
Example IP address plans using EIUs 252
Datafilling EIUs 256
Table interdependency and autoconfiguration 263
XA-Core Ethernet interface engineering and provisioning 264
IP address restrictions 265
Routing restrictions 265
Capacity restrictions 266
Security restrictions 266
Datafilling XA-Core Ethernet interfaces 267
Determining OSSAIN messaging bandwidth requirements 272
OAP messaging 272
OSAC messaging 272

Part 5: Provisioning 275

Chapter 7: OSSAIN data schema 277
OSSAIN datafill descriptions 277
Alphabetical reference for OSSAIN table descriptions 278
OSSAIN base datafill 279
QAPLNDEF 281
QMSCQDEF 281
OANODNAM 282
OANODINV 284
OQCQPROF 292
OASESNPL 295
OAFUNDEF 301
OAFNDISP 310
OSAC parallel datafill 315
Tables requiring parallel datafill 315
OSSAIN voice connectivity datafill 316
OAVLMAP 318
OSCVLGRP 322
SNVLGRP 323
Direct transfers and new subscriber originations datafill 325
OACTLDEF 326
CT4QNAMS 327
OAINCTLA 329
Function blocking datafill 329
OAFUNBLK 330
TOPSDB 331
Trigger datafill 332
OADTFPRF 334
OACAUPRF 336
OACNNPRF 338
OADSCPRF 341
OATLKPRF 344
OATPRFIX 348
OAINCTLA 349
OSSAIN preprocessing datafill 350
CT4QNAMS 350
OAINPRE 351
OSSAIN alternate routing datafill 352
OFRT 353
OAINRTE 354
OSSAIN QMS MIS application datafill 355
QMSMIS 355
OSSAIN parameter datafill 357
OAINPARM parameters 357
TOPS parameter datafill 361
TOPSPARM parameters 361

Chapter 8: OSSAIN software optionality control 363
OSAN functional group 363
OSAN0101: OSSAIN Base 364
OSAN0102: OSSAIN Enhancements 364
OSAN0103: OSSAIN session pricing 366
OSAN SOC report example 368

OSSAIN SOC MAP CI commands 368
SELECT command 368
OSSAIN SOC session limiting 369
Error message for exceeding the SOC LIMIT 369
Error message for not enabling OSAN0101 370
Table TOPSFTR 370

Part 6: Billing 373

Chapter 9: OSSAIN billing 375
Custom AMA 375
Comparison of BAF and TDR 376
OSSAIN AMA recording 379
BAF AMA records 379
TDR AMA records 388
TOPS Charge Adjust service 392
NO AMA 392

Part 7: OA&M 393

Chapter 10: OSSAIN maintenance 395
OSSAIN node maintenance 395
Defining nodes in table OANODINV 395
Automated audit process 396
Automated RTS process 402
State transitions 402
Unsolicited busy request from an OSNM node 404
Centralized SNs in an OSAC network 405
Related logs 405
Related OM groups 406
Protocols 406
User interface—OAINPMDIR directory 407
Session pool maintenance 424
Defining session pools in table OASESNPL 424
Automated audit process 424
State transitions 427
Unsolicited busy request from a session pool 428
Throttle request from a session pool 429
Related logs 430
Related OM groups 431
Protocol 431
User interface—SESNPOOLDIR directory 432
Additional utilities for OSSAIN 444
OSSAINCI directory 444
QCALL directory 450
QVIEW directory 450
OSSAIN alarms 452
OSNM node and session pool alarms 452
OSSAIN node state transition alarms 452

Chapter 11: OSSAIN logs 453
OAIN logs 453

OAIN200 455
OAIN201 456
OAIN203 456
OAIN205 456
OAIN206 457
OAIN207 457
OAIN208 458
OAIN209 458
OAIN210 458
OAIN300 459
OAIN301 459
OAIN302 460
OAIN303 461
OAIN304 461
OAIN305 462
OAIN306 462
OAIN500 463
OAIN502 464
OAIN503 464
OAIN504 465
OAIN505 465
OAIN506 465
OAIN507 466
OAIN600 466
OAIN601 467
OAIN602 467
OAIN603 468
OAIN604 468
OAIN605 469
OAIN606 469
OAIN607 473
OAIN608 475
OAIN609 476
OAIN610 477
OAIN611 477
OAIN612 478
OAIN613 479
OAIN614 479
OAIN615 480
OAIN616 480
OAIN617 481
OAIN618 482
OAIN619 483
OAIN620 484
OAIN621 485
OAIN622 486
OAIN623 487
OAIN624 488
OAIN625 488
OAIN700 489
OAIN701 489
OAP logs 490
OAP600 490

OAP601 492
OAP602 493
OAP603 494
OSAC logs 496
OSAC200 497
OSAC201 497
OSAC202 498
OSAC203 498
OSAC204 498
OSAC300 499
OSAC600 499
OSAC601 500
EXT logs 501
EXT106 501
EXT107 502
EXT108 502
AUD logs 503
AUD607 503
AUD609 504
AUD629 504
PM logs 505
PM100 505
PM102 506
PM103 506
PM104 507
PM105 507
PM106 507
PM128 508
DAS logs 508
DAS105 508
DAS106 509
TCCI logs 510
TCCI100 510
TCCI101 511
TCCI102 511
QMIS logs 512
QMIS101 512
TOPS logs 513
TOPS129 513
TOPS132 514
TOPS610 515

Chapter 12: OSSAIN operational measurements 517
AABSHAND 519
Associated OM groups 520
CTRYDIR 520
Associated OM groups 521
DAMISC 521
Associated OM groups 522
EXT 522
Associated OM groups 522
OADATCOM 522
Associated OM groups 524

OAFLTRIG 524
Associated OM groups 525
OAINNODE 525
Associated OM groups 526
OAINQMS 526
Associated OM groups 528
OAINRTE 528
Associated OM groups 529
OANODEDC 529
Associated OM groups 531
OAPCALP1 531
Associated OM groups 533
OAPCALP2 533
Associated OM groups 535
OAPCALP3 535
Associated OM groups 537
OAPCALP4 537
Associated OM groups 539
OAPCALP5 539
Associated OM groups 541
OAPCALP6 541
Associated OM groups 543
OAPCALP7 543
Associated OM groups 545
OAPCALP8 545
Associated OM groups 547
OAPCALP9 547
Associated OM groups 549
OAPCP10 549
Associated OM groups 551
OAPMERRN 551
Associated OM groups 552
OAPMERRS 552
Associated OM groups 554
OAPMTYPN 554
Associated OM groups 556
OAPMTYPS 556
Associated OM groups 557
OAPNMIS 558
Associated OM groups 558
OAPNMTC 558
Associated OM groups 561
OAPSPMTC 561
Associated OM groups 563
OASNPLDC 563
Associated OM groups 565
OASNPOOL 565
Associated OM groups 565
OASVNDCP 566
Associated OM groups 566
OSACCP1 566
Associated OM groups 568
OSACCP2 568

Associated OM groups 570
OSACND 570
Associated OM groups 571
OSACSP 572
Associated OM groups 573
OSNND 574
Associated OM groups 575
OSNSP 575
Associated OM groups 576
QMSMIS 576
Associated OM groups 577
TOPPACT3 578
Associated OM groups 578
TOPPACT4 579
Associated OM groups 579
TOPPDID5 579
Associated OM groups 580
TOPSRLT2 581
Associated OM groups 582

Appendixes 583

Appendix A: OSSAIN data communications 585
Additional documentation 585
OSSAIN data network 586
Data communications software architecture and protocols 586
Application layer—OAP and OSAC protocols 587
Transport layer—UDP and TCP protocols 588
Network / internet layer—IP protocol 589
Network access layer 589
IP addressing 589
IP address classes 589
Subnets and masks 590
Data port numbers 592
OSSAIN datafill for data communications 593

Appendix B: Advanced OSAC configuration 597
Advanced OSAC network example 597
Datafill examples 598

Appendix C: Provisioning broadcast voice links in a JNET office 605
JNET terminology 605
Network module A-side and B-side 607
Time switching 607
Channel mapping in a broadcast network connection 609
Network blocking 611
Effect of broadcast voice links on NM capacity 611
Port deloading 612
Special provisioning cases 612
Determining the number of ports to deload 612
Provisioning rules 614

PM port and channel mapping 615
DTC/LTC 615
DCM 620
TM8 620
Provisioning examples 620
Example 1 620
Example 2 621
Example 3 621

Appendix D: OSSAIN datafill example for Billing and Access Services 623
Example standalone OSSAIN configuration 623
Example datafill for Billing and Access Services 625
QAPLNDEF 626
CT4QNAMS 626
TQCQINFO 627
QMSCQDEF 628
TQMSFCQA 629
OANODNAM 629
OANODINV 629
OQCQPROF 630
OASESNPL 630
OAFUNDEF 630
OACTLDEF 631
OADTFPRF 632
OACNNPRF 632
OADSCPRF 632
OATPRFIX 633
OAINCTLA 633
TRKGRP 634
OAVLMAP 634
OAFNDISP 635
OAINPARM 636
OFCENG 636
AABSOST 636
Example OSAC configuration 637
Example OSAC datafill for Billing and Access Services 638
OANODNAM 638
OANODINV 639
OASESNPL 639
OAFUNDEF 640
OAVLMAP 641
OSCVLGRP 642
OFCENG 642
Billing and Access Services calling scenarios 643
0- BAS call in the standalone/host 643
0- BAS call and handoff to BAC node in the standalone/host 643
Operator handoff to BAC node in the standalone/host 644
0- BAS call in the OSAC remote 644
0+ DA BAS call 646

Appendix E: DA Sequence Calling 647
Understanding how DA Sequence Calling works 647

Network diagram 647
Call processing 648
Datafilling DA Sequence Calling 648
Assumption 648
Datafill sequence 649
Step 1—CT4QNAMS 649
Step 2—QMSTOPS 650
Step 3—OAFUNDEF 650
Step 4—OACTLDEF 650
Step 5—OADTFPRF 651
Step 6—OACNNPRF 651
Step 7—OATLKPRF 651
Step 8—OADSCPRF 652
Step 9—OATPRFIX 652
Step 10—OAINCTLA 652
Step 11—OAINPARM 652
Sample DA Sequence Calling scenario 653
Initial DA call 653
Subsequent DA call 653
AMA records 654
OMs 656
OAFLTRIG 656
Other considerations 657
Hardware requirements 657
Software requirements 658
Interactions 658
Restrictions 659
Datafilling other DTMF digits as triggers 660
Additional CT4Q refinements for operator calls 660

List of terms 661

Index 683

About this document

The *Operator Services System Advanced Intelligent Network (OSSAIN) User's Guide* accompanies the OSSAIN software product. The guide describes OSSAIN functionalities and how they work together to deliver services. It provides the user with an overview of the product, a detailed description of the software, and supplementary information on datafill and maintenance activities.

This guide is intended for users who are familiar with DMS Traffic Operator Position System (TOPS) processing and features, including Queue Management System (QMS).

Chapters in this document

Following is a summary of each chapter.

Chapter 1: OSSAIN product overview

This chapter includes an overview of the system architecture and an introduction to OSSAIN call processing.

Chapter 2: OSSAIN software functionality

This chapter describes in detail the call processing software of OSSAIN.

Chapter 3: OSAC call processing

This chapter describes in detail the call processing software for centralized OSSAIN (OSAC).

Chapter 4: OSSAIN enhancements

This chapter describes the enhancements to the OSSAIN software.

Chapter 5: OSSAIN interactions

This chapter describes the interactions and restrictions of OSSAIN software on features in the TOPS network.

Chapter 6: OSSAIN engineering

This chapter includes information on engineering OSSAIN.

Chapter 7: OSSAIN data schema

This chapter includes information on how to datafill OSSAIN tables. It discusses each table, its fields and values, and gives example datafill.

Chapter 8: OSSAIN software optionality control

This chapter describes how OSSAIN implements software optionality control (SOC).

Chapter 9: OSSAIN billing

This chapter describes billing and AMA records for OSSAIN.

Chapter 10: OSSAIN maintenance

This chapter describes the maintenance user interface for OSSAIN.

Chapter 11: OSSAIN logs

This chapter describes logs for OSSAIN.

Chapter 12: OSSAIN operational measurements

This chapter describes operational measurements (OM) for OSSAIN.

Appendix A: OSSAIN data communications

Appendix A discusses data communications software in the OSSAIN network.

Appendix B: Advanced OSAC configuration

Appendix B shows an advanced OSAC configuration.

Appendix C: Provisioning broadcast voice links in a JNET office

Appendix C describes the provisioning requirements of broadcast voice links in a junctored network (JNET) office.

Appendix D: OSSAIN datafill example for Billing and Access Services

Appendix D shows the TOPS switch datafill required to provision Billing and Access Services.

Appendix E: DA Sequence Calling

Appendix E discusses the TOPS switch datafill for the DA Sequence Calling service.

List of terms

This chapter lists OSSAIN terms and definitions.

Related OSSAIN software and documentation

The Open Automated Protocol (OAP) interface and the OSSAIN Application Programmer's Interface (API) are integral parts of the OSSAIN software product suite. They are described thoroughly in two separate documents.

OAP

OAP is the interface between a DMS TOPS switch and an external computer, the service node (SN). OAP gives control to the SN and allows it to direct the switch processing of calls. Many types of network-based services can be created for the SN which include, but are not limited to, applications that use operators or the operator services environment. These services can be delivered without corresponding DMS switch development.

The *OSSAIN Open Automated Protocol Specification* (Q235-1) document describes in detail the protocol; message and data formats; data block and field definitions; and call processing and non-call processing operations.

Note: The OSSAIN OAP is a licensed interface. To receive this document, please contact the Network Information Services (NIS) Marketing group of Nortel Networks.

OSSAIN API

The OSSAIN API software minimizes the effort to create applications that use OAP and reside on an SN. The API has the following characteristics that make it easy to use:

- It is portable to a broad range of computing platforms.
- It contains an object library that models key OSSAIN concepts.
- It provides full OAP support, including protocol encoding and decoding, encryption algorithms, and current versions of OAP.
- It provides basic OSSAIN maintenance, which can be easily extended for application-specific maintenance requirements.

The *OSSAIN API User's Guide* (Q260-1) provides an understanding of the OSSAIN API framework. It describes how to build and configure an application using the API. This book is intended for software developers and managers who plan to develop SN applications using the OSSAIN API.

Note: The OSSAIN API is a licensed interface. To receive this document, please contact Nortel Networks NIS Marketing.

Related OSSAIN training

Training for OSSAIN is available through Nortel Networks TOPS Professional Services.

References in this document

Following are the DMS-100 documents referenced in this book. The middle section of the document number is represented by *nnnn* because this number is determined by the PCL to which it belongs.

- Translations Guide, 297-*nnnn*-350
- Customer Data Schema Reference Manual, 297-*nnnn*-351
- Operational Measurements Reference Manual, 298-*nnnn*-814
- Log Report Reference Manual, 297-*nnnn*-840

Following are the other documents referred to in this book:

- *Nonmenu Commands Reference Manual*, 297-1001-820.
- *Bellcore Format Automatic Message Accounting Reference Guide*, 297-1001-830
- *DMS-100 E800 SSP Toll-free Numbers Service Guide*, 297-5151-021
- *DMS-100 800Plus and End-Office Display Service Guide*, 297-5151-022
- *North American DMS-100 AMA Bulletin*, 297-8021-830B
- *TOPS Local Number Portability User's Guide*, 297-8403-902
- *TOPS Unbundling User's Guide*, 297-8403-903
- *TOPS Call Detail Recording (TDR) User's Guide*, 297-8403-904
- *TOPS Translations and Screening User's Guide*, 297-8403-905
- *Provisioning Manual Active Detail*, PLN-8991-104
- *Software Optionality Control User's Manual*, 297-8991-901
- *OSSAIN API User's Guide*, Q260-1
- *OSSAIN Open Automated Protocol Specification*, Q235-1
- *TOPS Custom AMA*, Q259-1

  Note: To receive this document, please contact Nortel NIS Marketing.
- *Open Position Protocol Specification*, Q214-1

  Note: Open Position Protocol (OPP) is a licensed interface. To receive this document, please contact Nortel NIS Marketing.
- *Standard Nortel-DMS/DAS Protocol*, Q210-1

  Note: Standard Nortel-DMS/DAS Protocol is a licensed interface. To receive this document, please contact Nortel NIS Marketing.
- *TCP/IP Network Administration*, by Craig Hunt (O'Reilly and Associates, Inc. 1992)
- *Billing and Access Services Administration Guide*, 203-3261-301
- *Billing and Access Services Planning and Engineering Guide*, 203-3261-101

- *Ethernet Interface Unit User Guide*, 297-8991-910

Part 1: Introduction

Part 1: Introduction includes the following chapter:

Chapter 1: “OSSAIN product overview,” beginning on page 33.

Chapter 1: OSSAIN product overview

The Operator Services System Advanced Intelligent Network (OSSAIN) product provides an interface between a DMS TOPS switch and several external service nodes (SN). The interface allows an SN to control switch functionality associated with operator services.

OSSAIN software defines the following capabilities:

- how a switch and an SN communicate
- how a call transitions between SNs, between a TOPS operator and an SN, and between a TOPS automated system and an SN
- how operators and SNs work together to provide services

The OSSAIN product supplies the *framework* for SNs to deliver operator services. However, OSSAIN makes no assumptions about the types of services that use the interface. With OSSAIN, operating companies can deploy the services of an SN independently of DMS switch software.

This chapter gives an overview of the OSSAIN product, focusing on the following areas:

- the *system architecture*, which describes OSSAIN hardware components and illustrates two simple network topologies
- the *types of OSSAIN nodes*, which show the relationship between different nodes in OSSAIN networks
- an *overview of call processing*, which introduces concepts of OSSAIN processing and shows a sample call flow scenario

The last section provides a road map to the location in this book of detailed OSSAIN information.

System architecture

OSSAIN system architecture consists of two basic network configurations, as follows:

- standalone OSSAIN
- centralized OSSAIN, known as OSAC

This section provides an *overview* of each network configuration. Detailed information is in Chapter 2: “OSSAIN software functionality,” and in Chapter 3: “OSAC call processing.”

Note: Users need to understand the concepts of standalone OSSAIN call processing before they proceed to OSAC call processing.

Standalone OSSAIN

Standalone OSSAIN is a TOPS architecture in which call processing control is distributed among a single switch and one or more SNs. The switch is responsible for automatic call distribution, making voice connections, and maintaining each SN. The SNs rely on the switch for call processing.

Standalone OSSAIN consists of the following components:

- one TOPS switch
- one or more SNs
- Ethernet local area network (LAN) and optionally a wide area network (WAN)
- voice links

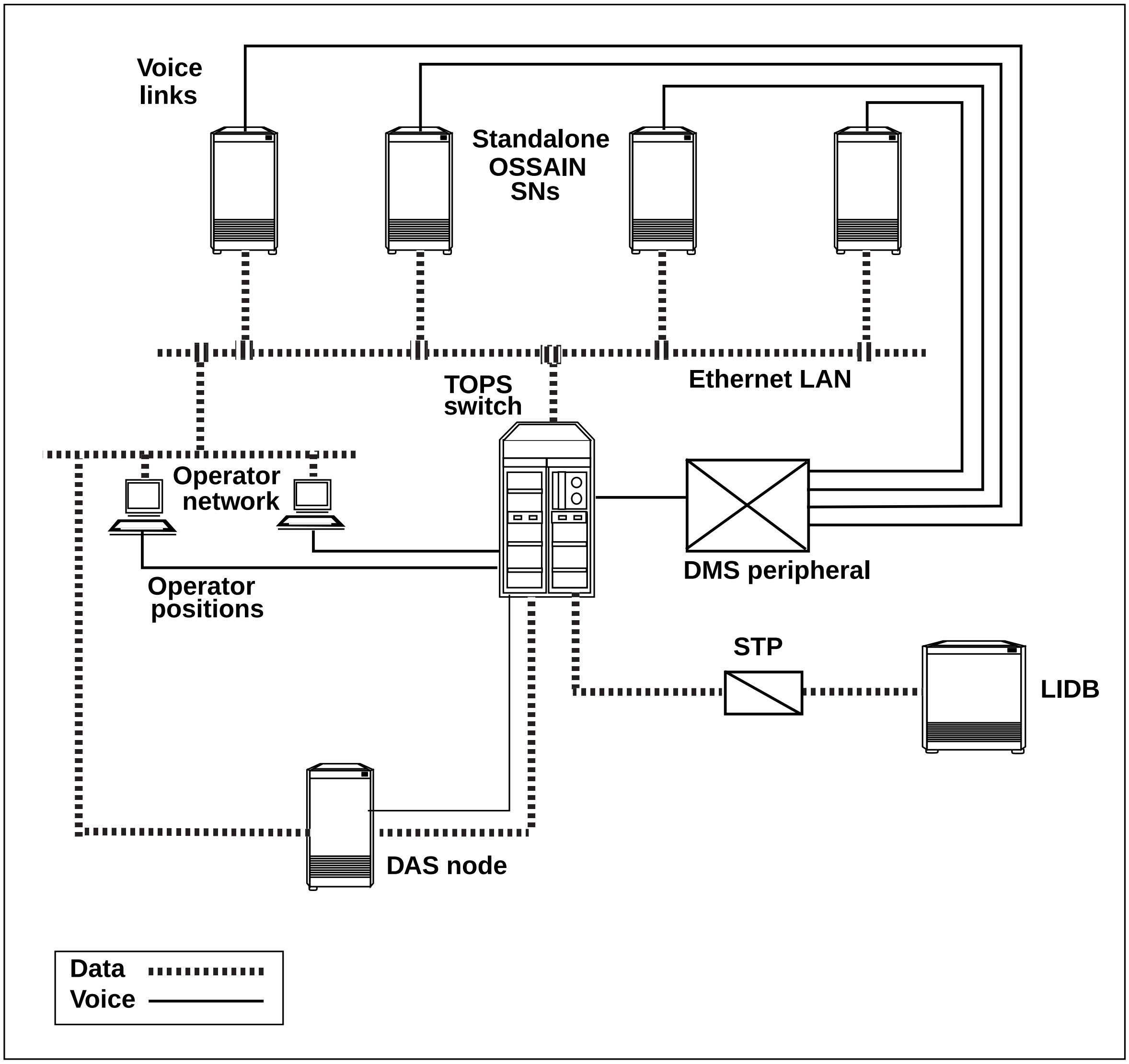
The following optional components comprise the Intelligent Services Environment (ISE):

- operator network
- operator positions
- directory assistance system (DAS) node
- signal transfer point (STP)
- line information database (LIDB)

Note: The location in the network of the ISE components is independent of the location of the standalone OSSAIN components.

The following figure gives a rough idea of a standalone OSSAIN network topology.

Figure 1 Standalone OSSAIN network topology



Data connections in standalone OSSAIN

The switch and SNs have data communication with each other over a LAN and possibly also a WAN. Each SN connects to the LAN using standard Ethernet technology. The switch connects to the LAN through either XA-Core Ethernet interface cards or an Ethernet interface unit (EIU). EIUs are provisioned on either a Link Peripheral Processor (LPP) or a Fiber Link Interface Shelf (FLIS). SNs can communicate with each other over the LAN.

Voice connections in standalone OSSAIN

DMS Series 1 or Series 2 peripherals provide the voice links between the switch and the SNs.

OSAC

OSAC is a TOPS architecture in which call processing control is distributed among more than one switch and several *centralized* SNs. OSAC allows the services offered by the centralized SNs to be shared by multiple switches.

The switches are set up in a host and remote configuration. The OSAC host switch is responsible for providing automatic call distribution, making voice connections, and maintaining the centralized SNs.

OSAC consists of the following components:

- TOPS OSAC host switch
- TOPS OSAC remote switch
- centralized SNs
- Ethernet local area network (LAN)
- wide area network (WAN)
- routers
- voice links

The following optional components comprise the ISE:

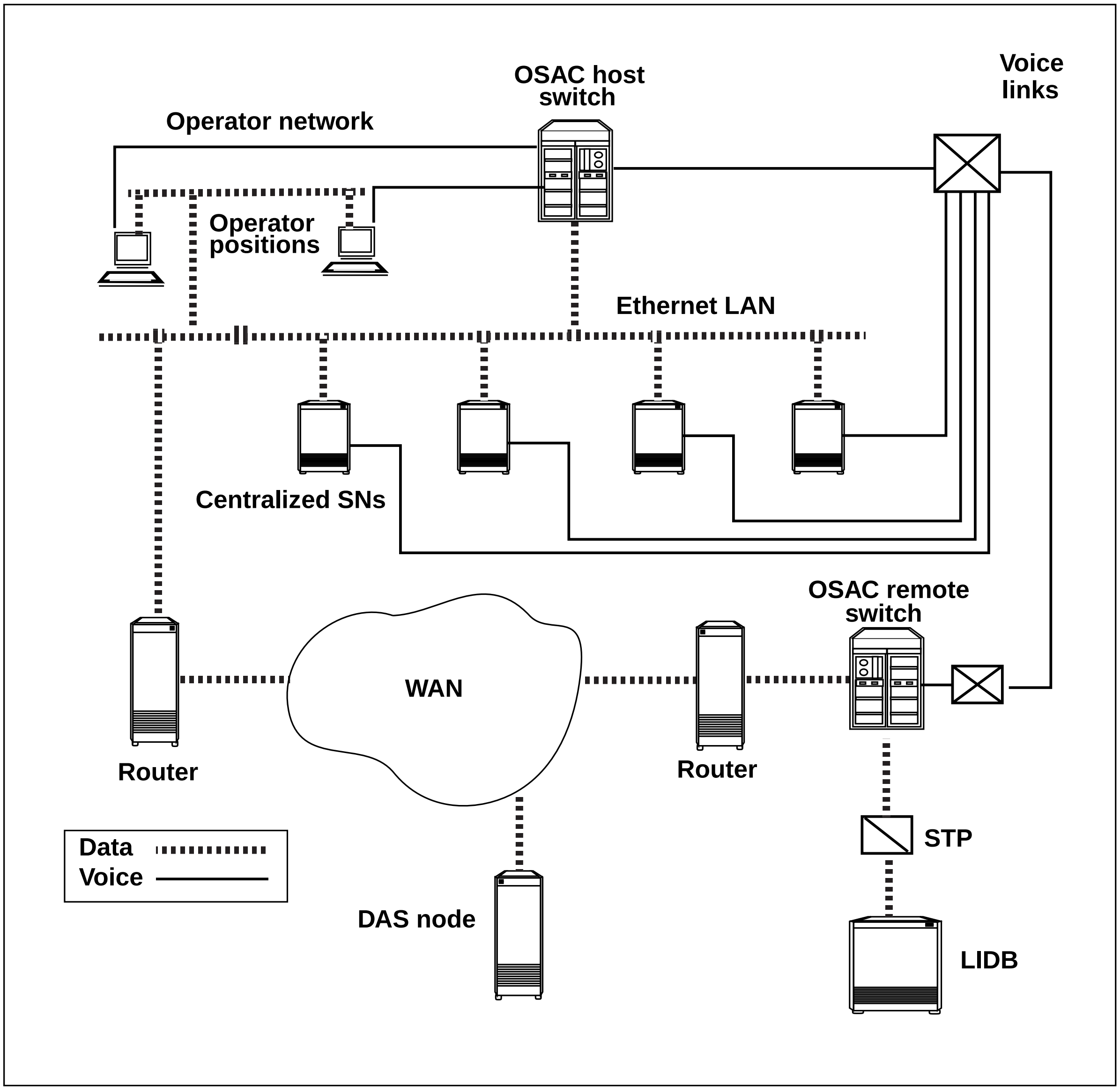
- operator network
- operator positions
- directory assistance system (DAS) node
- signal transfer point (STP)
- line information database (LIDB)

Note 1: The location in the network of the ISE components is independent of the location of the OSAC components.

Note 2: The location of hosts and remotes in the OSAC network is independent of their location in operator centralization (OC).

The following figure gives a rough idea of an OSAC network topology. (This figure is an over-simplification. The actual data network should be designed by a qualified network designer.)

Figure 2   OSAC network topology



Note: For information on call processing in the OSAC network, refer to Chapter 3: “OSAC call processing.” For details on an advanced OSAC configuration, refer to Appendix B: “Advanced OSAC configuration.”

Data connections in OSAC

The OSAC host switch and centralized SNs normally communicate with each other over an Ethernet LAN. The OSAC host switch connects to the LAN through either XA-Core Ethernet interface cards or an Ethernet interface unit (EIU), which is provisioned on either a Link Peripheral Processor (LPP) or a Fiber Link Interface Shelf (FLIS). Centralized SNs can communicate with each other over the LAN.

WAN technology is used to span longer distances between the OSAC host switch and the OSAC remote switch. The OSAC remote switch can exchange call processing messages with the centralized SNs over the WAN and the LAN.

Voice connections in OSAC

DMS Series 1 or Series 2 peripherals provide the voice links among the switches and centralized SNs. The OSAC host switch coordinates the voice connection between a centralized SN and the OSAC remote switch.

Types of OSSAIN nodes

The term *OSSAIN node* refers to both SNs and switches. The type of node is datafilled in DMS switch tables so that the switch knows how to interact with other OSSAIN nodes in the network.

There are three types of OSSAIN nodes, as follows:

- OSAC node
- OSNM (Operator Services Node Maintained) node
- OSN (Operator Services Node) node

This section describes the types of nodes and shows how they fit in to the two OSSAIN network configurations—standalone OSSAIN and OSAC.

OSAC node type

The OSAC node type applies *only to switches*. The OSAC node type is used in the OSAC network to identify the host switch and the remote switch. OSAC nodes communicate with each other over the data network to coordinate the use of services of centralized SNs. In the OSAC network, each switch is datafilled as an OSAC node.

For a call that resides in the OSAC remote switch, the OSAC nodes have the following tasks:

- The OSAC host switch provides call distribution to the centralized SN and makes the voice connection between the centralized SN and the OSAC remote switch.
- The OSAC remote switch performs call processing for the call at the centralized SN.

OSNM node type

The OSNM node type applies *only to SNs*. The OSNM node type is used for SNs in the standalone OSSAIN network and for centralized SNs in the OSAC network.

The OSNM node type describes the *maintenance relationship* the SN has to the switch where it is datafilled. In a standalone OSSAIN network, the switch is responsible for maintaining the SN. So at the standalone switch, the SN is datafilled as an OSNM node type.

In an OSAC network, SNs are centralized at an OSAC host switch, which is responsible for their maintenance. So at the OSAC host switch, each centralized SN is datafilled as an OSNM node type.

OSN node type

The OSN node type applies *only to centralized SNs*. The OSN node type is used when an OSAC switch needs to know about its connectivity to a centralized SN, but is not responsible for the maintenance of that SN. The OSN node type is used only in an OSAC remote switch.

In the OSAC network, an OSAC remote switch can use the services of an SN that is centralized at an OSAC host switch. The OSAC host switch maintains the centralized SN.

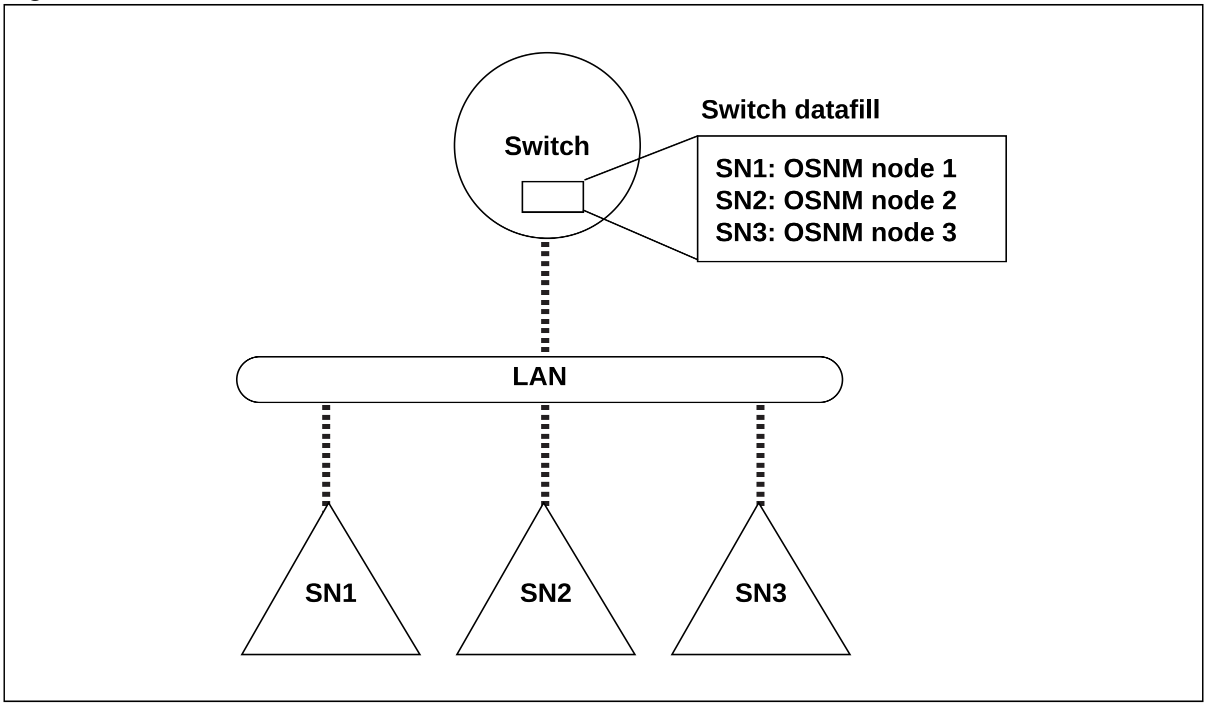
Therefore, in the OSAC network a centralized SN is datafilled in *both* of the following ways:

- At the OSAC remote switch, the centralized SN is datafilled as an OSN node type. The OSAC remote switch does not perform node maintenance on the OSN; however, the OSAC remote does monitor the data connection to the OSN and can perform a node connectivity test.
- At the OSAC host switch, the centralized SN is datafilled as an OSNM node type. The OSAC host switch performs node maintenance on the OSNM.

OSSAIN node datafill

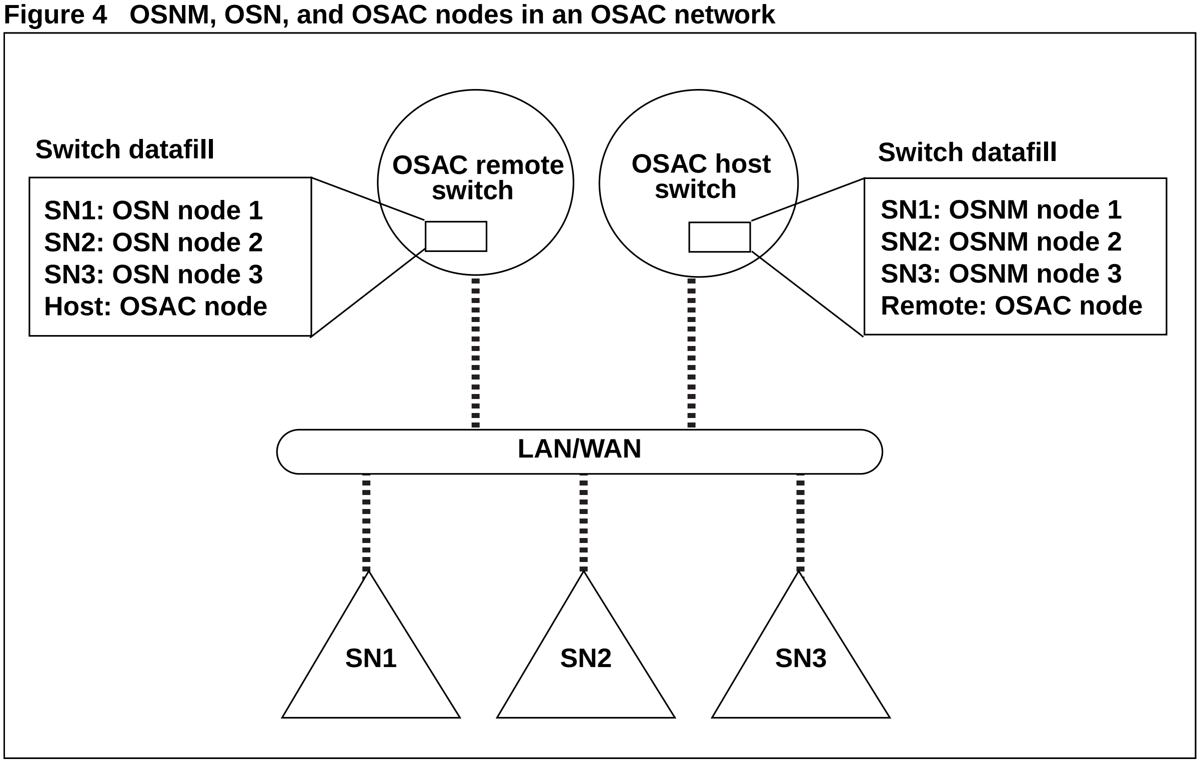
The following figure shows how three SNs and one switch are datafilled in a standalone OSSAIN network.

Figure 3 OSNM nodes in a standalone OSSAIN network



The following figure shows how three SNs and two switches are datafilled in an OSAC network.

Figure 4 OSNM, OSN, and OSAC nodes in an OSAC network



OSSAIN node type matrix

The following table shows the switch where each OSSAIN node type is datafilled.

Table 1 OSSAIN node datafill

Switch	OSNM	OSN	OSAC
Standalone switch	X		
OSAC host switch	X		X
OSAC remote switch		X	X
Note: For complete information on datafill for OSSAIN nodes, refer to Chapter 7: “OSSAIN data schema.”			

OSAC network tasks

The following table shows the tasks that each switch in an OSAC network performs.

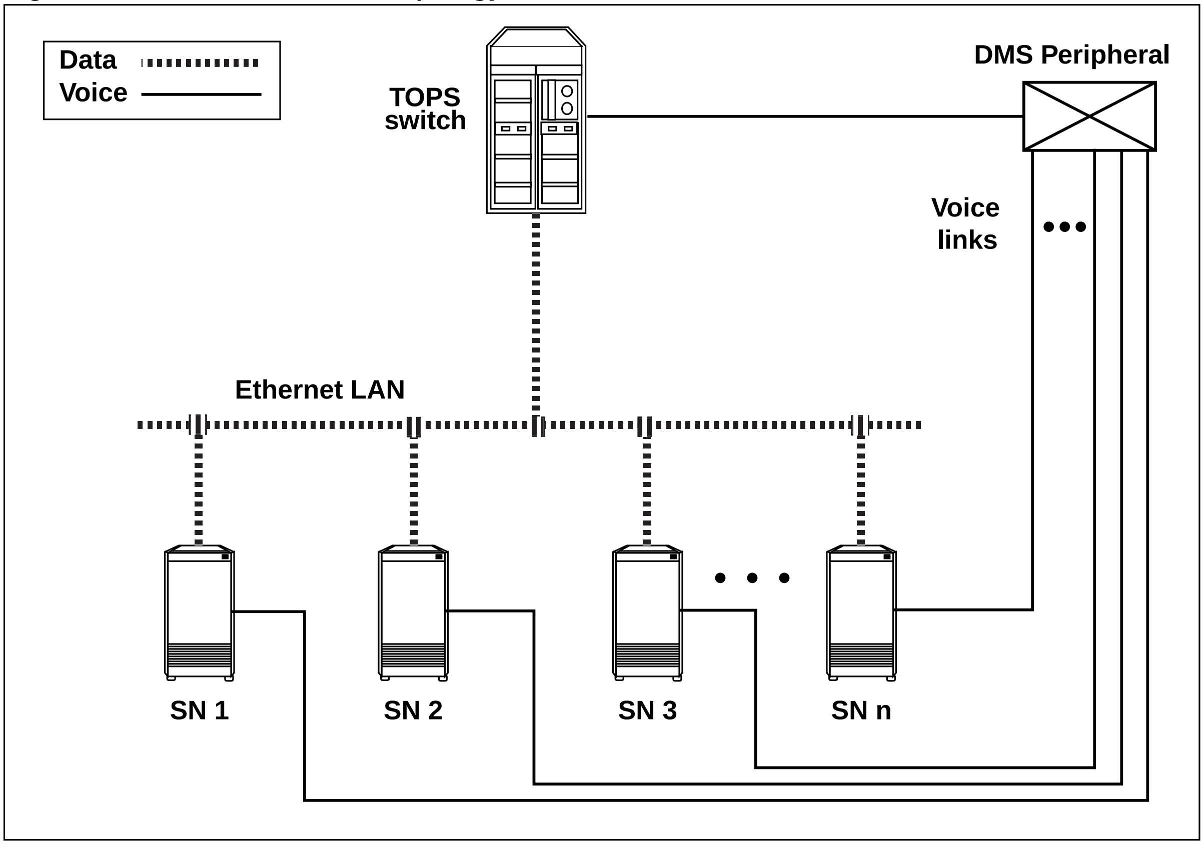
Table 2 OSAC network tasks

Task	OSAC host switch	OSAC remote switch
Call distribution	X	
Voice connections with SN	X	
Node maintenance (see Note)	X	
Call control		X
Note: For complete information on maintenance of OSSAIN nodes, refer to Chapter 10: “OSSAIN maintenance.”		

OSSAIN base call processing overview

In call processing, OSSAIN is able to interwork TOPS functionality with SN functionality. This section describes how OSSAIN performs this interworking. Refer to the following figure of the *standalone* OSSAIN topology for this discussion of OSSAIN call processing.

Figure 5 Standalone OSSAIN topology



Note: For information on centralized *OSSAIN call processing, refer to Chapter 3: "OSAC call processing."*

OSSAIN call flow concepts

OSSAIN introduces the following fundamental concepts related to call flow:

- function
- control list
- session
- transition
- recall

The following paragraphs briefly describe these concepts.

Function

A function is a service or portion of a service that is provided by an SN, an operator, or a TOPS automated system. Examples of functions are branding, alternate billing, and Yellow Pages service.

Control list

A control list is the mechanism used to interwork OSSAIN with TOPS. It contains the name of a function, such as branding, that OSSAIN applies to the call.

Note: OSSAIN software supports only one function in a control list.

Session

Equivalent to an operator, a session is an agent that serves call queues. Just as TOPS distributes calls to operators that serve call queues, OSSAIN distributes calls to sessions that serve call queues. So when a call is directed to an OSSAIN function, OSSAIN maps the function to a call queue and distributes the call to a session. If a session is not available, the call is queued until one becomes available.

Note 1: The OSSAIN product provides the ability to queue a call for an automated system; existing TOPS automated systems do not provide this ability.

Note 2: For details on how OSSAIN functions, sessions, and call queues work, refer to Chapter 2: “OSSAIN software functionality.”

Voice and data connectivity

A major difference between an operator and a session is in voice and data connectivity. The operator agent always has both voice and data connectivity. However, a session with an SN always has data connectivity, but does not have voice connectivity unless necessary.

Session pool

A session pool is a group of sessions that serves specific call queues. A session pool is associated with a particular SN.

Note: In the OSAC network, a session pool also is associated with a particular OSAC switch. For more information, refer to Chapter 3: “OSAC call processing.”

Transition

A TOPS operator can be specified as a function in an OSSAIN control list. In addition, an SN can transfer a call *directly* to another SN or to a TOPS operator for backup or for a complex service interaction. This direct transfer is called a transition. A transition bypasses processing of the control list, but it allows for greater control of the call by the SN.

Note: For details on how OSSAIN performs transitions among SNs, automated systems, and operators, refer to Chapter 2: “OSSAIN software functionality.”

Recall

OSSAIN call processing allows a recall back to an SN after the call has been floated. This ability is called call floated trigger processing. Based on the datafill for various events (such as busy, ringing, no answer, DTMF input, or disconnect), call floated trigger processing allows floated calls to transfer to a function or a control list. The event that causes such a recall is a *trigger*.

Note: For details on how call floated trigger processing works, refer to Chapter 2: “OSSAIN software functionality.”

Standalone OSSAIN call flow scenario

To illustrate the OSSAIN call flow concepts of functions, control lists, and sessions, consider the following scenario. In the scenario, a subscriber wishes to alternately bill a Yellow Pages call, as follows:

1 The subscriber dials 0+9991234.

2 A message welcomes the subscriber.

3 The subscriber is presented with alternate billing choices.

4 Billing is established and the subscriber is connected to the Yellow Pages service and presented with listings.

In this scenario, the *functions* are branding (message), alternate billing, and Yellow Pages. Assume that the functions have been defined in datafill as follows: SN_1 brands the call, SN_2 alternately bills the call, and SN_1 provides Yellow Pages service to the call.

The following paragraphs analyze each step in the scenario.

Step 1—distributing a call to a session

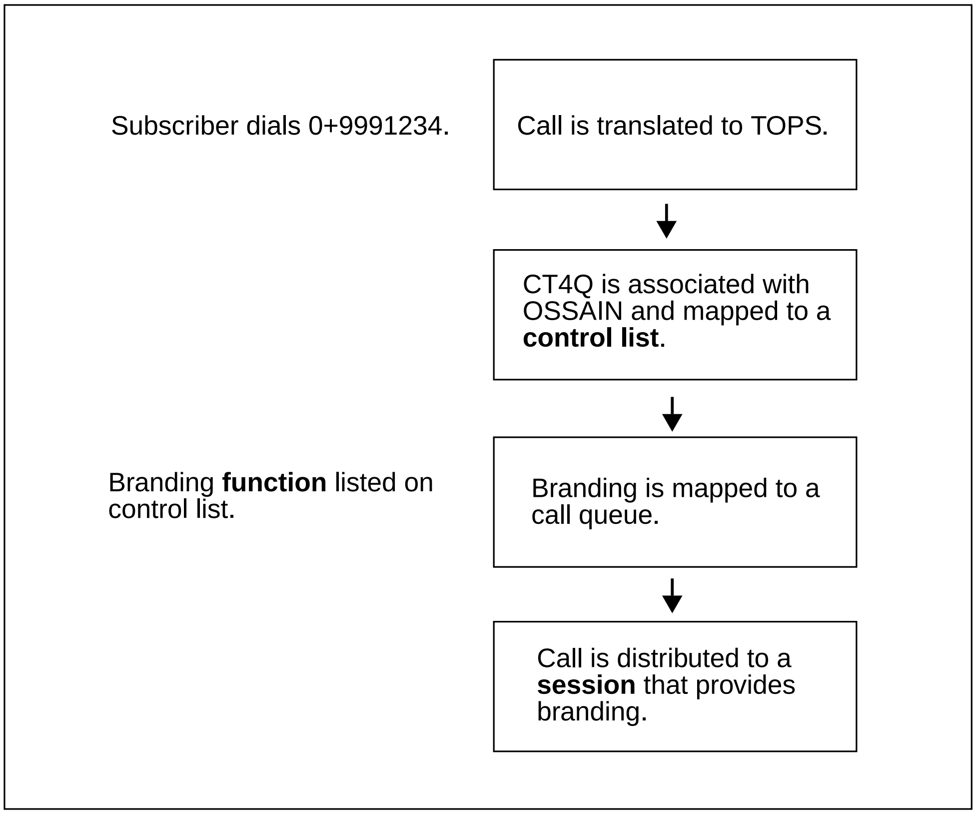
In the scenario, call flow starts where interoffice signaling completes and the call is translated to TOPS. The Queue Management System (QMS) refines the call, by using various call attributes, to a single call type for queuing (CT4Q).

At this point, the call typically is routed to an operator that serves the call queue. However, with OSSAIN, the CT4Q is evaluated to determine whether TOPS or OSSAIN call processing is in effect. In this case, the call origination type (0+) and the dialed digits (9991234) are used to route the call to a CT4Q for OSSAIN.

When a call is routed to OSSAIN, the final CT4Q obtained during QMS refinements is mapped to a *control list*. The control list contains the name of a function. Using datafill, OSSAIN determines which system provides the function—SN, operator, or TOPS automated system—and maps the function to a call queue.

The first function in this scenario is branding, and it is provided by SN_1. OSSAIN distributes the call to a *session* on SN_1 that serves the branding call queue. The following figure shows how OSSAIN processing handles the call flow in step 1 of the scenario.

Figure 6 Distributing a call to a session



Step 2—branding on SN_1

Once OSSAIN distributes the call to a session on SN_1, the SN requests a voice link between the call and the SN. SN_1 then brands the call and transfers the call to a control list.

Step 3—alternate billing on SN_2

The control list supplied by SN_1 determines that the next function to process is alternate billing at SN_2. OSSAIN processing transfers the call to a session on SN_2. SN_2 provides the subscriber with alternate billing choices and transfers the call to a control list.

Step 4—Yellow Pages on SN_1

The control list supplied by SN_2 determines that the next function is Yellow Pages service at SN_1. OSSAIN processing transfers the call to a session on SN_1. SN_1 presents the subscriber with Yellow Pages listings.

Note: For information on OSAC call flow, refer to Chapter 3: “OSAC call processing.”

OSSAIN information road map

The chapters that follow provide details on these user tasks:

- understanding how OSSAIN call processing works (Chapter 2)
- understanding how OSAC call processing works (Chapter 3)

  Note: Users need to understand the concepts of standalone OSSAIN call processing before they proceed to OSAC call processing.
- understanding the enhancements made to core OSSAIN functions (Chapter 4)
- understanding the interactions and restrictions of OSSAIN functions (Chapter 5)
- engineering the network (Chapter 6)
- datafilling OSSAIN switches (Chapter 7)
- using the software optionality control (SOC) tool (Chapter 8)
- understanding billing records (Chapter 9)
- maintaining nodes and session pools (Chapter 10)
- understanding log reports (Chapter 11)
- understanding operational measurements (Chapter 12)
- understanding the role of data communications in the network (Appendix A)
- datafilling an advanced OSAC network (Appendix B)
- provisioning broadcast voice links in a junctored network (Appendix C)
- datafilling Billing and Access Services at the TOPS switch (Appendix D)
- understanding how the DA Sequence Calling service works (Appendix E)

Part 2: Functional description

Part 2: Functional description includes the following chapters:

Chapter 2: “OSSAIN software functionality,” beginning on page 49.

Chapter 3: “OSAC call processing,” beginning on page 129.

Chapter 4: “OSSAIN enhancements,” beginning on page 153.

Chapter 2: OSSAIN software functionality

This chapter describes in detail how OSSAIN software functions across the network, focusing on the following areas:

- call processing (page 49)
- routing and queuing (page 55)
- trigger processing (page 79)
- transitions (page 103)
- Open Automated Protocol (OAP) overview (page 118)
- sanity timers (page 124)

Call processing

This section discusses the following switch-related components of OSSAIN call processing:

- OSSAIN call model

  The operator services call model guides the design of OSSAIN call processing and its interaction with operator services. A call model represents possible call flows.

- OSSAIN base call processing architecture

  The OSSAIN base call processing architecture defines a service-independent infrastructure for automating telecommunication services with operator assistance and backup.

Call model

Using a call model helps to understand at a high level which call flows are possible. In particular, Advanced Intelligent Network (AIN) call models help define which system components (switch or external computer) control the call flow at a given point in the call, and when the components should communicate.

In the TOPS environment, a call can consist of many interactions that involve connections to different entities such as an operator, an automated system, or a forward party. Each interaction can be billed independently. Some examples of interactions are call completion to a forward party, directory assistance service, directory assistance with call completion, and calling card sequence calls.

A calling card sequence call, for example, could involve two distinct completions. Each completion is to a different forward party and is billed separately. However, the entire interaction is considered a single call.

OSSAIN call model terminology

The following terms are used in the OSSAIN call model discussion in this section.

Point in call (PIC)

A PIC is a call state. In the call model, the call states are null, initial call setup, call control, and call floated.

Event

An event is a detectable condition that occurs during a call. Events include translations conditions, connection status information, and subscriber dialed input.

Trigger

A trigger is an event that causes the call to be redirected to an SN, operator, or automated system. An event that has no effect on normal call processing is *not* a trigger.

Triggering and trigger detection point (TDP)

Triggering is the switch process that determines the action for an event at a particular point in call processing. This point is referred to as a TDP.

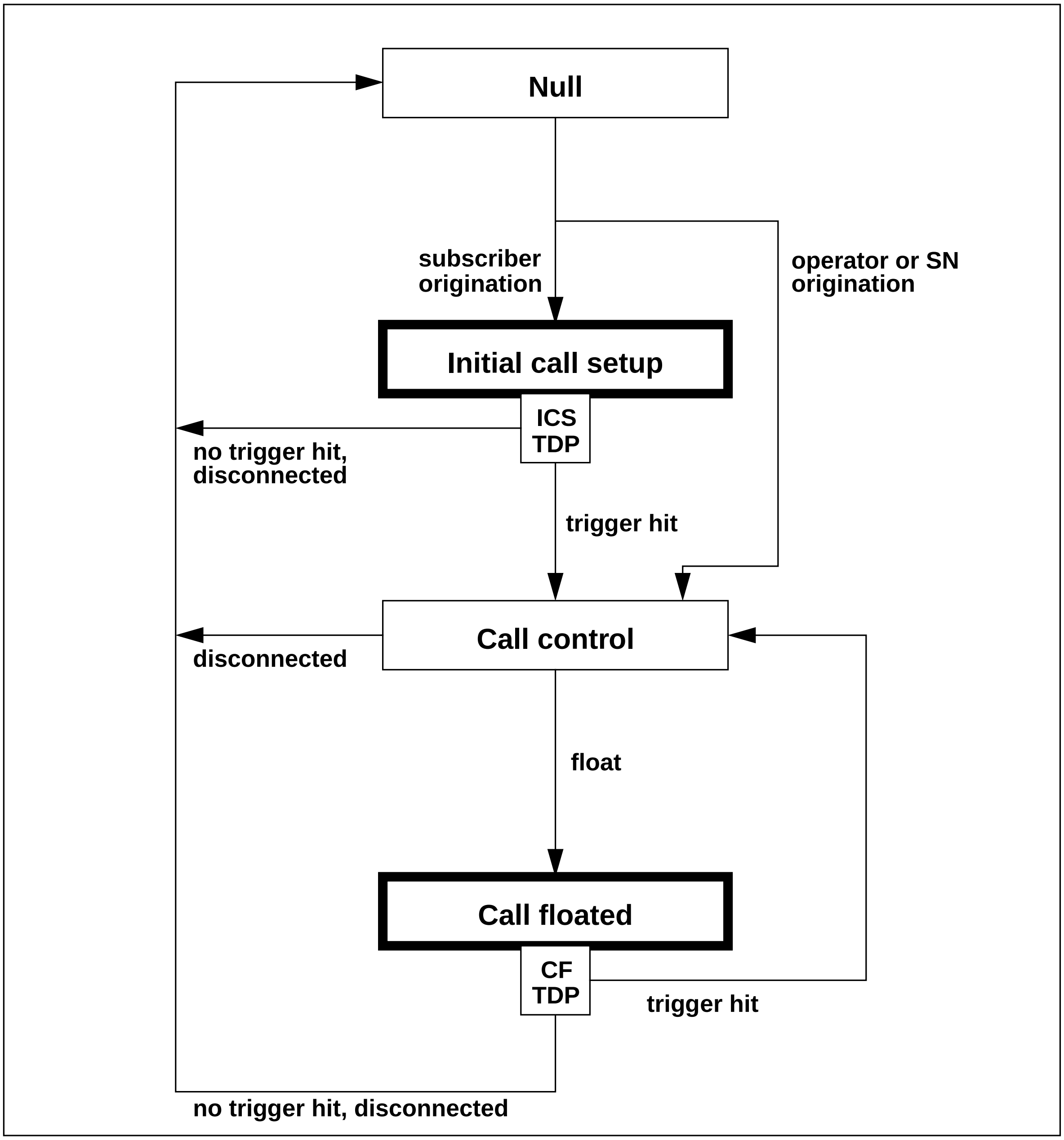
Two possible actions are associated with a TDP: continue with normal call processing (no trigger is hit) or bridge on an SN or operator (trigger is hit). Trigger processing associated with the PICs is fundamental to the OSSAIN call model. More details on triggers are provided in the section “Trigger processing” on page 79.

Transitions

A transition allows SNs to transfer the call to another SN, an operator, or a TOPS automated system. For example, OSSAIN allows an SN to transfer a call to an operator for operator backup. More details on transitions are provided in the section “Transitions” on page 103.

Refer to the following figure for an illustration of the OSSAIN call model.

Figure 7 OSSAIN call model



In the figure, the four large boxes denote the PICs, the two small boxes denote the TDPs, and the arrows denote the events.

The switch is solely in control of the call at the initial call setup and the call floated PICs. The model indicates the PICs at which an SN, an operator, or a TOPS automated system can be bridged into a call. During the initial call setup, calls can be directed to any of these systems.

Operator and SN originated calls are also shown in the model, as well as how an operator or SN can be triggered back into a call once it has been floated.

The following subsections describe in more detail each of the four PICs (null, initial call setup, call control, and call floated) and the events that cause it to be entered or exited. The TDPs are discussed in terms of their effect on the exit events associated with the PIC.

Null PIC

The null PIC is the state where no call is present.

- Entry events
  - — This PIC is entered upon initialization or when a call is disconnected.
- Action
  - — not applicable
- Exit events
  - — A subscriber originates a call that is routed to TOPS software. Subscriber initiated calls move immediately to the initial call setup PIC.
  - — SN or operator originated (for example, delay) calls move directly to the call control PIC.

Initial call setup PIC

The initial call setup PIC is the state entered by the switch to determine which system—SN, operator, or automated system—should control the call.

- Entry events
  - — This PIC is entered during call set up for subscriber originated calls.
- Action
  - — This PIC is responsible for routing the call to the appropriate system. And it is used to obtain an agent (or session) to control the call. If queuing for an available agent is necessary, the call remains in this PIC until the agent is obtained (for example, queued for operator).
- Exit events

  When any event occurs in this PIC, the call encounters the initial call setup (ICS) TDP processing. ICS TDP processing may result in one of the following transitions, which then exit ICS TDP processing.

  - — During ICS TDP processing, the following events cause the call to be disconnected (cleared or taken down):
    - – A disconnect event occurs but is not found to be a trigger.
    - – An unexpected event occurs (such as a loss of a voice connection due to network failure).

- The following events cause the call to transition to the call control PIC:
  - A session identifier to an SN is obtained.

    Note: A session identifier identifies a sequence of related message exchanges with the SN. A session identifier is in use starting with the initial message sent from the switch to the SN during a specific call and ending when the SN releases control back to the switch.
  - An operator is obtained.
  - Resources for an automated system are obtained.

Call control PIC

The call control PIC is entered to allow a service interaction to occur. An SN, operator, or automated system performs call control.

- Entry events

  The following events are associated with different types of subscriber initiated calls:

  - An SN session is obtained; this PIC is entered to present the call to the node.
  - An operator is obtained; this PIC is entered to present the call to the operator.
  - Resources for an automated system are obtained; this PIC is entered to initiate the automated service.

  The following events are associated with calls originated by either an operator or an SN:

  - An operator delay call is initiated by the operator.
  - An SN originates a call.
- Action

  Within this PIC, the switch performs requests on behalf of the controlling entity and informs the controlling entity of the results. Examples are as follows:

  - a party may be added to the call
  - billing information may be updated
  - a billing validation query may be performed
  - switch-based resources (for example, a digit receiver or a coin detection circuit) may be connected to the call
  - information may be appended to the AMA record
  - an AMA record may be generated

- the call may be transferred or floated
- another node or operator may be connected

- Exit events
  - When the controlling entity floats the call, the call transitions to the call floated PIC.
  - When the controlling entity transfers the call, the call transitions to the call control PIC.
  - When the controlling entity disconnects the call, the call transitions to the null PIC.

Call floated PIC

The call floated PIC is entered to allow the SN, operator, or automated system to release—float— from the call as well as have the switch monitor for events that result in triggering. (For simplicity, the call floated PIC is shown in the model as a single PIC.)

- Entry events
  - The controlling entity floats from the call. Floating occurs under one of three conditions: prior to connecting the parties; after connecting, but prior to answer; and after the parties are connected and talking.

  Note: The state of the connections is somewhat independent of the call states. For example, two parties are capable of talking while in the call control PIC *or* in the call floated PIC.

- Action

  This PIC has several responsibilities as follows:
  - It must establish a connection between the parties if they are not already connected.
  - It must determine which resources are necessary to detect potential triggers (for example, digit receivers and timers) as well as connect and start the resources.
  - It must detect any events that occur while floated. Note that CF TDP processing actually processes the events.

- Exit events

  When any event occurs in the call floated PIC, the call encounters call float (CF) TDP processing. CF TDP processing determines if an event is a trigger. If the event is a trigger, the associated action is taken (for example, transfer to a specified SN).

CF TDP processing may result in one of the following transitions, which then exit CF TDP processing:

- The following events cause the call to be disconnected (cleared or taken down):
  - A disconnect event occurs but is not found to be a trigger.
  - An unexpected event occurs, such as, loss of a voice connection due to network failure.
- Any event that results in a trigger being hit causes the call to transition to the call control PIC.
- Any event that does not result in triggering or normal disconnect causes the call to remain in the call floated PIC (for simplicity, these events are not indicated in the diagram).

OSSAIN base call processing architecture

The OSSAIN base call processing architecture is compatible with the DMS cross-threaded architecture. It is designed to closely mimic the TOPS implementation of DMS architecture to facilitate transitions between SNs and TOPS operators and automated systems.

Routing and queuing

OSSAIN routing and queuing is responsible for selecting an idle queuing agent to service a call once OSSAIN determines that the requested function is provided by an SN.

Note: An SN can handle a maximum number of simultaneous call sessions, with one queuing agent associated with each session. A queuing agent, therefore, represents the data connectivity between the switch and the SN. Voice connectivity can be done at the request of the node, independently of data connectivity.

The same OSSAIN function can be provided by several SNs, with calls distributed among them on a most idle agent basis. If an idle agent for the call is not available, the call is queued until an agent becomes available.

Note: Although OSSAIN routing and queuing has the capability to queue calls for OSSAIN services, this queuing capability is optional and can be prevented through switch datafill.

OSSAIN routing and queuing is accomplished using the QMS CAM (Call and Agent Manager). A QMS application is created for OSSAIN with its own set of call queues and call queue profiles. Other functionality provided by routing and queuing is the connection and release of a voice circuit between the switch and an SN at the request of the SN.

Routing and queuing components

The routing and queuing components of OSSAIN call processing are described in the following subsections:

- Overview—introduces how OSSAIN routing and queuing is done (page 56)
- Routing and queuing configurations—provides a high-level view of the different configurations (page 57)
- Functions and session pools—shows how functions and session pools are used to implement the configurations (page 61)
- Routing and queuing table relationships—gives a pictorial view of the call processing sequence for the OSSAIN routing and queuing tables (page 65)
- Datafilling functions and session pools—describes the switch datafill required to define the functions and session pools for OSSAIN (page 66)
- Datafilling the QMS setup for routing and queuing—describes the switch datafill required to set up the OSSAIN application of QMS (page 69)
- Voice link connections between the switch and SN—describes how voice connections are made between a DMS TOPS switch and an SN (page 75)
- SN originated calls—describes how SNs originate calls on the switch (page 78)

Overview

SN functions

As stated in Chapter 1, a *function* is a service or a portion of a service. Some examples include branding, automated billing, and Yellow Pages service. While a function can be provided by an SN, an operator, or a TOPS automated system, the routing and queuing capability of OSSAIN deals solely with functions provided by SNs.

Function identifier

Once OSSAIN determines that the desired function is provided by an SN, routing and queuing begins. The function has an associated *function identifier,* which is a number that uniquely defines the function. The function identifier is provided as input by call processing when a call session with an SN is required.

Session pool identifier

A session pool is a group of sessions on an SN that serve the same function or functions. Like functions, each session pool has an associated *session pool identifier,* which uniquely defines the session pool.

One SN can have one or more session pools. Each session pool contains a number of sessions. Each session has an associated session identifier (ID) which uniquely defines the individual session in that session pool.

When a call needs to be routed to a function provided by an SN, the routing and queuing capability examines the switch datafill against the function identifier. This determines which QMS call queue is associated with the function.

QMS CAM

A request is then sent to the QMS CAM for an agent that serves the call queue. Session pools have call queue profiles datafilled against them to indicate which call queues are served by agents in that session pool. These agents are defined and made available to the QMS CAM prior to call processing.

There is a one-to-one mapping between QMS CAM agents and OSSAIN sessions. Once an agent is selected by the CAM, the data for that agent indicates the session pool and session ID associated with the agent. The session pool and session ID then are used to establish a call session with the SN.

Routing and queuing configurations

The following configurations are discussed and illustrated:

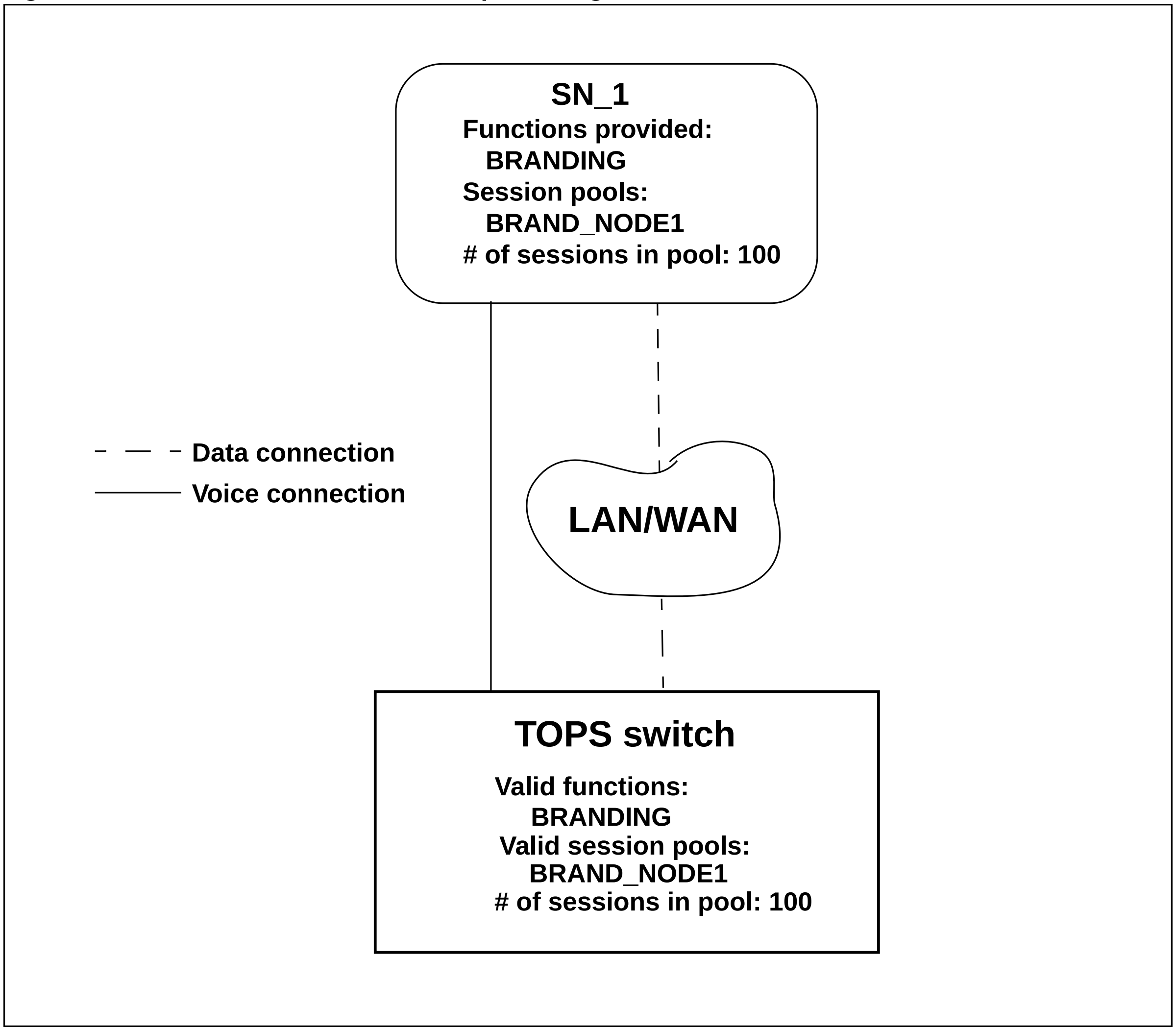
- one function on one SN
- multiple functions on one SN
- one function on multiple SNs

One function on one SN

The simplest configuration for distributing calls is a single function provided by a single SN. The function identifier and session pool identifier are datafilled on both the switch and the SN, as well as the maximum number of simultaneous sessions that can be handled by that session pool.

The following figure shows a sample of the one function on one SN configuration. In the figure, the function is branding.

Figure 8 One function on one SN sample configuration



In the previous figure, SN_1 provides branding. Once OSSAIN determines that a call requires branding, the routing and queuing capability selects an idle agent for branding. Because the BRAND_NODE1 session pool has 100 sessions serving the branding call queue, there are 100 agents from which to choose for branding.

If all 100 agents are in use, the call can be queued until an agent becomes available. Once an agent is selected, a call session is established with SN_1 through the LAN/WAN using the session pool and session ID stored in the agent data for the selected agent. After the call session is established, SN_1 can request that a voice connection be made.

Note: See “OAP overview” on page 118, for more information on how the data connection is established. See“Voice link connections between the switch and SN” on page 75, for details on how the voice connection is made.

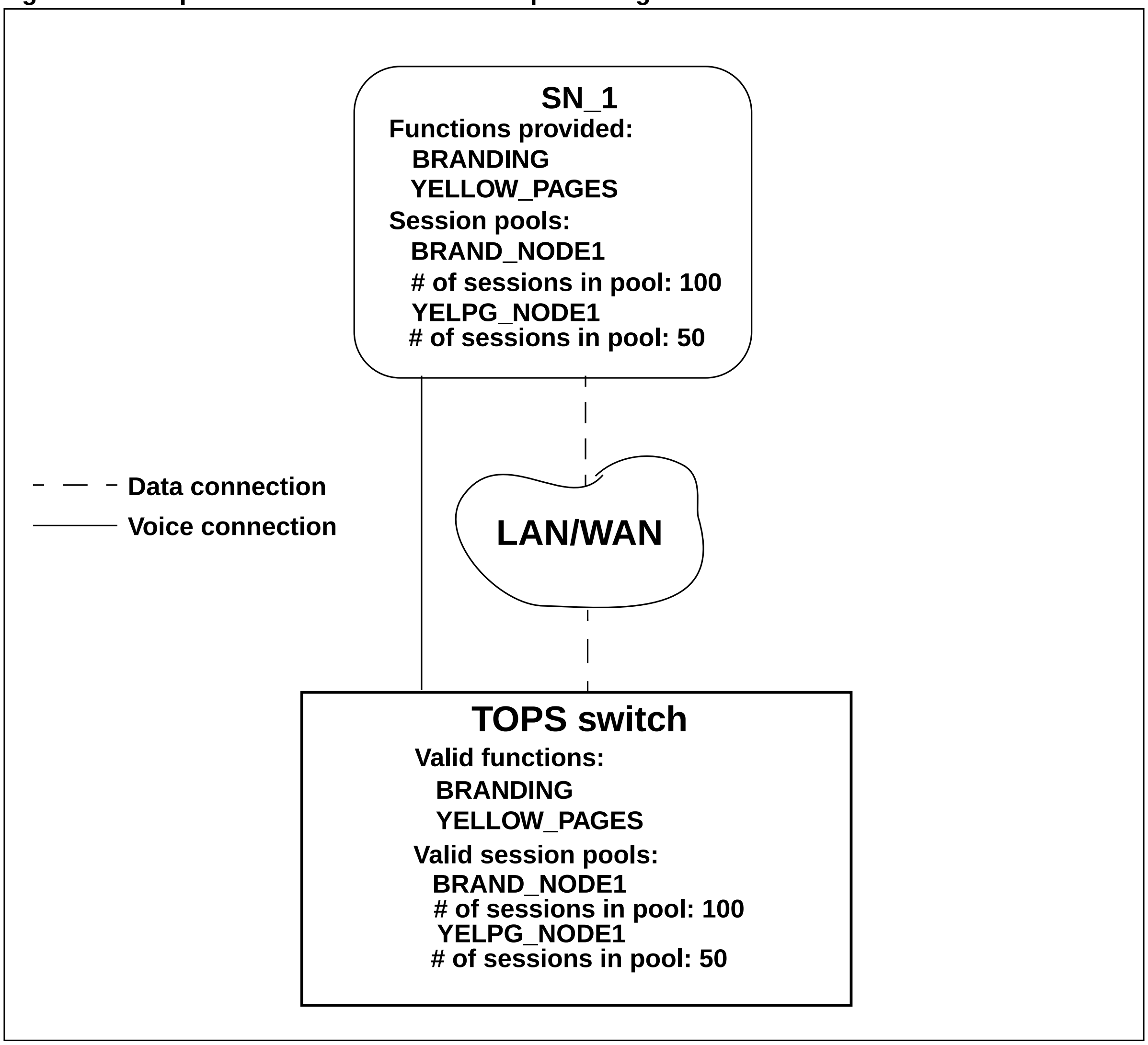
Multiple functions on one SN

One SN can provide many functions. Each function has a function identifier datafilled on both the switch and the SN. Session pools and the number of sessions in each pool must also be datafilled on both the switch and the SN.

There are several ways that session pools can be used to distribute calls between the various functions on one SN. One way is to establish a dedicated session pool for each function.

The following figure shows a sample of the multiple functions on one SN configuration. In the figure, the functions are branding and Yellow Pages service.

Figure 9 Multiple functions on one SN sample configuration



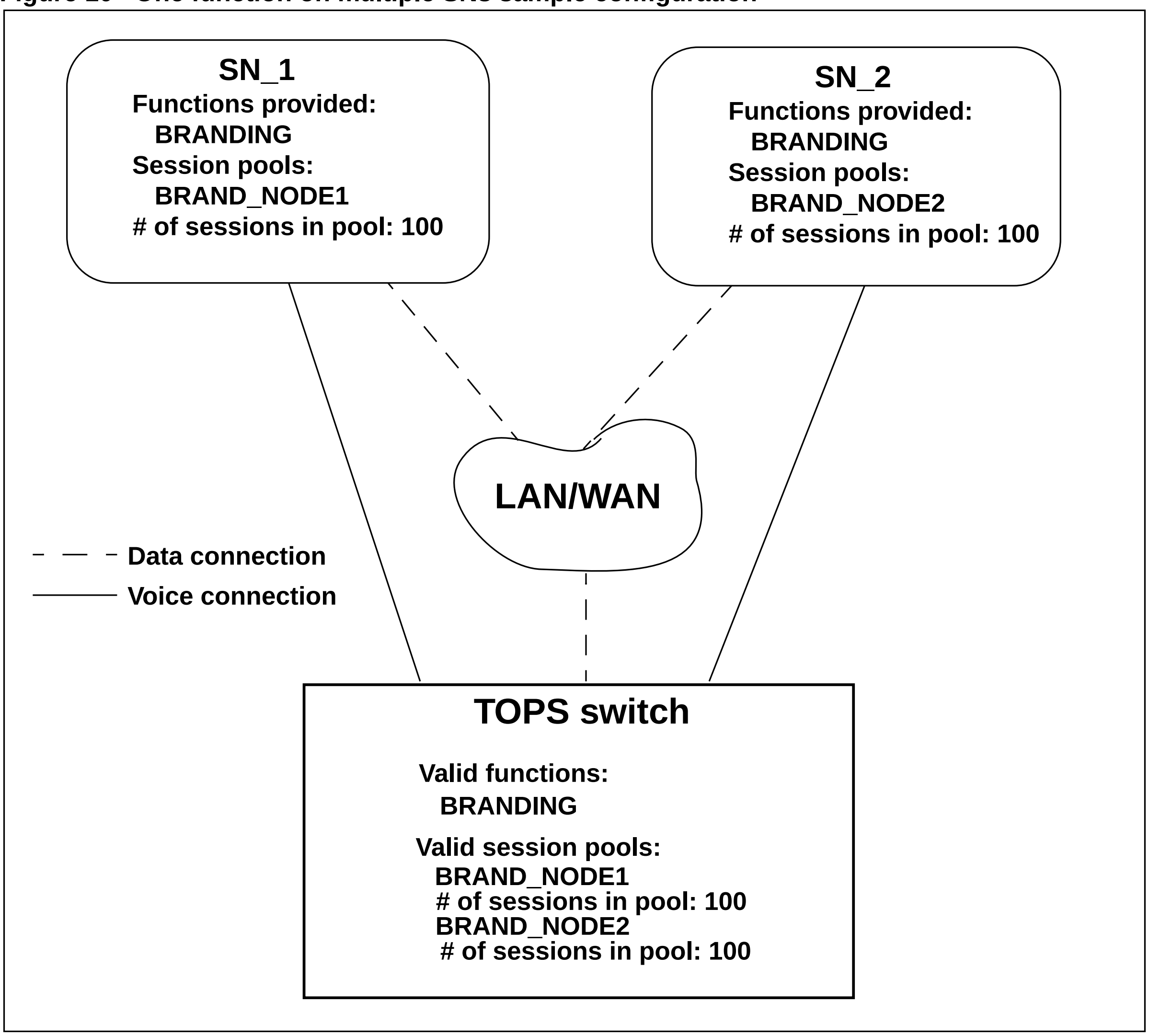
In the previous figure, SN_1 provides both branding and Yellow Pages functions. If branding is requested, OSSAIN routing and queuing selects from one of the 100 BRAND_NODE1 agents. If Yellow Pages is requested, OSSAIN selects from one of the 50 YELPG_NODE1 agents. A BRAND_NODE1 agent can be used only for the branding function and a YELPG_NODE1 agent can be used only for the Yellow Pages function.

One function on multiple SNs

One function can be provided on multiple SNs. This configuration is especially useful when one SN is not sufficient to handle the volume of traffic for a given function.

The following figure shows a sample of the one function on multiple SNs configuration. In the figure, the function is branding.

Figure 10 One function on multiple SNs sample configuration



In the previous figure, both SN_1 and SN_2 provide the branding function. The BRAND_NODE1 and BRAND_NODE2 session pools are both set up to serve the BRANDING call queue. Once the BRANDING function is requested, routing and queuing selects the most idle agent from among the two session pools. Since there are 100 agents in the BRAND_NODE1 session pool and 100 in the BRAND_NODE2 pool, there is a total of 200 agents from which to choose. Once an agent is selected, a call session is established with the associated SN.

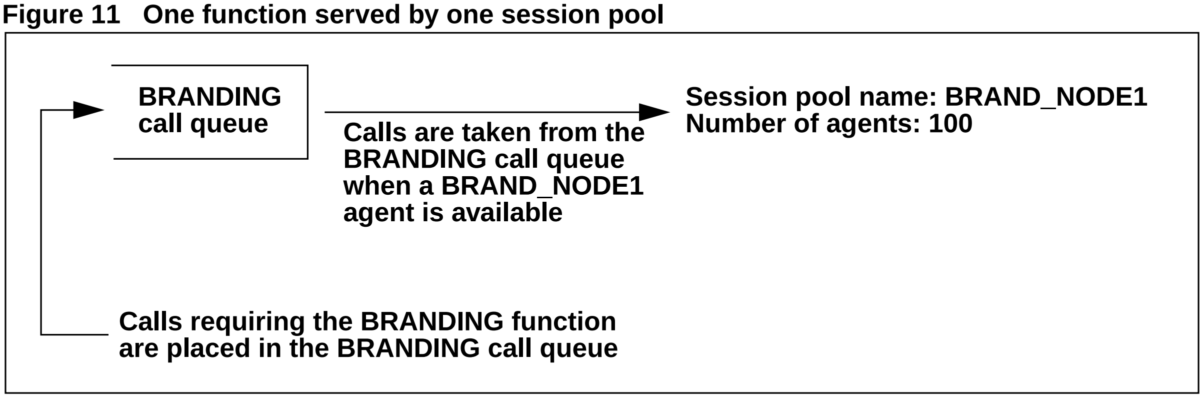
Functions and session pools

This section expands on the sample configurations discussed in the previous section. Each configuration is examined to see how functions and session pools are used to implement OSSAIN routing and queuing.

One function on one SN

In this configuration, one branding call queue and one branding session pool are set up on SN_1. There are 100 agents in the BRAND_NODE1 session pool that can serve the branding function. The following figure shows one branding session pool serving one branding call queue.

Figure 11 One function served by one session pool

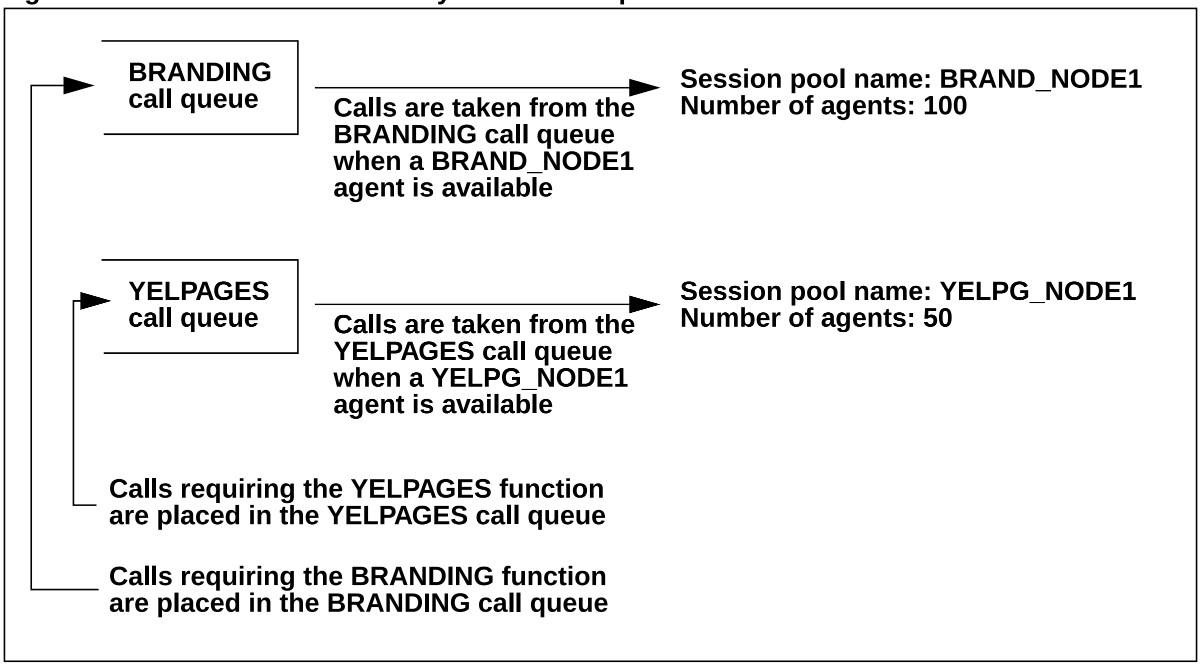


Note: Although this figure shows a branding call queue, it should be noted that calls are *not* queued if an agent is immediately available. Furthermore, queuing is optional and can be prevented through switch datafill. However, all calls requiring an OSSAIN agent are assigned a call queue so that an agent can be requested from the CAM.

Multiple functions on one SN

In this configuration, both branding and Yellow Pages functions are set up on SN_1. This type of configuration can be set up in one of three ways. First, two separate session pools can be set up, one for each function. The following figure shows two session pools, one for branding and the other for Yellow Pages.

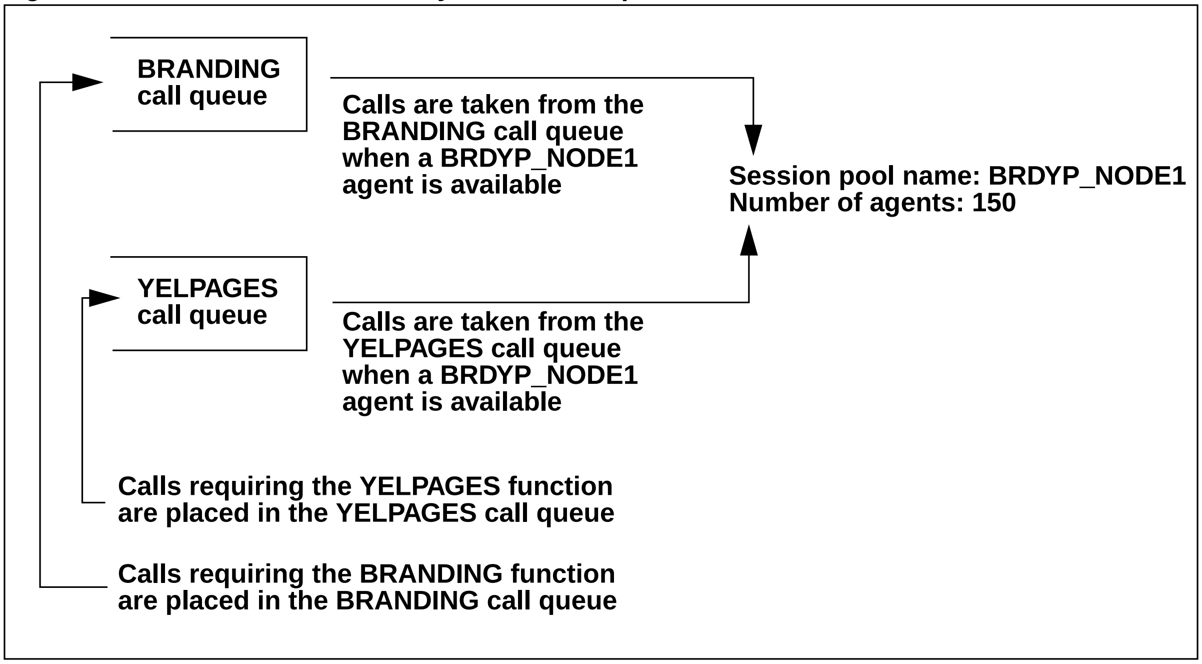
Figure 12 Two functions served by two session pools



The second way to set up two different functions is by having a single session pool that serves both functions. This strategy allows SNs to group their sessions together so that any individual session can be used for either function.

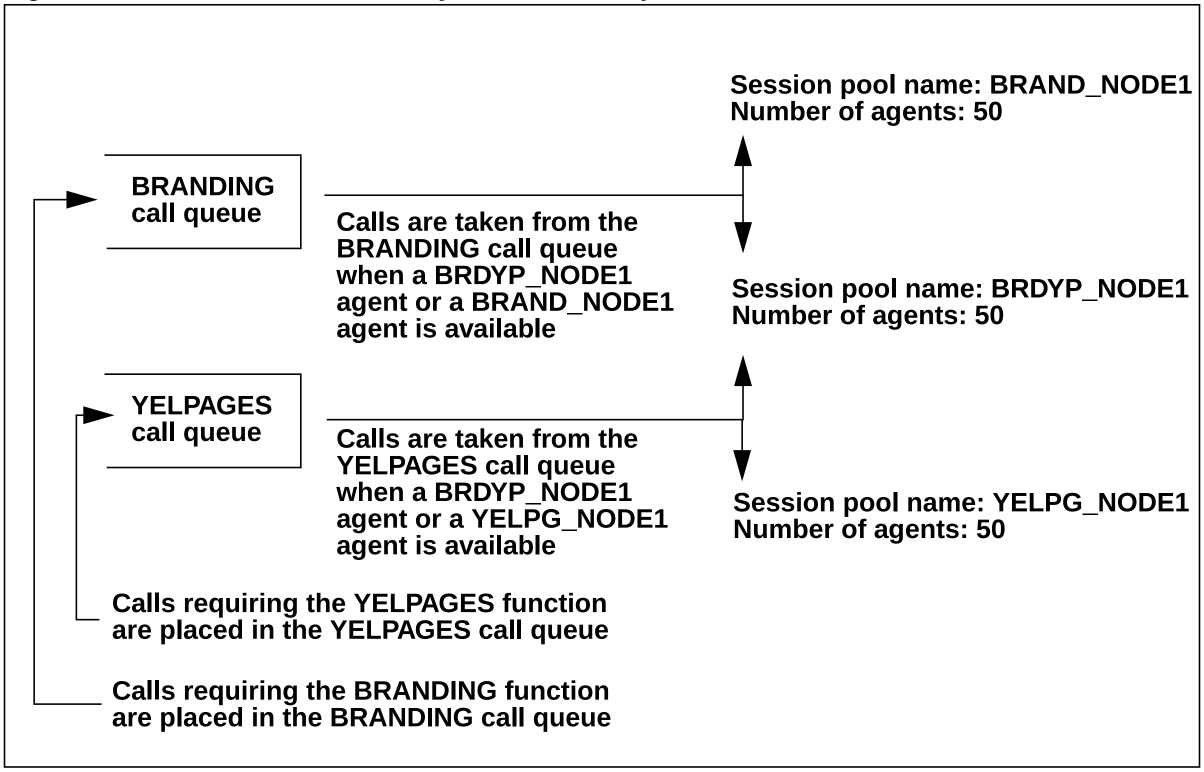
Once OSSAIN establishes a call session with the SN, the SN uses the function identifier to invoke the appropriate process. The following figure shows one session pool with 150 agents that serves both branding and Yellow Pages.

Figure 13 Two functions served by one session pool



A third way to set up multiple functions is to combine these two strategies. For example, SN_1 could provision three session pools, as follows: 50 agents to serve branding calls, 50 agents to serve Yellow Pages calls, and 50 agents to serve both. This type of setup ensures that at least 50 agents are dedicated to each function, while the remaining 50 agents can be used by *either* function. The following figure shows three session pools serving branding and Yellow Pages.

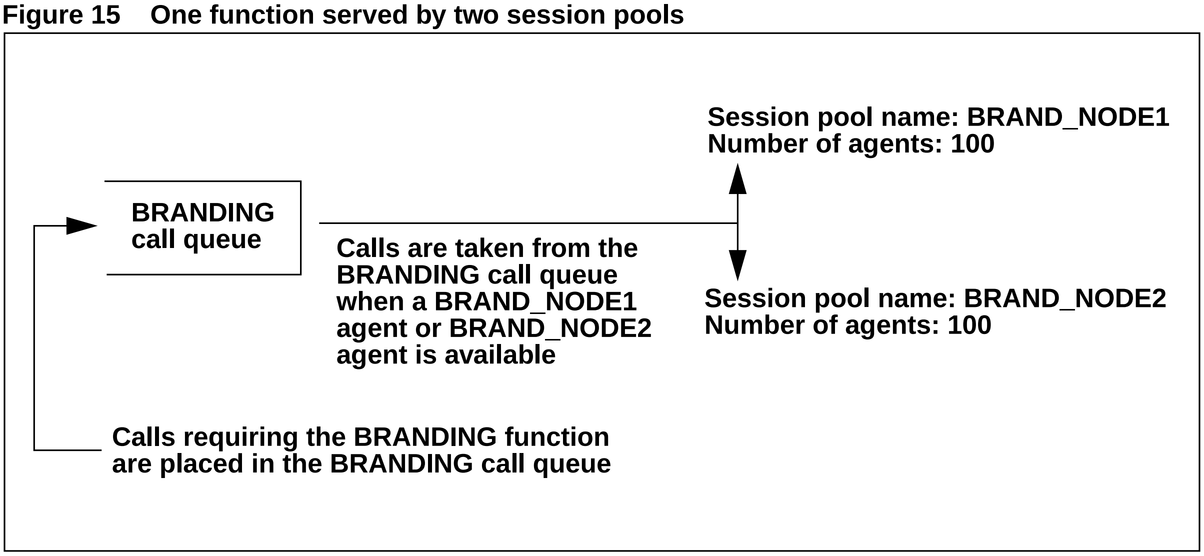
Figure 14 Two functions served by three session pools



One function on multiple SNs

In this configuration, two SNs both serve branding. This is accomplished by having one function served by two session pools. The following figure shows one session pool on each SN, both serving branding.

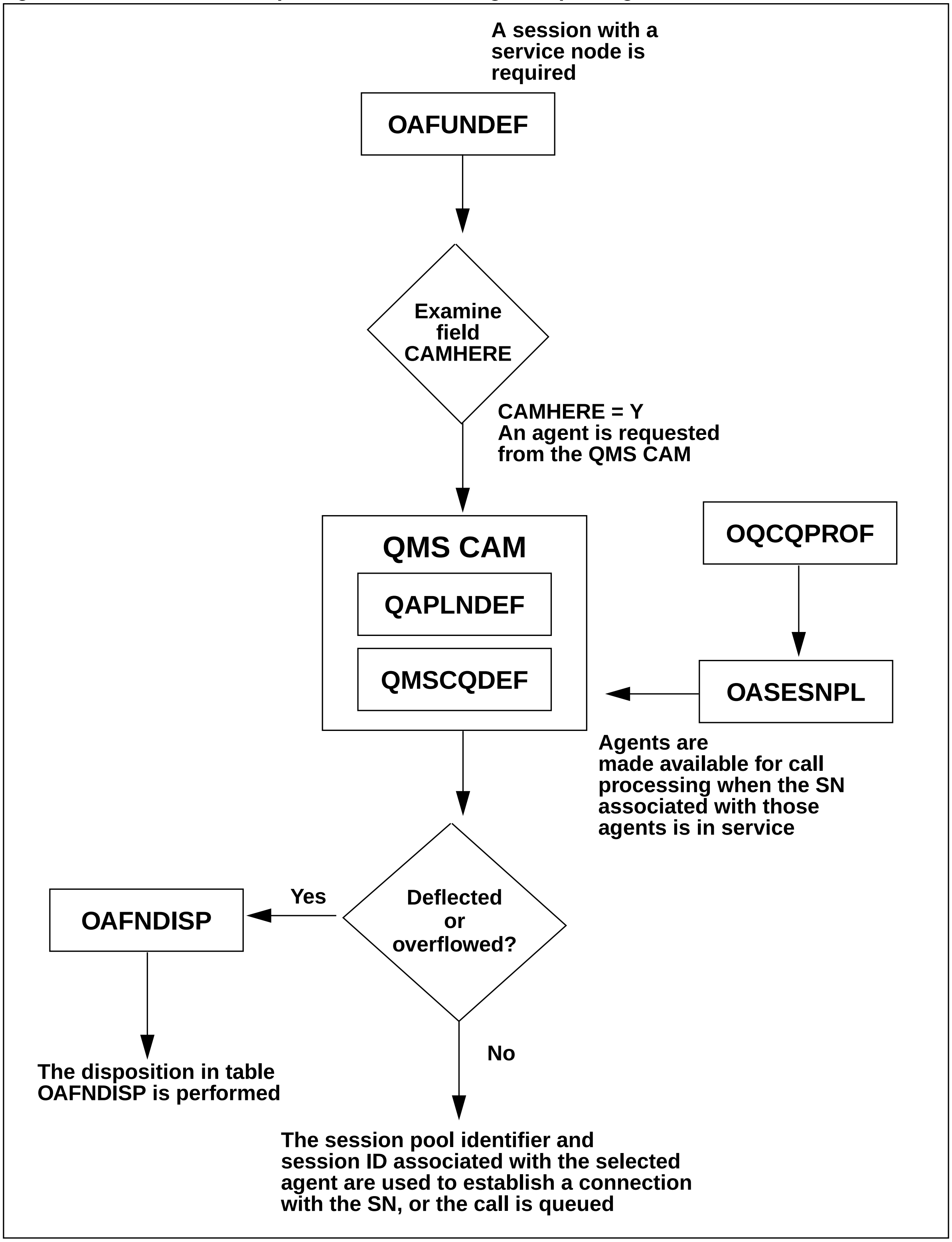
Figure 15 One function served by two session pools



Routing and queuing table relationships

The following figure shows the tables described in this section.

Figure 16 Table relationships for OSSAIN routing and queuing



Datafilling functions and session pools

This section focuses on how OSSAIN routing and queuing is achieved. Table datafill required for functions and session pools is discussed as well as the relationships between the tables. Example datafill for the tables in this discussion corresponds to the examples provided in the routing and queuing configurations.

Note: For a complete description of table datafill, refer to Chapter 7: "OSSAIN data schema."

The following tables are described:

- OAFUNDEF (OSSAIN Function Definition)
- OASESNPL (OSSAIN Session Pool)

Defining functions—table OAFUNDEF

All functions are defined in table OAFUNDEF. Functions in table OAFUNDEF are provided by an SN, an operator, or an automated system. However, the routing and queuing functionality is responsible for functions provided *only by SNs*. Therefore, the example tuples are for functions on SNs only.

The FUNCID (function identifier) field is the key to the table. In addition, the function identifier is passed in the Open Automated Protocol (OAP) and needs to be coordinated between the switch and any SNs that provide the function being datafilled.

The FUNCNAME (function name) field is a character string that describes the function. This character string is used to datafill other OSSAIN tables that contain a reference to this function.

The FUNCAREA (function area) field contains information specific to the type of function provider. The FUNCTYPE (function type) selector subfield defines one of three types of functions, depending on which system provides it:

- SN (for functions provided by an SN)
- TOPSOPER (for functions provided by an operator)
- TOPSAUTO (for functions provided by an automated system)

The ORIGSERV subfield also requires datafill when FUNCTYPE = SN. It is used to apply a base service to an OSSAIN call routed to an SN, as follows:

- TASERV (toll assistance service)
- DASERV (directory assistance service)
- INTCSERV (intercept service)

Note 1: When calls are transitioned to SN functions, subfield USESERV of table OAFUNDEF is used. When USESERV is set to Y, it indicates the service defined in ORIGSERV should be used for a call in transition (transfer, trigger, or disposition routing).

Note 2: Calls receiving DA or intercept service are in a restricted mode while at the SN. DA and intercept calls should be routed to an SN only to determine billing for the call before transferring the call to an operator.

When FUNCTYPE = SN, the CAMHERE subfield indicates whether the QMS CAM is located at this switch or if it is centralized at a host switch for this function.

When CAMHERE = Y, the OSSAIN QMS call queue for this function also must be specified. (See “Defining the OSSAIN call queues—QMSCQDEF” on page 71, for more information on call queues.)

The following figure shows two tuples in table OAFUNDEF, one for branding and the other for Yellow Pages.

Figure 17 MAP display example for table OAFUNDEF

```
FUNCID          FUNCNAME             FUNCAREA
-----------------------------------------------------------------------------------
0               BRANDING             SN  TASERV N N N N Y Y CQ0 Y Y
1               YELLOW_PAGES         SN  DASERV N N N N Y Y CQ1 N
```

Note: Table OAFUNDEF also has subfields that apply to trigger events, session begin timeouts, no-automation calls, and automatic voice connections. Refer to Chapter 7: “OSSAIN data schema,” for more information on these fields.

Defining session pools—table OASESNPL

All session pools are defined in table OASESNPL. A session pool can be thought of as a team of operators set up to serve the same call queues.

There are two types of session pools, depending on how the sessions in the pool are selected. The type of session pool is determined by datafilling the ORIGTYPE (origination type) subfield with one of the following values:

- SUBSCRIBER—This type of session selection is used for OSSAIN calls originated by a subscriber. This type of session pool is used for sessions that are selected by routing and queuing for communication between the switch and the SN.
- SN—This type of session pool is used for sessions that are selected by an SN. This occurs when an SN wants to cause a call to be spun up in the switch, much like an operator delay call.

The SESNPLID (session pool identifier) field is the key to this table. The session pool identifier is passed in the OAP protocol and needs to be coordinated between the switch and the SN that uses the session pool being datafilled.

The SESNPLNM (session pool name) field is a character string that describes the session pool being datafilled.

The MAXSESN (maximum sessions) field indicates the maximum number of simultaneous sessions that are allowed for this session pool. The number of simultaneous sessions handled by a session pool also indicates the maximum number of simultaneous calls that the session pool can handle.

The NODENAME (node name) field is the name of the SN associated with this session pool. Node names must first be datafilled in table OANODNAM before they can be datafilled in this table.

Note: For a complete description of table OANODNAM, refer to Chapter 7: “OSSAIN data schema.”

The ORIGAREA (origination area) field contains information specific to the switch. The ORIGTYPE (origination type) selector subfield defines one of two types of session pools: SUBSCRIBER or SN. The value SUBSCRIBER means that OSSAIN routing and queuing selects the session for communication with the SN, whereas the value SN means that the SN selects the session for communicating with the switch.

Note: Only the value SUBSCRIBER is shown in the example. See “SN originated calls” on page 78, for more information on the SN value in the ORIGTYPE subfield.

The SWTCHTYP (switch type) selector subfield indicates the role of the TOPS switch in the routing and queuing for this session pool. This subfield only appears if ORIGTYPE = SUBSCRIBER. The S (standalone) value indicates that the agents corresponding to the sessions in this session pool are defined in the QMS CAM associated with this switch.

The ONHKTMR and CALLTMR subfields specify the sanity timer information. See “Sanity timers” on page 124, for information on the values for these subfields in table OASESNPL.

The TRIGEVNT (trigger event) subfield applies when the ORIGTYPE is SUBSCRIBER. It specifies whether or not the session pool is used only for trigger event inform messages. For details, refer to Chapter 7: “OSSAIN data schema.”

The OCQPROF (OSSAIN call queue profile) subfield is an index into table OQCQPROF, which is discussed in “Datafilling the QMS setup for routing and queuing” on page 69. OSSAIN call queue profiles are similar to operator profiles in that both are used to indicate which call queues are served by agents with a given profile. A separate set of call queue profiles is used for OSSAIN and has no effect on existing operator call queue profiles.

The following figure shows three tuples in table OASESNPL, each one with a different session pool setup.

Figure 18 MAP display example for table OASESNPL

```
SESNPLID SESNPLNM  MAXSESN NODENAME  ORIGAREA
-----------------------------------------------------------------------------
4      AABS      100   AABS_01   SUBSCRIBER S USEDEFLT USEDEFLT N 3 UDP 7001 7002
5      DEBIT     50    DEBIT_01  SUBSCRIBER S USEDEFLT USEDFLT  N 5 UDP 7001 7002
6      BRANDING  100   BRAND_01  SUBSCRIBER S USEDEFLT USEDEFLT N 2 UDP 7001 7002
```

Note: Table OASESNPL also has subfields that apply to the transport protocol used in communicating with the SN associated with the session pool. Refer to Chapter 7: “OSSAIN data schema,” for more information on these fields.

Datafilling the QMS setup for routing and queuing

OSSAIN routing and queuing uses the QMS CAM to achieve its functionality. OSSAIN is defined as a QMS application with its own set of call queues, call queue profiles, and agent queues. Like TOPS QMS, OSSAIN allows both absolute and relative call queue priorities, deflection criteria, and optional priority agent queue assignment.

This section shows example datafill for the QMS portion of OSSAIN routing and queuing. The actual datafill for OSSAIN routing and queuing must be configured and engineered on an individual basis.

Note: For complete information on TOPS QMS queuing and related datafill, refer to the *Customer Data Schema Reference Manual*.

The following four steps are performed to set up routing and queuing:

1 Define the OSSAIN QMS application in table QAPLNDEF (QMS Application Definition)

2 Define the OSSAIN call queues in table QMSCQDEF (QMS Call Queue Definition)

3 Define the OSSAIN call queue profiles in table OQCQPROF (OSSAIN QMS Call Queue Profiles)

4 Define the deflection/overflow processing (disposition routing) in table OAFNDISP (OSSAIN Function Disposition)

Defining the OSSAIN application in QMS—QAPLNDEF

The OSSAIN application is defined in table QAPLNDEF. The following figure shows two sample tuples, one for TOPS and one for OSSAIN.

Figure 19 MAP display example for table QAPLNDEF

```
APLNAME    CALLQS    CQELEMS   AGENTQS   NUMAGNTS   CTSEARCH
----------------------------------------------------------------------
TOPS       255       3000      127       170        0
OSSAIN     220       5000      100       1500       0
```

Note: For details on table QAPLNDEF, please refer to the *Customer Data Schema Reference Manual*.

Defining the OSSAIN call queues—QMSCQDEF

OSSAIN call queues are defined in table QMSCQDEF. This table has a two part key: application name and call queue number. OSSAIN call queues are independent of existing operator call queues.

Note: For details on table QMSCQDEF, please refer to the *Customer Data Schema Reference Manual*.

Figure 20 MAP display example for table QMSCQDEF

APPLNCQ	CQPRIO	CQAGS	CQMAXSIZ	DEFLAREA			PRAQAREA		
TOPS 0	20	18	300	Y	600	5	Y	4	100
TOPS 1	20	10	200	N			Y	4	0
TOPS 2	35	18	300	Y	600	5	Y	3	100
TOPS 3	35	10	240	Y	600	5	N		
OSSAIN 0	20	18	0	N			N		
OSSAIN 1	20	10	0	N			N		
OSSAIN 2	35	18	300	Y	1200	5	N		
OSSAIN 3	35	10	240	Y	1500	5	N		

Note: Call queues OSSAIN 0 and OSSAIN 1 are set up to disallow queuing of calls by setting CQMAXSIZ = 0. On the other hand, OSSAIN 2 and OSSAIN 3 are set up to provide a queuing capability for functions associated with these call queues.

Defining the OSSAIN call queue profiles—OQCQPROF

OSSAIN call queue profiles are defined in table OQCQPROF. This table is very similar to its TOPS counterpart, table TQCQPROF (TOPS QMS Call Queue Profiles). The only difference between the tables is that table TQCQPROF includes information on how to handle delay calls from operators with a given call queue profile. This information is not needed for OSSAIN because SN originated calls are not handled in the same way as delay calls from operators.

The CQPROFNM (call queue profile number) field defines the call queue profile numbers that can be datafilled in table OASESNPL. Call queues datafilled in this table are defined as OSSAIN call queues. For example, CQ0 refers to the OSSAIN 0 call queue datafilled in table QMSCQDEF.

The AGENTQ (agent queue) field specifies which agent queue should be used for idle agents with this call queue profile number. When an agent is requested from the QMS CAM, the agent queues are searched to find an agent that has the requested call queue in its profile. Any number of agent queues can be used, but a good starting point would have one agent queue for each call queue profile datafilled in table OQCQPROF.

The PRIOAREA (priority area) field indicates which call queues are served by agents with this profile number. This area is identical to the PRIOAREA in the table TQCQPROF used for TOPS (Refer to the *Customer Data Schema Reference Manual* for information on table TQCQPROF).

The PRIOAREA field has a selector value of either OFC or PROF, as follows:

- The OFC value indicates that priorities and AGS (Assignable Grade of Service) values are to be determined by office-wide datafill in the QMS CAM. A simple list of call queues is specified here.
- The PROF value indicates that priorities and AGS values are to be specified uniquely for this profile. Two priorities can be set up, as follows:
  - — Absolute priorities are established by creating from one to four lists of up to 32 call queues each, where the lists are delimited by $.
  - — Relative priorities are established by specifying an AGS value for each call queue in the range of 10 to 80 (specify 10 if no aging is to be applied).

Figure 21 MAP display example for table OQCQPROF

```
CQPROFNM        AGENTQ       PRIOAREA
----------------------------------------------------------------------
0               AQ0          OFC (CQ0) $
1               AQ1          OFC (CQ1) $
2               AQ2          OFC (CQ0) (CQ1) $
3               AQ3          PROF (CQ0 10) (CQ1 80) (CQ2 10) $
```

Sample scenario for table OQCQPROF

Using the example configurations (see “Routing and queuing configurations” on page 57), the branding function is associated with CQ0 (from table OAFUNDEF), and call queue profile number 0 is used for agents that serve CQ0 (from table OQCQPROF).

From table OASESNPL, both the BRAND_NODE1 and BRAND_NODE2 session pools were datafilled as using call queue profile number 0 (OCQPROF subfield). Since the MAXSESN field for each of those session pools was set to 100, there would be 200 agents placed in agent queue AQ0, waiting to serve calls from CQ0 (the branding function).

Note: Agents are not actually made available for call processing until the SN associated with those agents is brought into service. For more information on bringing nodes into service, refer to Chapter 10: “OSSAIN maintenance.”

Defining the deflection/overflow processing—OAFNDISP

Deflections and overflows for OSSAIN calls can cause the call to be routed to treatment, transferred to another function, transferred to a control list, or outpulsed to a directory number (DN). Deflections and overflows are determined by the QMS CAM in the same way that they are currently determined for operator calls.

Deflections occur when the following three conditions are met:

- An agent is requested from the QMS CAM for a given call queue.
- The queue is set up to allow deflections.
- The QMS CAM has determined that the predicted wait time before the call can receive an agent exceeds the limit for that call queue. This limit is set in table QMSCQDEF.

Overflows occur when no agents are available for a given call queue and the call queue would exceed its maximum size if the call were to be queued. The maximum size also is set in table QMSCQDEF.

Disposition routing for deflections and overflows are datafilled in table OAFNDISP on an individual function basis. In addition to deflections and overflows, this table also specifies the disposition for calls that are blocked from a particular function, or for calls that fail to send an OAP session begin message to an SN once an agent for that function has been selected (also called *origination failures*).

Note: Refer to “Transitions” on page 103, for information on control lists, function blocking, and origination failures.

Outpulsing disposition

OSSAIN has the option to outpulse to a designated DN when a session with an SN cannot be obtained. This disposition is available only to calls that have not received any processing from an SN and which are in the *initial call setup* state. It is not available for calls that transition or trigger to an SN.

A call in the initial call setup state is defined as one that arrives at the TOPS switch and, through QMS refinements, is routed to OSSAIN processing. The final CT4Q obtained during QMS refinements is mapped to a control list, which contains the name of a function provided by an OSSAIN SN. Datafill in table OAFNDISP specifies the disposition used for the call if it fails to establish a session with an SN.

Note: At initial call arrival, if the OSSAIN function used to route the call to an SN is blocked for the calling subscriber, the call receives the default treatment datafilled in table OAINPARM instead of receiving the CLDOUTP disposition action datafilled in the IBLKACTN field.

OAFNDISP actions

In table OAFNDISP, the FUNCNAME field refers to the name of the requested function (defined in table OAFUNDEF). The other eight fields in table OAFNDISP reflect two sets of four types of actions. One set applies to initial call setup calls. The other set applies to transition/trigger calls.

The eight action fields are as follows:

- IDFLACTN (initial deflection)
- IOVFACTN (initial overflow)
- IBLKACTN (initial blocking)
- IOFLACTN (initial origination failure)
- TDFLACTN (transition deflection)
- TOVFACTN (transition overflow)
- TBLKACTN (transition blocking)
- TOFLACTN (transition origination failure)

Note: Refer to Chapter 7: “OSSAIN data schema” for details on the disposition values and refinements.

The following figure shows a tuple for the branding function in table OAFNDISP.

Figure 22 MAP display example for table OAFNDISP

```
FUNCNAME          IDFLACTN                  IOVFACTN
          IBLKACTN               IOFLACTN
                  TDFLACTN                  TORFACTN
          TBLKACTN               TOFLACTN
------------------------------------------------------------------------
BRANDING         CLDOUTP Y Y 9191234567 Y   CLDOUTP Y N N
          GOTOFN TOPS_BRANDING   GOTOFN LIVE_OPER
                  GOTOFN TOPS_BRANDING      GOTOCTL CTLLIST3
          TREAT VACT             GOTOFN ALT_NODE
```

Parameter MAX_TRANSFERS_BEFORE_CONNECT

An OSSAIN parameter, MAX_TRANSFERS_BEFORE_CONNECT, can be set to limit the number of transfers performed in the switch when attempting to route the call to an SN. This parameter ranges from 0 to 5. The parameter is limited to 5 to prevent failure cases from using excessive switch resources. Once the maximum is exceeded, the call is routed directly to a treatment indicated by another OSSAIN parameter, DEFAULT_TREATMENT.

Parameter DEFAULT_TREATMENT

The DEFAULT_TREATMENT parameter specifies a treatment name, which must be datafilled in table TMTCNTL. Once the call is successfully routed to a function provider, the internal transfer count in the switch is reset to 0 until a connection to the next function is attempted. The transfer count is incremented each time the call is transferred due to deflection, overflow, blocking, or origination failure.

Note: Refer to Chapter 7: “OSSAIN data schema,” for an example of the OSSAIN parameter table, OAINPARM.

Voice link connections between the switch and SN

Voice link connections between the switch and SN are made by the SN at any time after the data connection has been established. If the SN determines that a voice connection is required, it sends an OAP message requesting that the connection be made.

In the OAP request, the SN supplies a logical voice channel number. The switch maps the logical voice channel to an actual voice circuit datafilled in table TRKMEM. The switch and SN must have parallel datafill to ensure that both sides connect to the same voice circuit. The operating company and SN vendor are responsible for engineering the number of voice links that are used for a given node.

Note 1: For information on an alternate method that allows the switch to select the voice circuit, refer to “Auto voice link selection and connection” in Chapter 4: “OSSAIN enhancements.”

Note 2: For broadcast voice links, the SN must supply the node identifier (ID) as well as the logical voice channel number in the OAP request. For more information on broadcast voice links, refer to Chapter 4: “OSSAIN enhancements.”

Dedicated trunk groups

OSSAIN requires that dedicated trunk groups be defined for voice channels between the switch and SNs. Voice channels to one or more SNs can be combined in the same trunk group. The dedicated voice trunks for OSSAIN must be datafilled as group type TOPSVL.

Trunk group type TOPSVL

Trunks of group type TOPSVL share the following characteristics:

- Table control forces the selection sequence in table TRKGRP to be set to MIDL (Most Idle) for TOPSVL trunks.
- Trunk types can be 2W (two-way) instead of OG (outgoing). However, in order to datafill any 2W trunk in TRKSGRP, it must be datafilled as 2W in table TRKGRP.

- They require nonstandard supervision and signaling, which means the DMS switch does not have to seize a trunk and receive a return wink before using that trunk.
- The opulsettype field in table TRKSGRP must be set to NP (no pulse).
- The ipulsetype field (for 2W) in table TRKSGRP must be set to DP.
- Their start signal field must be set to IM (immediate) in table TRKSGRP.
- The RMB (remote make busy) field in table TRKSGRP must be set to Y. This field causes the voice trunk to go to the RMB state at the MAP when the SN goes off hook towards the switch.

Once dedicated trunk groups of type TOPSVL are datafilled and the voice circuits are datafilled in table TRKMEM, OSSAIN table OAVLMAP (OSSAIN Voice Link Mapping) must be datafilled to map the logical channel numbers received from the SNs to the actual circuits in table TRKMEM.

Note: For complete information on table TRKMEM, please refer to the *Customer Data Schema Reference Manual.*

Table OAVLMAP

All voice circuits datafilled in table OAVLMAP must first be datafilled in table TRKMEM. Tuples cannot be deleted from table TRKMEM until all references to that circuit are removed from table OAVLMAP.

The NDANDCH (node name and channel) field in table OAVLMAP is the key for table OAVLMAP, consisting of subfields NODENAME and VOICENUM.

- NODENAME is the name of the SN that submitted the voice connection request; this name requires datafill in table OANODNAM before it can be datafilled in table OAVLMAP.
- VOICENUM corresponds to the logical voice channel number supplied by the SN.

  Although each SN has its own set of logical voice channel numbers that map to actual voice circuits, the DMS switch makes no assumptions about what those voice circuits are connected to. This means that the set of logical voice channels for any given SN can be mapped to actual voice circuits that could terminate on another SN or on any other external system. The SN is responsible for coordinating the use of voice circuits on other external systems if those voice circuits are included in its set of logical voice channels.

The CLLI (common language location identifier) field and EXTRKNUM (external trunk number) field map to the two-part key in table TRKMEM.

For broadcast voice links, additional datafill in OAVLMAP specifies the method the switch uses to establish the voice path. Datafill also specifies the maximum number of simultaneous connections to a particular broadcast voice link and whether STR supervision applies.

Note: A voice link can function as only one type: either non-broadcast or broadcast. For details on broadcast voice links, refer to Chapter 4: "OSSAIN enhancements.".

The following figure shows two tuples in table OAVLMAP.

Figure 23 MAP display example for table OAVLMAP

```
NDANDCH                 CLLI         EXTRKNUM   BCSTAREA
----------------------------------------------------------------------
NODE_1 10               OSSAINVL1     2         N
NODE_2 11               OSSAINVL2     3         Y IMMEDIATE 800  Y OCTO
```

Non-broadcast voice link maintenance

OSSAIN two-way voice trunks use a basic maintenance strategy. To use a given voice trunk, both the switch and the SN must be on hook on that facility. If the SN goes off hook towards the switch, the trunk is displayed at the switch MAP as being in the RMB (remote make busy) state. Trunks in the RMB state are not used by the switch for call processing. If the SN attempts to select a trunk in a busy state, a return error message is sent to the SN.

When both the SN and the switch are on hook on the voice trunk facility, the trunk can be selected for call processing. After the switch returns a success response to the voice connection request, the SN can use that facility immediately.

Note: For details on the operation of broadcast voice links, refer to Chapter 4: "OSSAIN enhancements.".

OSAC voice connections

Refer to Chapter 3: "OSAC call processing," for information on OSAC voice connections.

SN originated calls

SN originated calls are the equivalent of operator delay calls. They are useful for services like message delivery where an SN would have to originate a call in the DMS switch to deliver a message to a subscriber.

SN originated calls are made by having the SN select the session pool and session ID for the call. This means that routing and queuing are bypassed because the key pieces of information are already present when the switch receives the OAP Session Initiation request from the SN.

To receive SN originated calls, table OASESNPL must indicate that the session pool is used for SN originations. This is done by datafilling the ORIGTYPE selector field with a value of SN. The only additional information needed when ORIGTYPE = SN are the sanity timer values for calls using sessions in this session pool.

The following figure shows a tuple of ORIGTYPE = SN in table OASESNPL. In the example, the MDS_01 node can originate calls using the MDS session pool. Since the MAXSESN field is set to 100, the switch is expecting the SN to originate using session IDs 0 to 99.

Positive assertion

Any time the switch receives an SN initiation request on a session that the switch determines is already active, the switch releases the first call (and any resources) and begins a new call.

Figure 24 MAP display example for table OASESNPL

```
SESNPLID    SESNPLNM    MAXSESN    NODENAME    ORIGAREA
------------------------------------------------------------------------------
7          MDS        100       MDS_01      SN USEDEFLT USEDEFLT UDP 7001 7002
```

Note: Table OASESNPL also has fields that apply to the transport protocol used in communicating with the SN associated with the session pool. For more information on these fields, refer to Chapter 7: “OSSAIN data schema.”

Trigger processing

This section focuses on TOPS and OSSAIN trigger processing. A trigger is an event that causes a call to be redirected to an SN, operator, or automated system. The following areas of OSSAIN trigger processing are discussed:

- call model
- TDPs (trigger detection points) and trigger events
- initial call setup TDP and datafill
- call floated TDP and datafill
- trigger table relationships
- trigger call flow scenarios
- float processing with TOPS operator and automated systems
- trigger processing parameters
- notification timer

Call model

OSSAIN trigger processing occurs during the following two call states, or points in call (PIC):

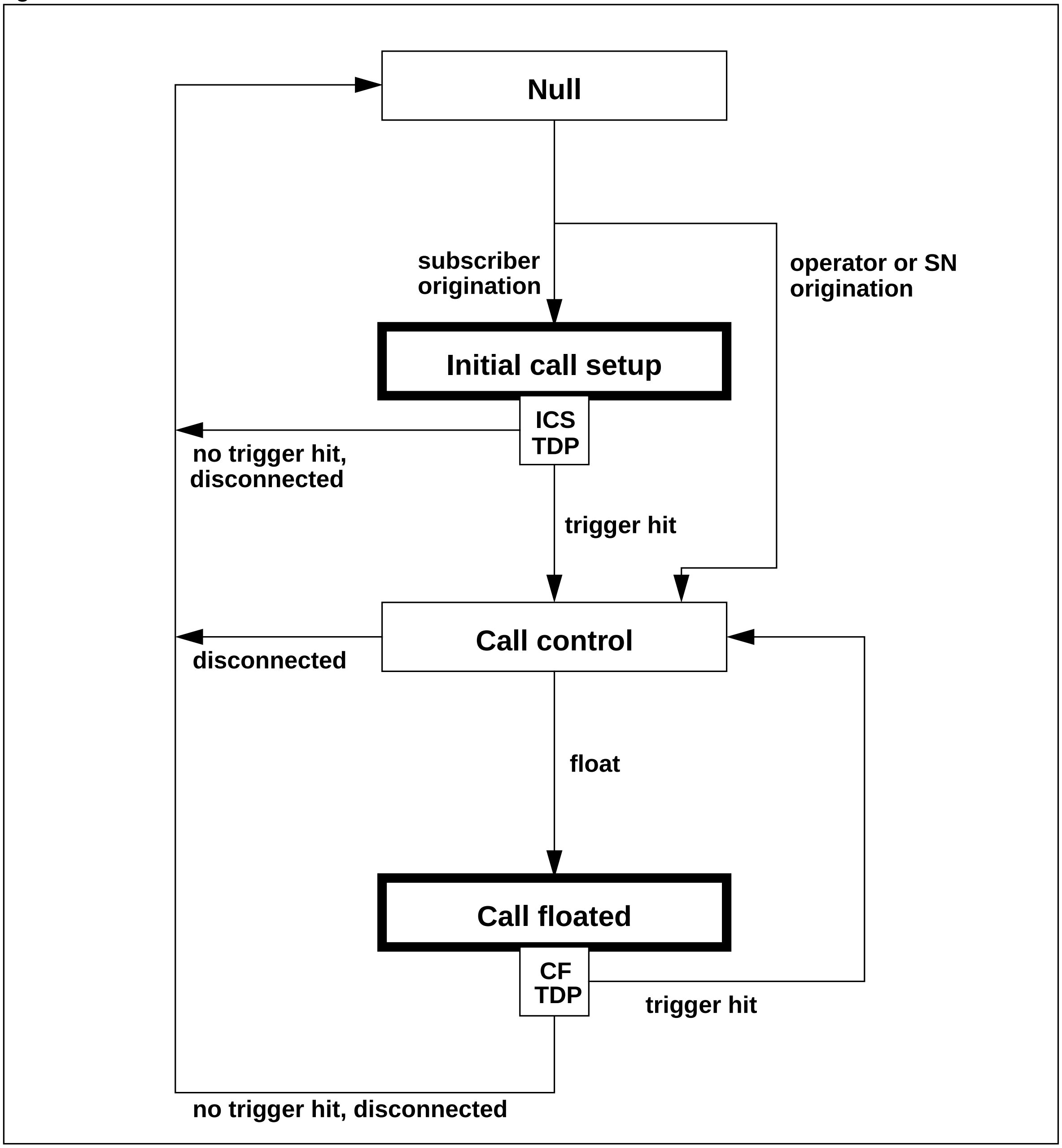
- initial call setup
- call floated

The DMS switch is solely in control of the call at these two PICs. Associated with the two PICs are two TDPs: initial call setup (ICS TDP) and call floated (CF TDP). At the TDP the switch examines trigger events and determines, based on OSSAIN trigger datafill, whether to initiate a session with an SN, operator, or automated system.

The following figure shows the OSSAIN call model, including the initial call setup and call floated PICs and their associated TDPs.

Note: Trigger profile index assignment can be provided, based upon QMS refinements, for both TOPS and OSSAIN calls. A trigger profile index is maintained on a per call basis and may be modified by the SN or operator as required.

Figure 25 OSSAIN call model



Note: DMS switch trigger processing occurs only for those PICs where the SN or operator is not linked to the call.

TDPs and trigger events

The term TDP describes the processing that occurs at the PICs where the DMS switch determines whether it needs to start a session with an SN, operator, or automated system.

Note: The ICS TDP can initiate a session with an SN, operator, or automated system, whereas the CF TDP can initiate a session only with an SN or operator.

Triggers define the conditions that must be met in order for the session to be started. A call triggers and starts a session when the following conditions are met:

- the call has reached the TDP
- the trigger event has occurred
- the trigger criteria are met

A TDP can contain one or more trigger events. The following table lists the TDPs, trigger events, and criteria for those events.

Table 3 OSSAIN TDPs and trigger events

TDP	Trigger event	Criteria
Initial call setup	Subscriber originated call	CT4Q name, based on call criteria using QMS, indicates call is OSSAIN
Call floated	Treatment Outpulsing failure Cause code No answer DTMF digits Flash Calling party disconnect Party disconnect (see Note) Answer Ringing Notification	Call is floated by one of the following: - SN - operator - TOPS automated system Trigger event is datafilled in the call floated trigger tables associated with the call
Note: This is any party other than the calling party (such as the called party). The distinction is made to support different actions depending on which event is processed.		

Initial call setup TDP and datafill

When a call arrives at a TOPS switch the TOPS QMS performs call queue assignment. QMS does this through a stepwise refinement of the *call type for queuing* or CT4Q. The refinement approach successively compares characteristics of the call against the datafilled queuing criteria. This allows the incoming traffic to be separated into different categories based on call attributes.

Before OSSAIN, QMS refinements resulted in a CT4Q assignment to a TOPS operator. With OSSAIN and the implementation of control lists, QMS allows CT4Q assignment to OSSAIN SNs. After passing through all the refinement criteria, a call is assigned a CT4Q, which determines the call queue and whether the call receives OSSAIN service.

Note*:* General TOPS calls can have a trigger profile index assigned to the call without requiring an OSSAIN CT4Q.

Call origination types supported in OSSAIN

In QMS processing, the call origination (CO) type is associated with a CT4Q in table QMSTOPS. For a call to receive OSSAIN service, the CO type assigned must be one of the following:

- UNSPEC (unspecified)
- OH (operator handled [0-])
- OA (operator assisted [0+])
- DD (direct dial [1+])
- DELAY
- 555
- TS (toll station)
- TSUB (toll subscriber)
- 211, 311, 411, 511, 611, 711, 811, 911
- 999
- HOM555 (home 555)
- FOR555 (foreign 555)
- SPARE1, SPARE2, SPARE3, SPARE4, SPARE5
- INTS (intercept service)
- CDIR (country direct)
- SLRN (Special Local Routing Number)

Note 1*:* If the CO type assigned is not supported, the call routes to treatment.

Note 2: A CO type of DD from a coin or hotel station is not supported and is routed to treatment.

Note 3: TOPS QMS also allows TOPS calls to receive OSSAIN preprocessing. Refer to Chapter 4: “OSSAIN enhancements” for more information and a list of additional CO types allowed for preprocessing.

Datafilling CT4Qs is required in order to direct calls to OSSAIN. Table CT4QNAMS (Call Type for Queuing Names) requires datafill to set up the initial call setup TDP trigger.

Note: Refer to Figure 36, “Table relationships for OSSAIN triggers,” on page 97, for an illustration of OSSAIN trigger table dependencies.

CT4Q and trigger profile index—table CT4QNAMS

The initial trigger profile index is located in table CT4QNAMS. The selector field ITRIGIDX optionally specifies the trigger profile assigned to a call during the initial call setup. The ITRIGIDX field applies to all CT4Qs datafilled in this table, regardless of whether the call is datafilled to receive TOPS or OSSAIN processing.

Although the CT4Q is changed in subsequent assignments, the trigger profile index assigned to the call remains unchanged. The trigger index specified by the ITRIGIDX field only applies to the initial call setup.

The field SYSAREA specifies unique data refinements of the CT4Q names for TOPS and OSSAIN. The SYSAREA field contains a SYSTEM selector subfield with two possible values: TOPSOPR or OSSAIN. When set to TOPSOPR the CT4QDISP, AUTOORDB, OAINPRE, and TAUDIO subfields also require datafill.

The following figure shows sample tuples in table CT4QNAMS.

Figure 26 MAP display example for table CT4QNAMS

```
CT4QNUM  CT4QNAME       NOAMA    ITRIGIDX   SYSAREA
-------------------------------------------------------------------
1        CAMA            N        N         TOPSOPR Y CAMA     N  N  N
2        1_PLUS          N        N         TOPSOPR Y 1+_Q     N  N  N
6        0_MINUS         N        N         TOPSOPR Y 0_MIN_Q  N  N  N
10       0_PLUS          N        N         TOPSOPR Y 0+_Q     N  N  N
14       DA_411          N        N         TOPSOPR Y DA_411_Q N  N  N
620      0_PLUS_OSSAIN   N         Y  0    OSSAIN
621      0_MINUS_OSSAIN  N         Y  0    OSSAIN
```

Note: For details on table CT4QNAMS, please refer to the *Customer Data Schema Reference Manual.*

Control list name—table OAINCTLA

The second main piece of data associated with a call is the control list name. When the SYSAREA selector in table CT4QNAMS is set to OSSAIN, table OAINCTLA requires datafill to specify the control list name as follows:

- The CT4Q field is the key to the table and specifies the value in the CT4QNAME field in table CT4QNAMS, which must be datafilled first.
- The OALISTNM field identifies a particular control list and is used by the OSSAIN transitions capability. See “Transitions” on page 103, for more information.

Note: Once assigned to a call, the control list and trigger profile indexes are maintained on an individual call basis and can be modified by the SN or operator terminal during the call control PIC.

The following figure shows two sample tuples in table OAINCTLA.

Figure 27 MAP display example for table OAINCTLA

```
CT4Q                          OALISTNM
-------------------------------------------------------------------
0+YELLOW                      YEL_PAGE
OSS_DA                        DA_NODE
0_PLUS_OSSAIN                 0_PLUS_C
0_MINUS_OSSAIN                0_MINUS_C
```

Final call queue assignment

Final call queue assignment is made after assigning the CT4Q to a call, as follows:

- For calls routed to TOPS, table TQMSFCQA (TOPS QMS Final Call Queue Assignment), indexed by the CT4Q, specifies the final call queue.

  Note: A TOPS call that meets the criteria for OSSAIN preprocessing uses the final CT4Q assignment as an index into table OAINPRE (OSSAIN Preprocessing). For details, refer to Chapter 4: “OSSAIN enhancements.”.

- For calls routed to OSSAIN, table OACTLDEF (OSSAIN Control List Definition) specifies the control list used for processing. Then call queue assignment, if required, is specified in table OAFUNDEF.

  Note: See “Transitions” on page 103, for information on table OACTLDEF. See “Routing and queuing” on page 55, for information on table OAFUNDEF.

QCALL utility

The QCALL Command Interpreter (CI) utility provides the ability to refine the CT4Q, using the QMS criteria tables, for a hypothetical call. OSSAIN trigger capability enhances the QCALL utility to display the system (TOPSOPR or OSSAIN), datafilled in table CT4QNAMS.

When the system is TOPSOPR, the QMS call queue from table TQMSFCQA is displayed. When the system is OSSAIN, the OSSAIN control list name from table OAINCTLA is displayed.

Note: Refer to Chapter 10: “OSSAIN maintenance,” for more information on the QCALL utility.

Call floated TDP and datafill

The call floated TDP occurs when the SN, operator, or automated system releases from the call to establish a connection between the parties involved in the call.

Note 1: The SN may decide to establish a connection between the parties *without* releasing from the call. In this scenario the call remains in the call control PIC and does not transition to the call floated PIC. DMS switch trigger event processing is not in effect, with the DMS switch forwarding all DMS switch-detected events to the SN for handling.

Note 2: The call floated TDP applies to the Automated Calling Card System (ACCS), sometimes also referred to as MCCS (Mechanized Calling Card System). Call floated trigger processing *is not supported* by the Automated Coin Toll Service (ACTS) automated system because of interactions with coin recalls during the talking and disconnect points in call. The Automated Alternate Billing System (AABS) is no longer supported in TOPS or OSSAIN.

Floating a call

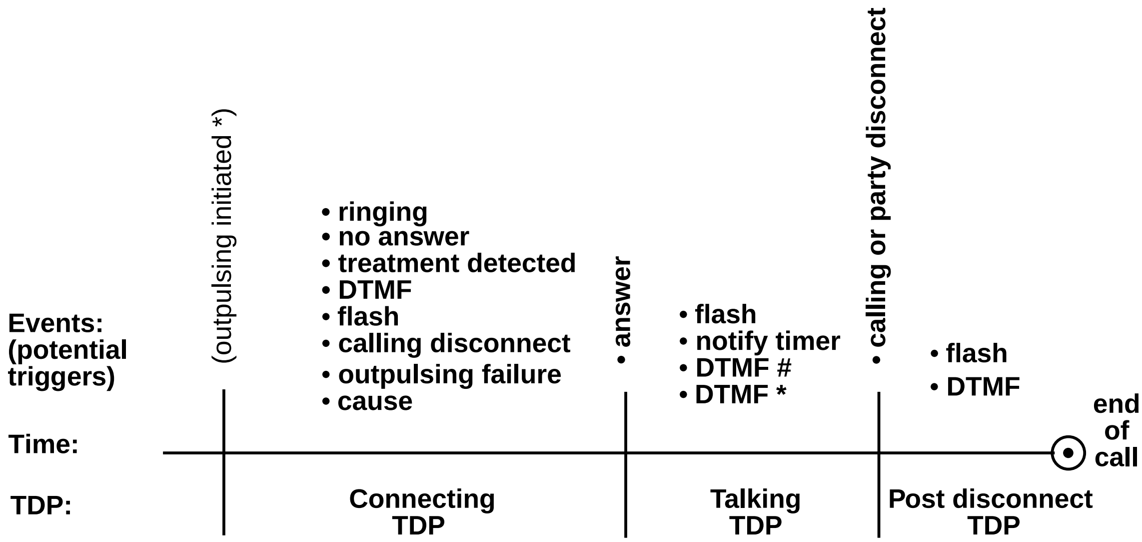
Floating a call is efficient because it frees SN resources while the call continues under switch control. For example, suppose a subscriber uses a voice link for 30 minutes. Because the SN does not need to remain in control at this point in the call, its voice link resources can be freed.

The switch performs many tasks associated with floating a call. These tasks are distributed over the following call floated TDPs:

- Connecting, where the call begins outpulsing to a forward party and is waiting for an answer.
- Talking, where the parties are talking and the call is waiting for disconnect.
- Post disconnect, where one or more parties have gone on hook.

The following figure shows the three call floated TDPs and their trigger events.

Figure 28 Call floated TDPs and trigger events



* Outpulsing initiated is the event that marks the beginning of float triggering. However, it is not a potential trigger itself.

The answer trigger event is special because it causes trigger processing to transition from the connecting TDP to the talking TDP. Similarly, a disconnect from either party causes trigger processing to transition from the talking TDP to the post disconnect TDP.

Refloat on failure

Trigger processing allows a call to be refloated if it fails to obtain a session with an SN as a result of a trigger event. Also, a call can be refloated if it fails to trigger to an operator due to a deflection or overflow. When refloating is enabled (datafilled in various trigger tables), disposition routing (datafilled in table OAFNDISP) is overridden.

Note: This capability is relevant only when control is passed to the SN or operator. Refer to Chapter 7: “OSSAIN data schema” for details.

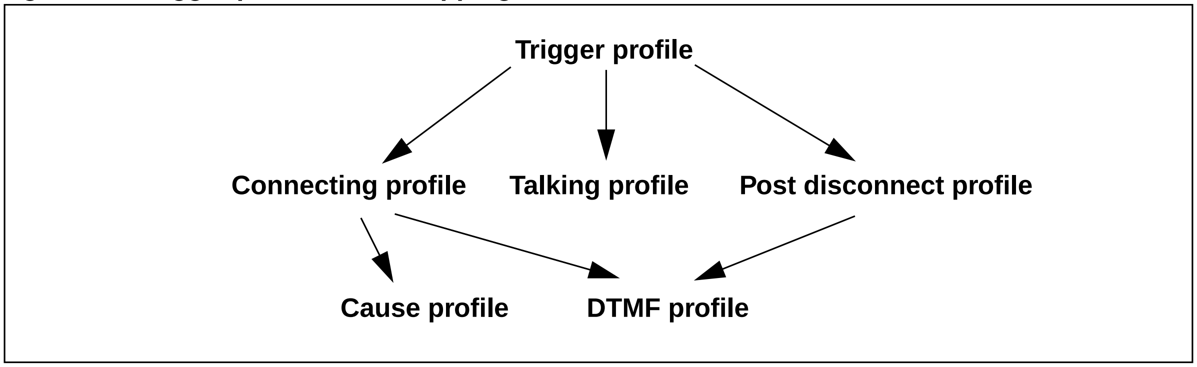
Call trigger profile index mapping

The call trigger profile index assigned during the initial call setup TDP (and optionally updated during the call control PIC) is used during the call floated TDP to access call floated trigger datafill. The call trigger profile is a mechanism that allows operating companies to control trigger processing on an individual call basis.

The call trigger profile index maps to three additional indexes, one for each of the three TDPs shown in the previous figure. The following figure shows the mapping.

Note: The cause and DTMF profiles are specified separately to support handling of multiple events.

Figure 29 Trigger profile index mapping



Call floated trigger profile datafill tables

For a call to be brought back to an SN during the call floated PIC, datafill for the call floated TDP is used. This subsection discusses the DMS switch tables that define profile indexes and their associated trigger events and actions.

Six tables handle call floated trigger processing, as follows:

- OACNNPRF (OSSAIN Connecting Profile) is used to datafill trigger profiles applicable to the connecting TDP.
- OATLKPRF (OSSAIN Talking Profile) is used to datafill trigger profiles applicable to the talking TDP.
- OADSCPRF (OSSAIN Disconnect Profile) is used to datafill trigger profiles applicable to the post disconnect TDP.
- OATPRFIX (OSSAIN Profile Index) maps the call trigger profile index to the connecting/talking/post disconnect profile indexes.
- OADTFPRF (OSSAIN DTMF Profile) is used to datafill DTMF profiles used by the connecting and disconnect profile tables.
- OACAUPRF (OSSAIN Cause Profile) is used to datafill ISUP cause profiles used by the connecting profile table.

Note: Not all triggers may be supported by an SN application. Datafill must be coordinated between the switch and SN.

A description of each table is provided in this subsection.

Table OACNNPRF

Table OACNNPRF allows the operating company to create connecting profiles by associating lists of trigger events and actions with a connecting profile index. The connecting profile index is then associated with a trigger profile index. (See “Table OATPRFIX” on page 94.)

When a DTMF profile is specified in table OACNNPRF, the DTMF digits that are input by the calling subscriber initiate a trigger at the connecting TDP. The actions associated with the DTMF index require datafill in table OADTFPRF. (See “Table OADTFPRF” on page 94.)

When a cause profile is specified in table OACNNPRF, the ISUP cause codes returned to the DMS switch can initiate a trigger at the connecting TDP. The actions associated with the cause index require datafill in table OACAUPRF. (See “Table OACAUPRF” on page 95.)

The following table lists the valid trigger events for table OACNNPRF, along with a description of each event, OSSAIN processing for the event, and normal processing for the event. OSSAIN processing occurs for the trigger event when it is datafilled in table OACNNPRF. Normal processing occurs for the trigger event when it is *not* datafilled in table OACNNPRF.

Table 4 Connection trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger hit processing	Trigger not hit processing
Treatment	TRMT	The call receives a local DMS switch treatment for conditions such as all trunks busy or vacant code (see Note 1).	The call is routed to an OSSAIN SN or TOPS operator.	The call receives a DMS switch treatment.
Cause	n/a	The call has terminated to an ISUP trunk (see Note 2). If the terminating office cannot successfully connect the call, it sends the TOPS switch an ISUP release message with the failure reason. This failure reason is also called a cause code.	The call is routed to an OSSAIN SN or TOPS operator based on the cause code received. (See table OACAUPRF.)	The call is routed to an appropriate treatment for that cause code.
Outpulsing failure	OPLSF	Outpulsing may fail due to glare or an inoperable trunk in the terminating office. To account for these conditions, the DMS switch automatically tries again to route the call to the terminating party. This event occurs after the second attempt fails.	The call is routed to an OSSAIN SN or TOPS operator.	The call is taken down.

Table 4 Connection trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger hit processing	Trigger not hit processing
No answer	NOANS (see Note 3)	The call is not answered. No answer occurs when the no answer timer expires before answer. The no answer timer is only started if the no answer event is datafilled in Table OACNNPRF. The no answer timer is started upon successful outpulsing of all digits on an MF or DP trunk. The no answer timer is started upon receiving an Address Complete Message (ACM) on an ISUP trunks.	The call is routed to an OSSAIN SN or TOPS operator.	Not applicable. The no answer timer is not started. Eventually the calling party goes on hook.
Ringing	RING	The call has terminated to an ISUP trunk. The terminating office indicates ringing by sending an ACM message, with the called party status field set to “Subscriber Free” or “Connect When Free” back to the TOPS office.	The call is routed to an OSSAIN SN or TOPS operator.	Ringing supplied to the originator by the terminating office.
DTMF	n/a	A DTMF receiver is connected and DTMF digits are input by the calling subscriber.	The call is routed to an OSSAIN SN or TOPS operator based on the DTMF digits received. (See table OADTFPRF.)	DTMF input is ignored.
Flash	FLASH	The calling party hook flashes.	The call is routed to an OSSAIN SN or TOPS operator.	Ignored. ACTS calls may route to an operator for flash during the initial period.
Calling disconnect	CLGD	The calling party goes on hook.	The call is routed to an OSSAIN SN or TOPS operator.	The call is taken down. Calls may be routed to ACTS or an operator for time and charges.

Table 4 Connection trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger hit processing	Trigger not hit processing
Answer	ANS	The terminating party answers.	Conversation begins. Connection time stamp taken. The call is routed to an OSSAIN SN or TOPS operator.	Conversation begins. Connection time stamp taken.

Note 1: Calls floating to treatment cannot then trigger to treatment.

Note 2: The cause code only applies to calls terminating to ISUP trunks.

Note 3: The no answer time applied for the no answer event is initially specified by the NOANS_TIMER parameter in table OAINPARM. This information is stored in per call data and may be altered by the SN through the OAP protocol.

In the PROFILE field, table OACNNPRF uses an action selector field—where CTRLLIST specifies a control list and FUNCTION specifies a function—to indicate the type of action that follows. Trigger events and profile actions are datafilled as: <trigger event> <SN control> <refloat on failure> <action selector> <action>.

The DTMF and cause codes profile fields are indexes to their respective tables, OADTFPRF and OACAUPRF.

The following figure shows two tuples in table OACNNPRF. Calls using index 0 would perform no trigger event processing at the connecting TDP because no triggers are datafilled. Calls using index 1 would perform trigger event processing for trigger events flash and no answer, using DTMF profile 3 and cause profile 2.

Figure 30 MAP display example for table OACNNPRF

```
CONNIDX    PROFILE                                          DTMFPRF CAUSEPRF
-------------------------------------------------------------------------
0          $                                                    N    N
1 (FLASH Y Y FUNCTION BRANDING) (NOANS Y N CTRLLIST YEL_PAGE)$   Y 3   Y 2
```

With index 1, flash would cause the call to be routed to the branding function, where BRANDING is datafilled as a function in table OAFUNDEF. No answer would cause the call to be routed to the Yellow Pages control list, where YELLOW_PAGES is datafilled in table OACTLDEF. DTMF entry would be processed as profile 3 and receipt of ISUP cause codes would be processed as profile 2.

Table OATLKPRF

Table OATLKPRF allows the operating company to create talking profiles by associating lists of trigger events and actions with a talking profile index. The talking profile index is then associated with a profile index in table OATPRFIX.

The following table lists the valid trigger events for table OATLKPRF, along with a description of each event, OSSAIN processing for the event, and normal processing for the event.

Note: The DMS switch does not monitor for DTMF events during the talking state, because detection and interpretation of such events may conflict with the exchange of DTMF information by the talking parties.

Table 5 Talking trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger hit processing	Trigger not hit processing
Calling disconnect	CLGD	The calling party goes on hook.	The call is routed to an OSSAIN SN or TOPS operator.	Trigger processing transitions to post disconnect processing.
Party disconnect	PTYD	Any party, other than the calling party, that goes on hook.	The call is routed to an OSSAIN SN or TOPS operator.	Trigger processing transitions to post disconnect processing.
Flash	FLASH	The calling party hook flashes.	The call is routed to an OSSAIN SN or TOPS operator.	Ignored.
DTMF * or DTMF #	n/a	A specialized tone receiver (STR) is connected and a DTMF '*' or '#' digit is input by either the calling or the called subscriber (see Note 1).	The call is routed to an OSSAIN SN or TOPS operator based on the DTMF digits received.	DTMF input is ignored.

Table 5 Talking trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger hit processing	Trigger not hit processing
Notify	NTFY	The notification event occurs when the notification timer expires (see Note 2).	The call is routed to an OSSAIN SN or TOPS operator.	Notification timers started by ACTS for coin notification take precedence over those specified by table OATLKPRF.

Note 1: The party (calling or called) to which STR trigger supervision is applied is specified in the STR_PARTY parameter in table OAINPARM. For information on OAINPARM, refer to Chapter 7: "OSSAIN data schema."

Note 2: The notification time applied for the notification event is initially specified by the NOTIFY_TIMER parameter in table OAINPARM. This information is stored in per call data and may be altered by the SN through the OAP protocol. For more information on the notification timer, refer to page 102.

In the PROFILE field, table OATLKPRF uses an action selector field—where CTRLLIST specifies a control list and FUNCTION specifies a function—to indicate the type of action that follows. Trigger events and profile actions are datafilled as: <trigger event> <SN control> <stop conversation> <gen ama> <release forward> <refloat on failure> <action selector> <action>.

The STAR and OCTO digit trigger event fields indicate whether to stop billable conversation timing, generate AMA, and release the forward party. The fields are datafilled as: digitsel <stop conversation> <gen ama> <release forward> <refloat on failure> <action selector> <action>.

The HOLDRCVR field is a Y or N value indicating whether or not to have a DTMF receiver connected to the calling subscriber when the call transitions to an SN.

The following figure shows two tuples in table OATLKPRF. Calls using index 0 would perform no trigger event processing because no triggers are datafilled. Calls using index 1 would perform trigger event processing for the following trigger events: flash, notify, and DTMF *.

Figure 31 MAP display example for table OATLKPRF

```
TALKIDX      PROFILE
                   STAR                            OCTO        HOLDRCVR
----------------------------------------------------------------------
0            $
                    N                               N           N
1   (FLASH  Y  N  N  N  Y  FUNCTION  PREPAY_NODE)(NTFY  Y  N  N  N  Y  FUNCTION
PREPAY_NODE) $      Y N Y Y N CALLING FUNCTION BRANDING N           Y
2 (PTYD Y N N N Y CTRLLIST ROUTE_NODE)  $
                    Y N Y Y N CALLING CTRLLIST ROUTE_NODE  N        N
```

Table OADSCPRF

Table OADSCPRF allows the operating company to create post disconnect profiles by associating lists of trigger events and actions with a disconnect profile index. The disconnect profile index is then associated with a profile index in table OATPRFIX.

When a DTMF profile is specified in table OADSCPRF, the DMS switch connects a DTMF receiver to the calling party. This allows select DTMF input to initiate triggering at the disconnect TDP. The actions associated with the DTMF index require datafill in table OADTFPRF.

The following table lists the valid trigger events for table OADSCPRF, along with a description of each event, OSSAIN processing for the event, and normal processing for the event.

Table 6 Post disconnect trigger event descriptions and processing

Trigger event	Trigger mnemonic	Description	Trigger processing	Trigger not hit processing
Calling disconnect	CLGD	The calling party goes on hook.	The call is routed to an OSSAIN SN or TOPS operator.	The call is taken down.
Flash	FLASH	The calling party hook flashes.	The call is routed to an OSSAIN SN or TOPS operator.	Ignored.
DTMF	n/a	A DTMF receiver is connected and DTMF digits are input by the calling subscriber. The DTMF receiver is attached to the calling party after all parties in the call have disconnected.	The call is routed to an OSSAIN SN or TOPS operator.	Ignored.

The following figure shows two tuples in table OADSCPRF. Calls using index 0 would perform no trigger event processing because no triggers are datafilled. Calls using index 1 would perform trigger event processing for the following trigger events: calling disconnect, party disconnect, and DTMF profile 4.

Figure 32 MAP display example for table OADSCPRF

```
DISCIDX          PROFILE            DTMFPRF
-----------------------------------------------------------------
0                $                  N
1      (CLGD Y FUNCTION DEBIT)  (FLASH Y FUNCTION DEBIT) $   Y 4
```

Table OATPRFIX

Table OATPRFIX allows the operating company to combine the profile indexes from tables OACNNPRF, OADSCPRF, and OATLKPRF into a single call trigger profile index. This combination allows for ease in assigning the initial call trigger profile index in table CT4QNAMS. It also allows the SN to update the call floated profile indexes through a single call trigger profile index in the OAP protocol.

The following figure shows two tuples in table OATPRFIX. Calls using a call trigger profile index of 0 would access the connecting, talking, and disconnect profiles of 4, 2, and 7 respectively. Likewise, calls using call trigger profile index 1 would access profiles 2, 2, and 3.

Figure 33 MAP display example for table OATPRFIX

```
IDX             CPROFIDX     TPROFIDX     DPROFIDX
----------------------------------------------------------------
0               Y 4            Y 2           Y 7
1               Y 2            Y 2           Y 3
```

Table OADTFPRF

Table OADTFPRF allows the operating company to assemble profiles consisting of a list of DTMF digits, generate AMA, and release forward actions, and a control list or function action. As stated previously, table OADTFPRF is accessed through the OACNNPRF and OADSCPRF tables.

In the PROFILE field, the range of DTMF digits that can be entered are 0 to 9, STAR, and OCTO. The actions associated with each DTMF digit include generate AMA, release forward party, refloat on failure, and route to a function or control list name.

Entry in this field is of the form: digit <gen ama> <release forward> <refloat on failure> <action selector> <action>. When generate AMA is set to Y, an AMA record is produced as part of trigger processing. When release forward is set to Y, the forward party is released as part of trigger processing.

The HOLDRCVR field is a Y or N value indicating whether or not to retain the DTMF receiver linked to the calling party when the call transitions to an SN.

The following table shows two tuples in table OADTFPRF.

Figure 34 MAP display example for table OADTFPRF

```
DTMFIDX         PROFILE                                         HOLDRCVR
---------------------------------------------------------------------
3   (0 N N Y FUNCTION BRANDING) (STAR N N Y FUNCTION ADS_NODE) $  N
4   (0 N N Y FUNCTION YELLOW_PAGES) (1 N N N CTRLLIST YEL_PAGE)
(OCTO N N N FUNCTION SEQUENCE) $ Y
```

Table OACAUPRF

Table OACAUPRF allows the operating company to assemble profiles of ISUP cause codes and corresponding actions. Table OACAUPRF is accessed through the OACNNPRF table.

Refer to the following lists for information on ISUP release classes and cause codes. The range of ISUP cause codes is 1 to 127. Note that there are gaps of unused cause codes within classes and between classes.

Release classes

- NORM - Normal Event Class - Cause codes 1-31
- CL2 - Resource Unavailable Class - Cause codes 34-47
- CL3 - Service Option Not Available Class - Cause codes 52-63
- CL4 - Service Option Not Implemented Class - Cause codes 65-79
- CL5 - Invalid Message Class - Cause codes 81-95
- CL6 - Protocol Error Class - Cause codes 97-111
- CL7 - Interworking Class - Cause code 127

NORM cause codes

- 1 - Unallocated Number
- 2 - No Route to Transit Network
- 3 - No Route to Destination
- 4 - Send Special Info Tone
- 5 - Misdialed Trunk Prefix
- 16 - Normal Clearing

- 17 - User Busy
- 18 - No User Responding
- 19 - No Answer From User
- 21 - Call Rejected
- 22 - Number Changed
- 25 - Translations Fail
- 26 - Call Returns
- 27 - Destination out of Service
- 28 - Address Incomplete
- 29 - Facility Rejected
- 30 - Apply Locally
- 31 - Normal Unspecified

Note: This list contains the current known ISUP release cause codes. However, any release cause in the range one to 31 can be datafilled.

When datafilling table OACAUPRF, the cause values falling within the Normal Event Class (Class1) must be datafilled individually. In the PROFILE field, the value NORM is followed by a cause value in the range 1 to 31. Datafill is of the form: NORM <cause values> <refloat on failure> <action selector> <action>.

Cause values for Class2 through Class7 must be datafilled as a class. This is done by making the class a selector field in the profile. Datafilling Class2 through Class7 does not allow a cause value to be entered. Datafill is of the form: <classn> <refloat on failure> <action selector> <action>.

The action for each cause value is a function or control list name.

The following figure shows a tuple in table OACAUPRF. Calls using cause profile index 2 would route calls to the message delivery service node for cause codes 17 and 19 in Class1 and for all cause codes in Class2.

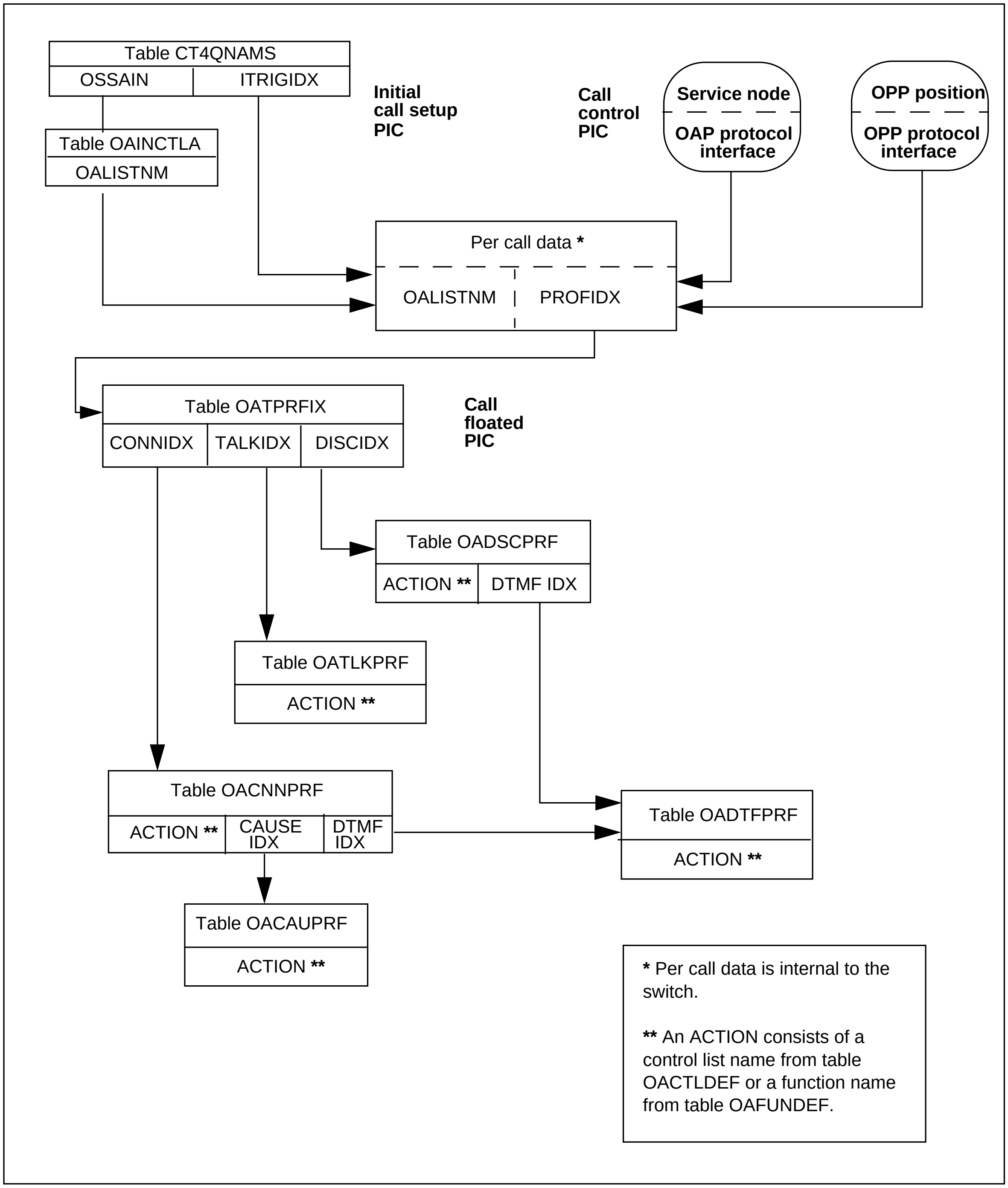
Figure 35 MAP display example for table OACAUPRF

```
IDX       PROFILE
-------------------------------------------------------------------
2        (NORM 17 Y FUNCTION ADS_NODE) (NORM 19  N FUNCTION ADS_NODE)
   (CL2 Y FUNCTION ADS_NODE) $
```

Trigger table relationships

The following figure illustrates the relationships among the trigger profile tables. The figure also shows at which PIC the data is accessed. A discussion follows the figure.

Figure 36 Table relationships for OSSAIN triggers



Initial call setup PIC

During the initial call setup PIC, table CT4QNAMS is accessed to obtain the initial trigger profile index for the call and to determine if the CT4Q assigned to the call required OSSAIN processing. If OSSAIN processing is required, table OAINCTLA is accessed to obtain the initial control list for the call.

This information is stored in per call data. The control list name is next used by the OSSAIN transitions functionality. (See “Transitions” on page 103.)

Call control PIC

Once a session to an SN or operator begins (during the call control PIC), the SN or operator can modify the control list name or profile index for the call.

Call floated PIC

During the call floated PIC, the PROFIDX retrieved from per call data is used to access table OATPRFIX with indexes to the other call floated trigger tables.

Trigger table datafill order

The ten tables shown in Figure 36 should be datafilled in the following order:

1 Table CT4QNAMS (for CT4Q name) is used by table OAFUNDEF.

   Note: A trigger profile index datafilled in table CT4QNAMS should also be datafilled in table OATPRFIX. If not, a log will be generated for calls using that CT4Q.

2 Tables OAFUNDEF and OACTLDEF (for function name and control list name, respectively) are used by the profile tables.

   Note: See “Transitions” on page 103, for information on table OACTLDEF and defining control lists.

3 Tables OADTFPRF and OACAUPRF (if required) set up the DTMF profile and cause profile, respectively.

4 Tables OACNNPRF, OATLKPRF, and OADSCPRF (for connecting, talking, and disconnect profiles, respectively) use the previously defined DTMF and cause profile indexes and function and control list names.

5 Table OATPRFIX (for profile indexes) uses the previously defined connecting, talking, and disconnect indexes.

6 Table OAINCTLA uses the previously defined control list name.

Trigger call flow scenarios

This section provides the following sample call flow scenarios using trigger profiles:

- Yellow Pages call completion with message delivery service
- Debit card

Yellow Pages call completion with message delivery service

In this scenario, the subscriber dials the Yellow Pages service with call completion capability. At initial call setup PIC, the DMS switch assigns a trigger profile index of 0 from table CT4QNAM. Profiles 0 and 6 are set up as follows.

Profile 0 points to connecting, talking, and disconnect profiles with no triggers datafilled (shown with a $ sign):

- Profile 0
  - Connecting profile $
  - Talking profile $
  - Disconnect profile $

Profile 6 points to a connecting profile with message delivery service triggers:

- Profile 6
  - Connecting profile
    - NOANS - ADS
    - CAUSE User Busy - ADS
    - DTMF * - ADS
  - Talking profile $
  - Disconnect profile $

The Yellow Pages call flows as follows:

1. The call arrives at the Yellow Pages SN with a profile of 0.
2. The SN offers the subscriber call completion.
3. The subscriber accepts.
4. The SN decides to allow the subscriber access to an ADS node if there is no answer or busy at the terminating party.
5. The SN sends a call float (through OAP messaging) that changes the profile index to 6.
6. The subscriber is now automatically routed to ADS if no answer or busy is detected. ADS can also be accessed by the subscriber keying a *.

Debit card

In this scenario, a subscriber makes a call to be billed to a debit card. The call arrives at the DMS switch and is assigned a profile index of 0. Profiles 0 and 8 are set up as follows.

Profile 0 points to connecting, talking, and disconnect profiles with no triggers datafilled:

- Profile 0
  - — Connecting profile $
  - — Talking profile $
  - — Disconnect profile $

Profile 8 points to profiles with debit card triggers:

- Profile 8
  - — Connecting profile $
    - – CLGD - DEBIT
  - — Talking profile
    - – NTFY - DEBIT
    - – CLGD - DEBIT
    - – PTYD - DEBIT
  - — Disconnect profile
    - – CLGD - DEBIT

The debit card call flows as follows:

1 The call arrives at the debit billing SN with a profile of 0.

2 The SN determines that the subscriber has only 5 minutes of conversation time remaining on the debit card.

3 The SN sends a call float (through OAP messaging) that changes the profile index to 8 and the notification timer to 5 minutes.

4 Upon answer, the call is floated and enters the talking state.

5 The DMS switch starts the 5 minute notification timer.

6 If the timer expires, the SN is brought back into the call, debits the card, informs the subscriber, and discontinues the call.

Note: Alternatively, either party could disconnect before the expiration of the 5 minute notification timer. In this case, the SN would be brought back into the call to debit the card.

Float processing with TOPS operator and automated systems

As previously stated, the call floated TDP applies to TOPS calls handled by an SN, operator, or TOPS automated system. The only TOPS automated system that supports float trigger processing is MCCS/ACCS. The AABS automated system is no longer supported, and the ACTS automated system does not support OSSAIN trigger processing because of interactions with coin recalls during the talking and disconnect points in call.

TOPS automated systems support a predefined set of call floated trigger events without OSSAIN. The following table lists the set of trigger events, the TOPS system, associated processing, and interactions with trigger datafill.

Table 7 Float trigger event interactions with TOPS automated systems

Trigger event	TOPS system	Processing	Interactions
Flash	ACTS	Route call to an operator	None. Calls floated by ACTS are routed to an operator. OSSAIN trigger processing is not supported for ACTS.
Notify	ACTS	Coin notification	None. OSSAIN trigger processing is not supported for ACTS.
#	MCCS/ ACCS	Sequence processing	Calls floated by the ACCS automated system use DTMF digit # to initiate a sequence call. Digit # datafilled in table OADTFPRF overrides this processing and routes the call to an OSSAIN SN.
*	MCCS/ ACCS	Route call to ADS	Calls floated by the ACCS automated system used DTMF digit * to initiate pre-OSSAIN AudioGram Delivery Service (ADS). Digit * datafilled in table OADTFPRF overrides this processing and routes the call to an OSSAIN SN.
Disconnect	Operator	Route call to ACTS or operator for Time and Charges	OSSAIN trigger processing is not supported for calls requiring Time and Charges.

With OSSAIN, trigger processing is extended to provide OSSAIN service access, through the call floated triggers, to calls that were originally handled and floated by an operator or TOPS automated system. This allows for consistent service offering across all call types.

As noted in the preceding table, OSSAIN trigger processing takes precedence over pre-OSSAIN trigger processing. Pre-OSSAIN trigger processing can still be achieved by not datafilling the trigger event against the OSSAIN trigger profile.

Trigger processing parameters

All TOPS and OSSAIN calls are assigned a trigger profile index if datafilled in table CT4QNAMS.

Three parameters for trigger processing are defined in table OAINPARM (OSSAIN Parameter). These parameters determine the following: the number of triggers allowed for a single call, the timing value that applies to the no answer trigger event, and the timing value that applies to the notification trigger event. The timing parameters are in effect unless SN or operator actions override them.

A description of these parameters follows. For information on values for these parameters, refer to Chapter 7: “OSSAIN data schema.”

Parameter MAX_NUM_TRIGGERS

MAX_NUM_TRIGGERS specifies the number of times a call is allowed to trigger while in the call floated PIC.

Parameter NOANS_TIMER

NOANS_TIMER specifies the amount of time that transpires while a call is waiting for answer before the no answer trigger occurs. This information is stored in per call data and can be changed by the SN through the OAP protocol.

Parameter NOTIFY_TIMER

NOTIFY_TIMER specifies the amount of time a call spends in the talking state before the notification trigger event occurs. This information is stored in per call data and can be changed by the SN through OAP messaging.

Notification timer

The SN can set an OAP notification timer that starts after the SN floats the call. If the float notification timer expires, the call triggers back to the SN for further processing.

The notification timer also has a *persistent* timer mode. The persistent notification timer runs *while the call has a session with an SN* and after the call is floated. If the persistent timer expires while the call has a session with the SN, the switch sends an OAP message to the SN to inform it of the timer expiration. If the persistent timer expires while the call is floated, the call triggers back to the SN for further processing (same as before).

The notification timer can be either relative or conversation-based. The relative timer runs for the specified duration. The conversation-based timer subtracts elapsed conversation time from the specified duration before starting the timer.

If the NTFY event is included in the trigger profile (table OATLKPRF), and the SN starts a notification timer explicitly before the call is floated, the timer is *reset* at float. It is reset with a timer value supplied by the SN or obtained from the NOTIFY_TIMER parameter in table OAINPARM.

Note: For details on OAP messages used for the notification timer, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Transitions

Transitions are mechanisms that allow calls to flow to and from function providers and to interact with existing systems. For example, transitions allow an SN to transfer a call to another SN, operator, or TOPS automated system.

Direct transfer to control list

With the transition mechanism, a function provider can implicitly request that the call transfer to a different function. That is, control of the call is transferred without the SN explicitly choosing a new function.

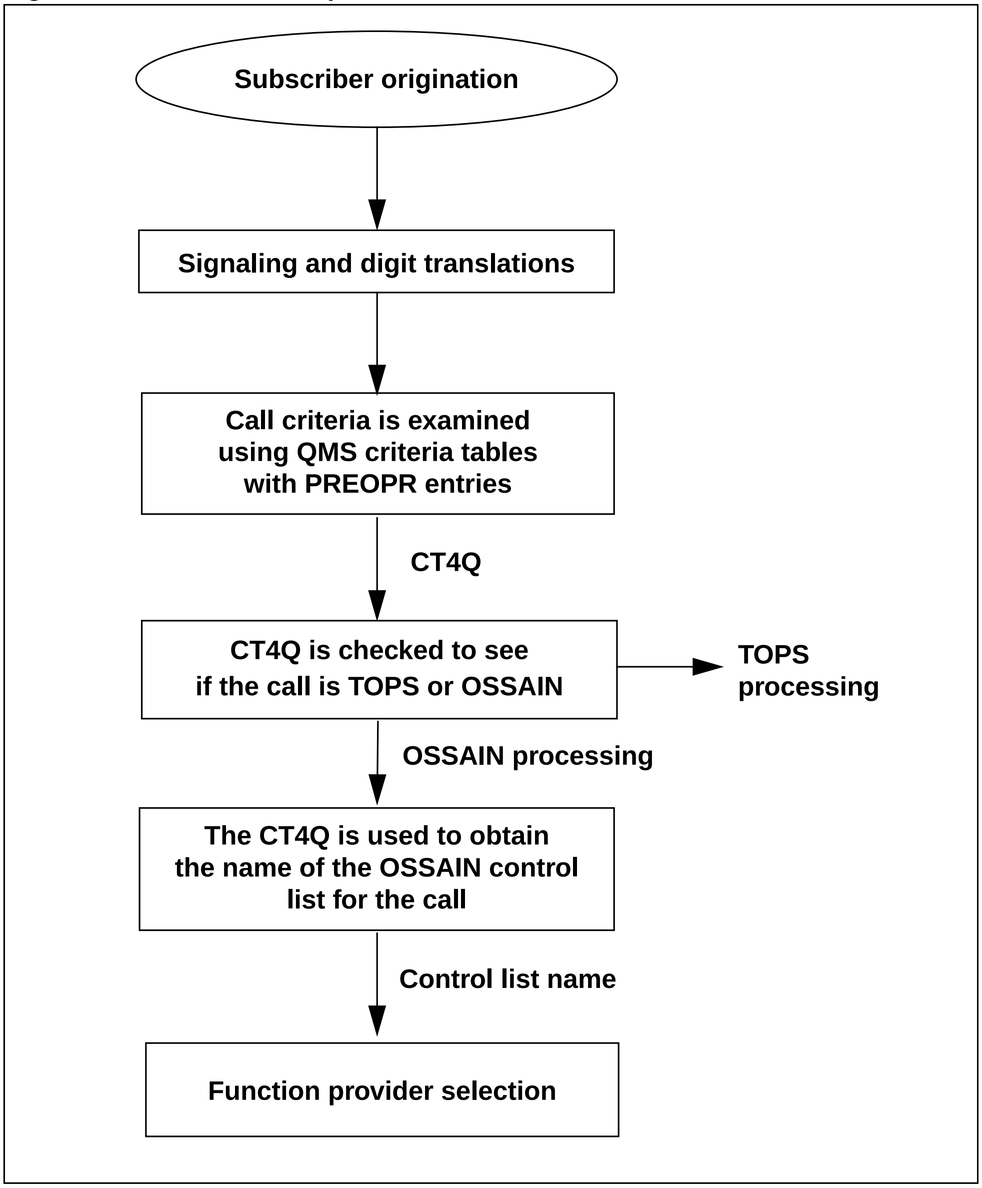
The *control list* allows this interworking of OSSAIN and other systems. As stated before, the control list contains the name of a function, such as branding, that OSSAIN applies to the call. The control list is maintained in the DMS switch.

Note: MCCS and DAS are the only existing automated system that can be datafilled in the control list. The AABS protocol is no longer supported.

Call flow

The following figure shows the basic initial call setup call flow. Transitions functionality begins at the point in processing where the CT4Q is associated with OSSAIN and is mapped to a control list.

Figure 37 Initial call setup call flow



OSSAIN transitions functionality is provided in the following two areas:

- function provider selection

  Function provider selection occurs when a new function is needed. In OSSAIN processing, requests for a new function occur after various transition mechanisms (for example, after initial call setup, as the result of a call float trigger, or at the request of an SN).

- transitions between functions

  The direct transfer to control list mechanism allows calls to move seamlessly between functions provided by SNs and functions provided by TOPS systems.

Function provider selection

The function provider selection component determines the next function provider (SN, operator, or automated system) and the function's availability to the calling subscriber. Function selection occurs for both direct transfers and control list processing.

Function provider selection involves the following two concepts:

- control list processing

  Once a control list name is obtained, the function identified in the list is processed. The identifier for that function is used to index table OAFUNDEF, which defines the actual provider of the function.

- function blocking

  Function blocking is based on the originating DN, and is performed by adding a field to table TOPSDB that indexes a table (OAFUNBLK) for including or excluding functions. The ability to block functions based on a DN allows operating companies to control a subscriber's access to certain functions.

  Note: For SN-originated calls, the initial function is not blocked; however, all subsequent functions are checked.

Function provider selection call flow and datafill

During OSSAIN call processing, the function selection state is entered by way of a function name or a control list name. Following are the points at which this state can be entered:

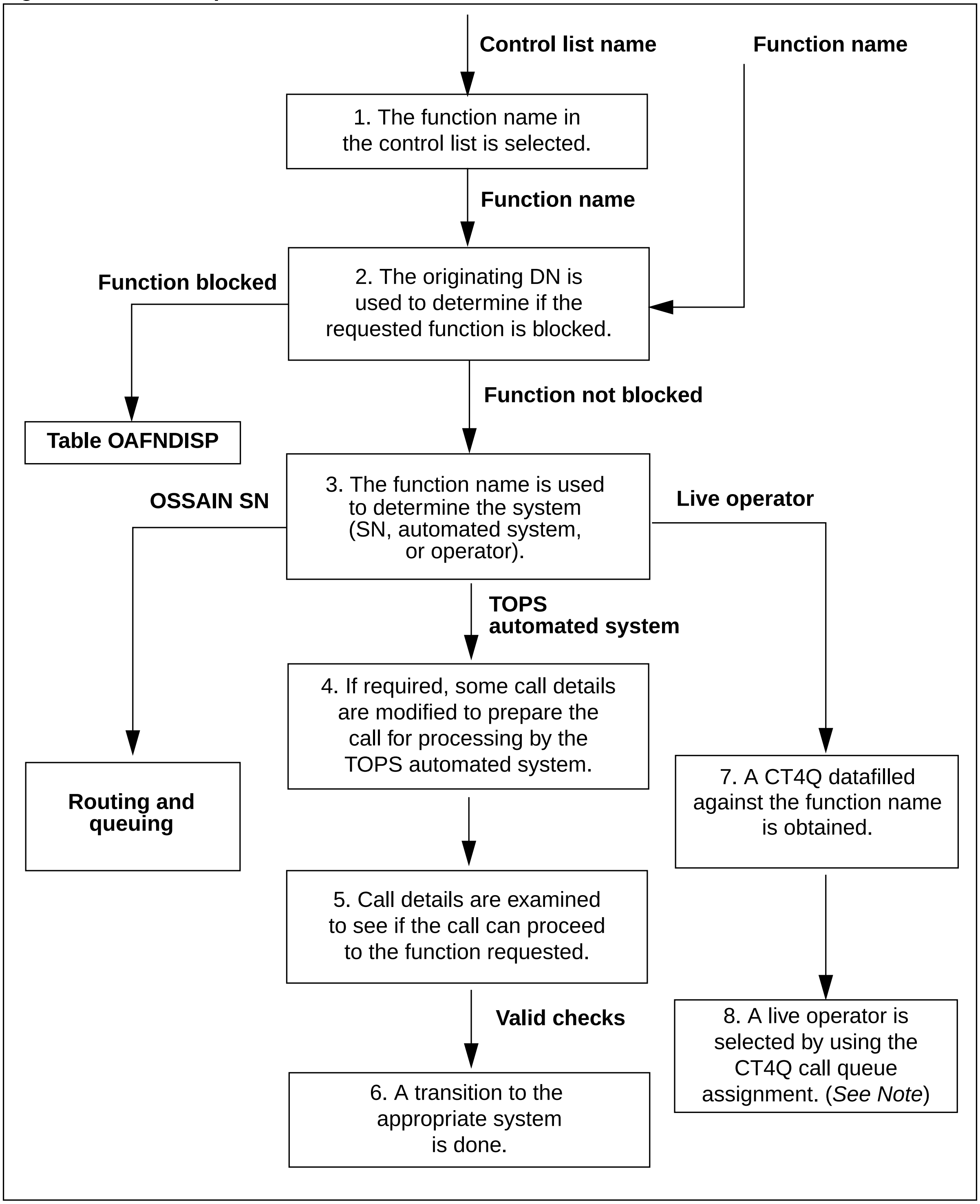
- after initial call setup, when the control list name is passed in
- at the transition from an SN, when the current SN provides a control list name
- at the transition from an operator, when the operator supplies a new CT4Q that maps to an OSSAIN control list
- after a call floated trigger, when the trigger supplies a function name or a control list name associated with the call floated trigger hit

The following six data tables apply to function provider selection:

- OACTLDEF (OSSAIN Control List Definition)
- DNSCRN (Directory Number Screen)
- TOPSDB (TOPS Database)
- OAFUNBLK (OSSAIN Function Blocking)
- OAFNDISP (OSSAIN Function Disposition)
- OAFUNDEF (OSSAIN Function Definition)

Refer to the following figure for an example of function provider selection call flow. A description and example of the table datafill for each step follows the figure.

Figure 38 Function provider selection call flow



Note: Additional CT4Q refinements can be applied to OSSAIN calls that are routed to an operator. This option affects the CT4Q assignment. For details, refer to Chapter 4: “OSSAIN enhancements.”

Step 1—table OACTLDEF

Step 1 is performed when a new control list name is provided. Table OACTLDEF defines OSSAIN control lists. Up to 4095 control lists can be datafilled in table OACTLDEF, each consisting of one function. The function must first be datafilled in table OAFUNDEF. (See “Datafilling functions and session pools” on page 66, for information on table OAFUNDEF.)

The following figure shows three tuples in table OACTLDEF.

Figure 39 MAP display example for table OACTLDEF

```
OACTLNUM    OACTLNAM       NETWRKID     OAFUNCTS
-------------------------------------------------------------------
0           0+AABS         3        (     TOPS_MCCA) $
1           DA_SYSTEM      7        (     OSSAIN_DA) $
2           ROUTER         2        (    ROUTE_NODE) $
```

Note: The NETWRKID field refers to a Network Service ID. This value is not used by the transitions component of OSSAIN. For more information, refer to Chapter 9: “OSSAIN billing.”

Step 2—tables DNSCRN, TOPSDB, OAFUNBLK, and OAFNDISP

Step 2 determines if the requested function is blocked for the originating DN. TOPS currently provides the ability to restrict DNs in table DNSCRN, which indexes table TOPSDB.

The following figure shows two tuples in table DNSCRN that point to indexes in table TOPSDB.

Figure 40 MAP display example for table DNSCRN

```
DN                        ATTROPTS
-------------------------------------------------------------
6193229801                (TOPSDB  0)
6193229900                (TOPSDB  1)
```

Note: For details on table DNSCRN, please refer to the *Customer Data Schema Reference Manual.*

Table TOPSDB has indexes to subtending tables TDBCLASS, RESTBIL, DARSTBIL, TOPSDEV, FXDNMAP, RESTAMA, and TDBDAOPT where the specific restrictions are datafilled. OSSAIN transitions use this same table structure to provide function blocking based on the originating DN.

The OAFUNBLK field in table TOPSDB is used for OSSAIN function blocking. The value in the OAFUNBLK field is used to index table OAFUNBLK.

The following figure shows tuples in table TOPSDB. In the example, a TDBKEY of 0 maps to an OAFUNBLK index of 1.

Figure 41 MAP display example for table TOPSDB

TDBKEY	TDBCLIDX	TBDAOPT	TDBNORM	TDBSERV	OAFUNBLK
0	0	0	0	0	1
1	1	0	0	0	0
2	2	0	0	0	2
3	4	0	0	0	1

Note: For details on table TOPSDB, please refer to the *Customer Data Schema Reference Manual.*

Table OAFUNBLK lists the functions that are included or excluded for a particular index (the DN indexed from tables DNSCRN and TOPSDB). The function name requires datafill first in table OAFUNDEF.

The following figure shows sample tuples in table OAFUNBLK. In the example, index 1 has the FNBLKSEL field set to INCLUDE with three function names from table OAFUNDEF listed. These values mean that the only functions allowed for this particular DN are BRANDING, NODE_AABS, and YELLOW_PAGES. All other functions are blocked.

Index 3 has the FNBLKSEL field set to EXCLUDE, which means that only the YELLOW_PAGES function is not allowed for this particular DN. All other functions are allowed.

Figure 42 MAP display example for table OAFUNBLK

IDX	FNBLKSEL	OAFUNCTS
0	EXCLUDE	$
1	INCLUDE	BRANDING NODE_AABS YELLOW_PAGES $
2	INCLUDE	TOPS_BRANDING TOPS_AABS $
3	EXCLUDE	YELLOW_PAGES $

Note: In the example, index 0 has the FNBLKSEL field set to EXCLUDE with no function names in the OAFUNCTS field. This means that no functions are excluded and all are allowed for the particular DN. This tuple is the default for table OAFUNBLK.

Table OAFNDISP determines the disposition for the call. If the function is blocked, the value in the IBLKACTN field (for initial call setup calls) or the TBLKACTN field (for transition/trigger calls) specifies the action.

The following figure shows a tuple in table OAFNDISP. In the example, the YELLOW_PAGES function is blocked for index 3 in table OAFUNBLK (page 108). So the blocking action (for both initial and transitions) is to transfer to a control list, specified as DA_SYSTEM.

Figure 43 MAP display example for table OAFNDISP

```
FUNCNAME          IDFLACTN                  IOVFACTN
          IBLKACTN              IOFLACTN
                  TDFLACTN                  TOVFACTN
          TBLKACTN              TOFLACTN
------------------------------------------------------------------------
YELLOW_PAGES      TREAT VACT                GOTOFN YELPGS_OPER
          GOTOCTL DA_SYSTEM      GOTOFN LIVE_OPER
                  TREAT VACT                GOTOFN YELPGS_OPER
          GOTOCTL DA_SYSTEM      GOTOFN LIVE_OPER
```

Step 3—table OAFUNDEF

Step 3 continues with function provider selection when the requested function is not blocked. Table OAFUNDEF defines the function name and type of function provider.

The FUNCNAME field specifies the name of the function. The FUNCAREA field contains more information on the type of function provider. The FUNCTYPE selector subfield defines one of three types of functions, as follows:

- SN (for functions provided by an SN)
- TOPSOPER (for functions provided by an operator)
- TOPSAUTO (for functions provided by an automated system)

When the function provider is an SN (FUNCTYPE = SN), OSSAIN routing and queuing is performed. See “Datafilling the QMS setup for routing and queuing” on page 69, for information on OSSAIN call queues. The base service (ORIGSERV) for the call is also datafilled in the FUNCAREA field.

When the function provider is an operator (FUNCTYPE = TOPSOPER), the OPRCT4Q subfield is datafilled to obtain an operator. Additional CT4Q refinements can be applied to OSSAIN calls that are routed to an operator. The QREFINMT and CQORDER subfields specify whether CT4Q refinements are applied and if so, which CT4Q order to use for the TOPSOPER function. (For details, refer to Chapter 4: “OSSAIN enhancements.”)

When the function provider is an automated system (FUNCTYPE = TOPSAUTO), the AUTOSYS is refined with either field OPRONFL or OPRCT4Q, QREFINMT and CQORDER. The DAS AUTOSYS is the only one which uses OPRCT4Q, QREFINMT and CQORDER to correctly setup the DAS call to determine the DA vendor. The other automated systems use the OPRONFL value as input to the QMS POSTAUTO refinement ordering if live operator backup is required for the specified automated system.

The following figure shows tuples in table OAFUNDEF.

Figure 44 MAP display example for table OAFUNDEF

```
FUNCID        FUNCNAME          FUNCAREA
--------------------------------------------------------------------
0             BRANDING          SN  TASERV  N N N Y Y CQ0 Y Y
1             YELLOW_PAGES      SN  DASERV  N N N Y Y CQ1 N
3             0_MINUS_OPER      TOPSOPER  0_MINUS Y POSTAUTO
4             TA_AUTO           TOPSAUTO  MCCS  0_PLUS
5             OSSAIN_DA         SN  INTCSERV N N N Y Y CQ3 N
6             DAS_ONLY          TOPSAUTO DA_NO_OPER N
```

Step 4

In Step 4 call details are altered by the switch, as necessary, to transition the call to the automated system. This is currently done for some operator services transitions where the primary change is the call origination type.

Step 5

In Step 5 various checks are performed to make sure the call is ready to transition to the requested TOPS automated system (AUTOSYS subfield in table OAFUNDEF). These are the same checks that are currently done for these functions. If the checks fail, the call is routed to an operator using the datafilled CT4Q in the OPRONFL subfield in table OAFUNDEF unless it is a DAS autosys. For DAS, the OPRCT4Q is refined as specified by QREFINMT and CQORDER and the final CT4Q is used to route to an operator upon failures.

Step 6

In Step 6 the call is now ready to transition to the appropriate system.

Step 7

Step 7 is entered if the requested function is provided by a live operator. The CT4Q datafilled against the function name is obtained (OPRCT4Q subfield in table OAFUNDEF).

Step 8

If QMS CT4Q refinements are not applied to the call (QREFINMT subfield = N in table OAFUNDEF), the CT4Q obtained in Step 7 is used for final call queue assignment. If additional CT4Q refinements should be applied to the call (QREFINMT subfield = Y in table OAFUNDEF), the order of refinement is determined by the CQORDER subfield in table OAFUNDEF. (For details on additional CT4Q refinements, refer to Chapter 4: “OSSAIN enhancements.”)

Transitions between functions

Transitions between functions deal primarily with passing along the associated data and text with a call as it transitions *from* SNs, automated systems, and operators. In a transition, the current SN, operator, or automated system is released and the switch is temporarily in control until it establishes the connection to the subsequent SN, operator, or automated system.

Transitions between functions involve the following concepts:

- call context block
- transition mechanisms
- transition types

Call context block

The call context block is a generic block of data that contains additional information about the call or the parties involved in the call. It is passed to the DMS switch by an SN, operator position, or DAS.

The current call context block is retained by the switch for the duration of a call, and it is passed to any SN or operator position brought into the call at initial connection. The format of the data in the call context block is unknown to the switch, and the switch never attempts to interpret the contents of the call context block. Because the switch passes the data without having service-specific information, the SNs and operators can synchronize the call context independently of the switch.

Note 1: For more information on the text-to-operator functionality, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Note 2: The switch also can pass call context information *to* the DAS in the standard DA protocol. For more information, please refer to *Standard Nortel-DMS/DAS Protocol*, Q210-1.

Transition mechanisms

Three transition mechanisms from source are available as follows:

— direct transfers from SNs—SNs are allowed to request a direct transfer to a control list. SNs accomplish the transfer by using a specific message in the OAP protocol

— direct transfers from TOPS operators—operator positions are able to directly choose the next control list for the call. This is done either by using OGT keying or the CT4Q ActID to change the CT4Q to one that routes to a new control list.

— direct transfers from a DAS—a DAS is able to request a direct transfer to a control list. This transfer is done by using a specific message in the standard DA protocol.

The following table describes the available transition mechanisms.

Table 8 Transition mechanisms (FROM source)

	Source (FROM)	**Direct transfer to control list**
SN type	OSSAIN SN	Yes
Operator type	Operator position (see Note)	Yes
Automated system	DAS	Yes

Transition types

Transitions are made by *directly* transferring the call to a new control list that contains one function. The switch checks the validity of the control list name or ID and, if valid, ends the session with the function provider. The function contained in the control list is not verified, so the transition is unconfirmed.

The following table specifies whether the transition is unconfirmed or is not allowed by the OSSAIN transition mechanism.

Table 9 Transition types for OSSAIN direct transfers

Source (FROM)	**Function provider (TO)**		
	SN	**Operator**	**Automated system**
SN	Unconfirmed	Unconfirmed	Unconfirmed
Operator position	Unconfirmed	Not allowed	Not allowed
DAS	Unconfirmed	See Note	Not allowed
Note: ADAS-to-operator transitions are not affected by DAS transitions.			

The following nine transition types are analyzed in this subsection:

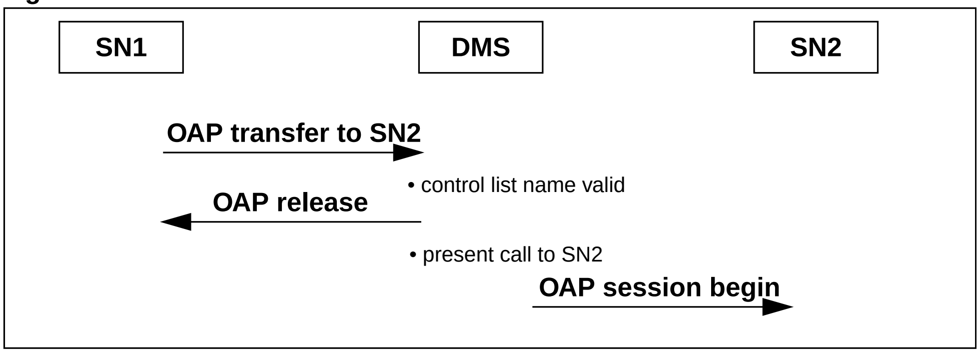
- SN to SN
- SN to operator
- SN to automated system
- Operator position to SN
- Operator position to operator
- Operator position to automated system

- DAS to SN
- DAS to operator
- DAS to automated system

SN to SN

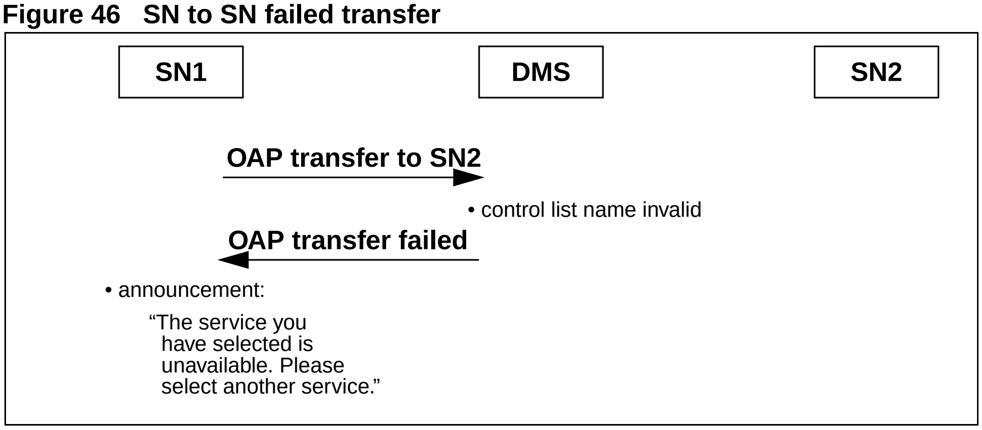
When this type of direct transfer is requested, the switch checks the validity of the control list name. The following figure shows a high-level view of the message exchange for a successful transfer.

Figure 45 SN to SN successful transfer



The following figure shows a high-level view of the message exchange for a failed transfer because of an invalid control list name.

Figure 46 SN to SN failed transfer



SN to operator

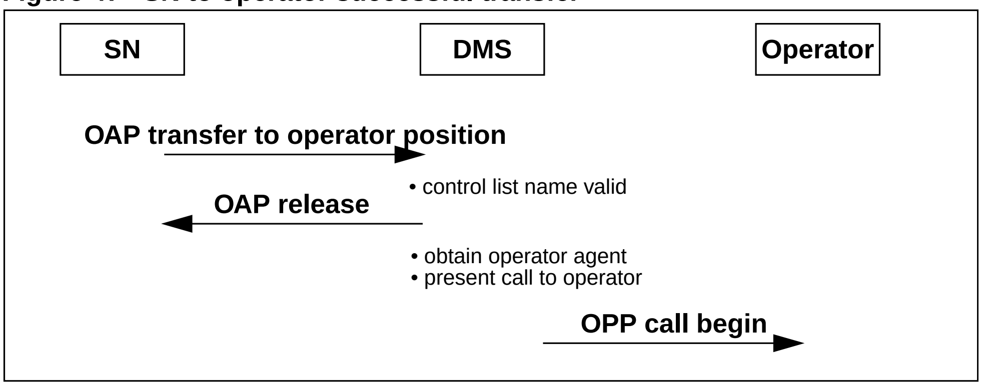
When this type of direct transfer is requested, the switch checks the validity of the control list name. If valid, the switch ends the session with the SN, regardless of the availability of the subsequent agent. The SN can update the context block when it releases itself from the call.

The OPP protocol, used between the switch and operator positions, allows the context block to be passed to the position. The position also receives the function identifier and control list identifier through an OPP message. (The identifier is the actual number of the function and the control list [field FUNCNUM in table OAFUNDEF, and field OACTLNUM in table OACTLDEF].)

Finally, when set up by an SN, a small amount of text to be displayed is passed to the operator position.

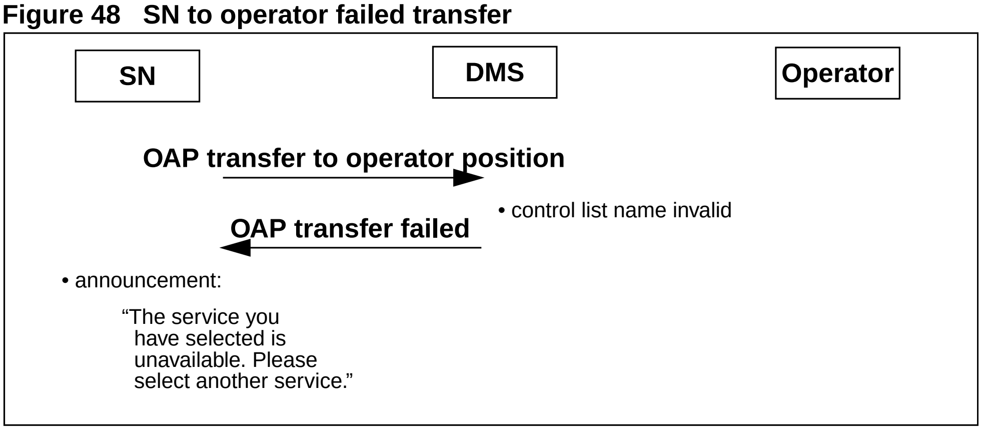
The following figure shows a high-level view of the message exchange for a successful transfer.

Figure 47 SN to operator successful transfer



The following figure shows a high-level view of the message exchange for a failed transfer because of an invalid control list name.

Figure 48 SN to operator failed transfer



SN to automated system

A call can be transitioned to a DAS automated system where the DAS is presented with the call without an operator being attached. The OAFUNDEF subfields OPRCT4Q, QREFINMT, and CQORDER determine the final CT4Q to be used in determining the service and associated DAS vendor. This is similar to a call routing to a DAS operator function, only without the operator being brought into the call unless failure occurs.

A call may also transition to MCCS/ACCS if it eligible; however, if it fails the call is routed to a TOPS operator.

When an SN requests a direct transfer to a control list, the switch checks the validity of the control list name. If valid, the switch ends the session with the SN, regardless of the call details.

If the call detail conditions are met, the switch then attempts to transfer to the automated system. If that fails, the call is routed to operator using the datafilled CT4Q in field OPRONFL in table OAFUNDEF; or, as noted above, it may be routed to MCCS/ACCS instead of AABS.

Operator position to SN

A direct transfer is made by setting a new CT4Q (through an OGT key or the CT4Q ActID) that routes to a control list containing a function provided by an SN. The operator position can update the context block while the call is at the position and the context block remains with the call for the life of the call.

The SN or operator can specify the number of operator-to-SN transitions allowed on a per call basis. The operating company can limit the number of times, typically 1 or 2, that a call is redirected to an automated system. This capability is controlled by the following mechanisms:

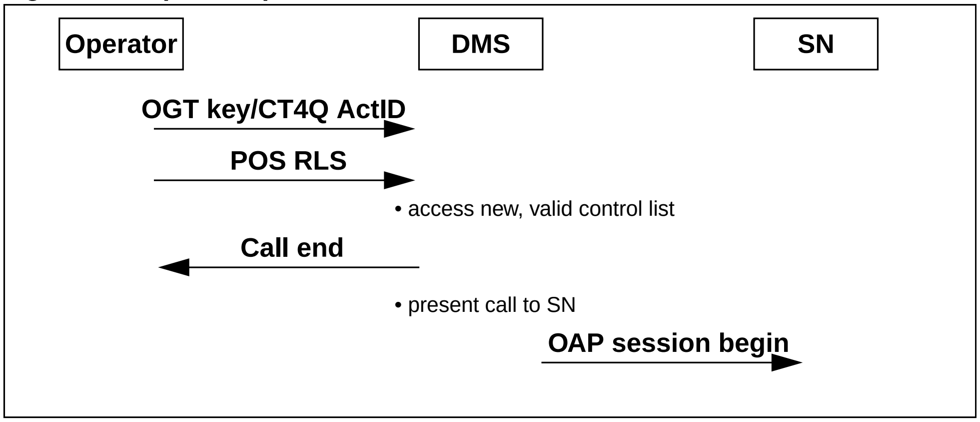
- Allow the SN to specify the number of operator-to-SN transitions allowed for a particular call using the OAP protocol.
- Allow the operator position application to specify the number of operator-to-SN transitions allowed for a particular call using OPP.

Each call maintains a counter which is zeroed only at the beginning of the call. The counter is not zeroed at call float. The counter increments each time a successful operator to SN handoff occurs. When the number of operator to SN handoffs limit is reached, the call remains at the operator position with a display indicating that the call can no longer be handed off.

Consider a service where the operator makes a non-switch connection to a database. When the operator transfers the call, the SN may need to contact the database to retrieve call data set up by the operator position.

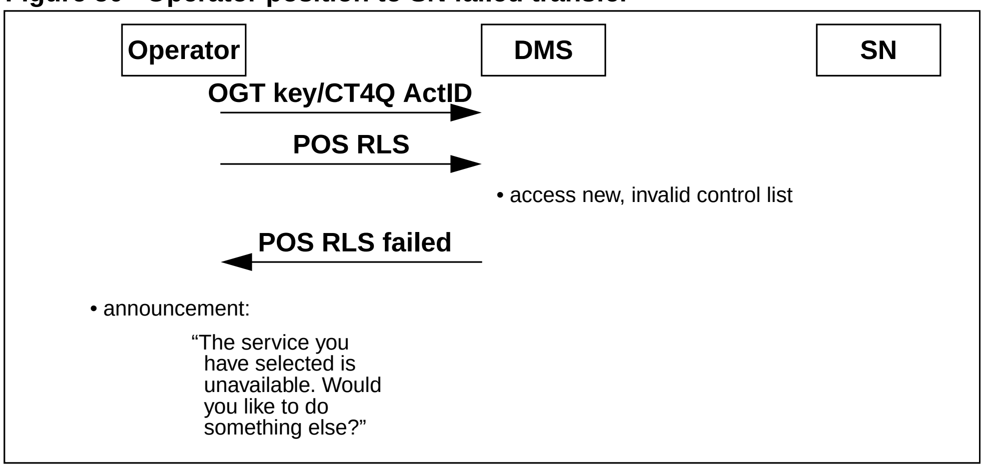
The following figure shows the high-level view of the message exchange for a successful transfer.

Figure 49 Operator position to SN successful transfer



The following figure shows the high-level view of the message exchange for a failed transfer because of an invalid control list.

Figure 50 Operator position to SN failed transfer



Operator position to operator

This type of transition is not allowed for OSSAIN transitions. If the requested CT4Q routes to a control list containing a function provided by an operator, the position release is NACKed (negative acknowledged) and the call remains in the initial operator's control. However, standard TOPS transfers from operator to operator are not affected.

Operator position to automated system

This type of transition is not allowed for OSSAIN transitions. If the requested CT4Q routes to a control list containing a function provided by an automated system, the position release is NACKed and the call remains in the initial operator's control.

DAS processing are not affected.

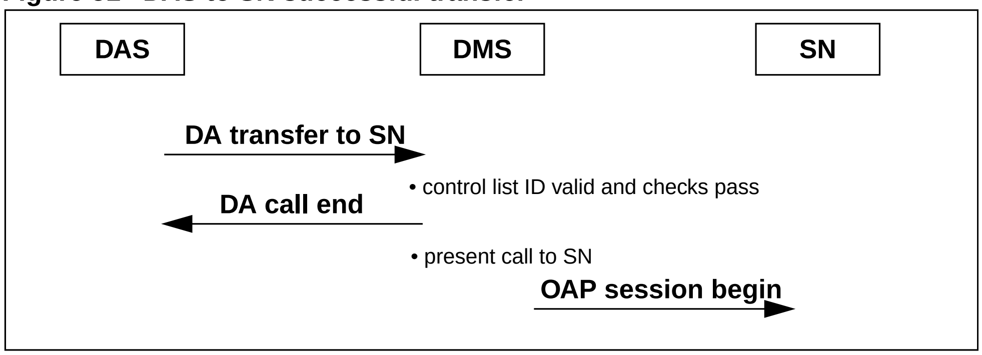
DAS to SN

When this type of direct transfer is requested, the switch checks the validity of the control list identifier contained in the standard DA protocol message. In addition, the switch validates the billing and changes the call to TA.

Note: Field USESERV in table OAFUNDEF can be datafilled to override the switch assumptions in changing from DA to TA. If the service is not changed, then the call origination will not change.

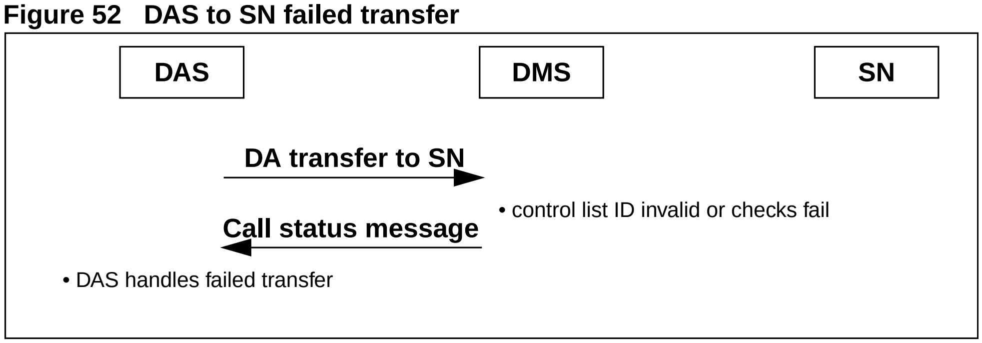
The following figure shows a high-level view of the message exchange for a successful transfer.

Figure 51 DAS to SN successful transfer



The following figure shows the high-level view of the message exchange for a failed transfer.

Figure 52 DAS to SN failed transfer



DAS to operator

This type of transition is an ADAS function and is unchanged by a DAS transition to SN.

DAS to automated system

This type of transition is not allowed for OSSAIN transitions.

OAP overview

The DMS switch and SNs communicate data messages using the Open Automated Protocol (OAP). OAP is used to perform the following OSSAIN tasks:

- identify calls to SNs
- allow SNs to request specific actions for a call
- inform SNs of changing call characteristics
- provide a means for basic maintenance of SNs

This section provides an overview of OAP, focusing on the following areas:

- OAP definition
- functional characteristics of OAP

For *complete* information on OAP, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

OAP definition

OAP is a switch-to-SN interface with the following characteristics:

- Deterministic—for each request, the expected responses are defined. Each request and response has a predefined set of data.
- Service-independent—no assumptions are made by the switch about the service being provided. Side effects are minimized, which gives SNs maximum control of the progress of the call.
- Abstract Syntax Notation/Remote Operations Service Element (ASN/ROSE)—OAP is designed using ASN/ROSE concepts, which provide a general description and encoding approach for passing structured data types between systems.

The OAP operation set includes a series of request and inform operations with their associated responses, if any. Each request operation is defined with a success response and an error response.

OAP message types

Following is a list of OAP message types:

- *Inform* messages update the message receiver of a call or SN maintenance characteristic (such as a subscriber party going on hook), or request an operation that does not require a response. This message is not responded to unless there is a protocol violation.
- *Request* messages request that a specific operation be performed by the message receiver (such as a connect voice link). The receiver responds to the request with either a success response message or an error response message.
- *Success response* messages indicate that the requested operation was successfully performed by the receiver of the request operation (for example, a voice connection was successful). This message is not responded to unless there is a protocol violation.
- *Error response* messages indicate that the operation request was attempted, but failed. An example of an error response is when a voice connection fails because the voice channel sent with the message is invalid. A set of possible errors is defined for each operation request message. This message is not responded to unless there is a protocol violation.
- *Reject* messages indicate that there is a protocol violation with a message or that the message could not be parsed. If the message is a request or inform type, an attempt is not made to perform the operation. A set of protocol violations is defined for OAP.

The following table lists the message types and the expected responses.

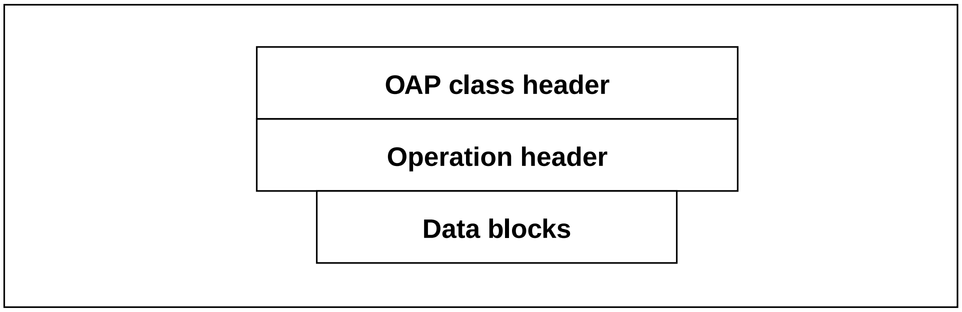
Table 10 Message types and expected responses

Message type	Success response	Error response	Reject
Inform	Not applicable	Not applicable	Applicable
Request	Applicable	Applicable	Applicable
Success response	Not applicable	Not applicable	Applicable
Error response	Not applicable	Not applicable	Applicable
Reject	Not applicable	Not applicable	Not applicable

OAP message format

Each OAP message has one class header and one operation header. Each operation can have zero or more data blocks. The following figure shows the OAP message format.

Figure 53 OAP message format



OAP class header

The OAP class header identifies the process that should receive the OAP message. The header contains basic information that all messages require, such as routing information, message length, and protocol version.

There is a class header for each process that acts on a specific message. The class header allows messages to be routed to more specialized handlers. Following are the predefined class headers used by OSSAIN:

- call processing class header, used for call-related messages
- node class header, used for node level messages, such as maintenance, alarms, and audits
- session pool class header, used for session pool level messages, such as maintenance, alarms, and audits
- MIS class header, used for QMS Management Information System (MIS) messages

In addition to the basic information, each class header has information specific to that class. For example, call processing messages require a function identifier and a call identifier. These fields are included in the call processing class header. In the node class header, however, these fields are not included, because they are not needed by node maintenance messages.

Operation header

The OAP operation header is a ROSE encoding header that contains information specific to the operation, such as the operation identifier, operation message length, and an invoke identifier. The invoke identifier is used by the receiver of a response message to correlate a response to the operation request.

All OAP operations and responses are of a specific message class. Each operation and response has zero or more logical groupings of data fields, known as a data blocks. The data blocks are specified as mandatory or optional. (See “Data blocks” for details.)

Operation classes

OAP makes a distinction between call processing operations and non-call processing operations. Call processing operations include the call processing class. Non-call processing operations include the node and session pool classes.

Message senders

The definition of each operation also includes a list of preconditions, postconditions, side effects, possible responses, and a list of valid senders. Following is a list of all potential message senders:

- the call processing switch
- the maintenance switch
- the service node

The list of senders stated with each operation can be used to determine whether the operation is unidirectional (sent by the switch only or sent by the SN only) or bidirectional (sent by both switch and SN).

For example, the valid sender for the Class Charge Request operation is the SN. This operation is unidirectional and is sent only by the SN that has control of the call.

Data blocks

Each operation and response has a predefined set of zero or more data blocks (DB). Each data block in an operation and response is specified as being mandatory or optional. If a data block is mandatory, it must be sent with the operation or response. If the data block is optional, it may or may not be sent, depending on the call characteristics.

Allowing for optional data blocks simplifies the protocol and reduces the message size. Following are the types of optionality supported:

- An optional data block is sent only if it is applicable at the time. For example, a Directory Number data block is sent in a session begin message only if there is directory number data.
- Multiples of the same data block may be sent for one operation. For example, the Directory Number data block may be sent twice in a session begin message, one for the calling party and one for a forward party.

- From a set of data blocks, only one is sent. There may be several data blocks that contain similar information and only the data block that is applicable is sent. Based on the trigger event, one of the following three types of trigger data blocks will be sent in a Trigger Event Inform message:
  - — a Trigger Information DB
  - — a Trigger Treatment DB
  - — a Trigger Ringing DB

Although the data blocks are listed in a specific order for each operation and response, they do not need to be sent in that order. The data blocks can be sent in any order and nothing is inferred from the ordering.

Note: The size of the operation header and the associated data blocks for the operation cannot exceed 280 bytes for incoming messages and 1500 bytes for outgoing messages.

How OSSAIN uses OAP

OSSAIN uses OAP as an interface between the switch and an SN. OAP is the means for passing information to perform the following tasks:

- presenting calls to an SN
- allowing an SN to initiate calls or retrieve floated calls
- allowing an SN to release itself from a call by the following methods:
  - — floating the call
  - — transferring the call to a carrier
  - — transitioning the call to another function provider
  - — sending the call to treatment
- allowing an SN to create a conference call
- allowing an SN to change the base service of a call
- allowing an SN to request specific billing operations for the call (such as class charge, validate billing number)
- allowing an SN to connect to a call simultaneously with another SN or with an operator
- connecting and releasing specific network resources (such as voice links) at the request of an SN
- providing centralized AMA operations (such as append AMA module, generate AMA record)
- informing an SN of call events and changes to resources associated with a call

- providing SN and session pool maintenance, which includes the following functions:
  - audits to ensure communication between the switch and SN/session pools
  - ability of an SN and a session pool to generate logs and alarms at the switch
- sending SMS messages to wireless callers

Note: The preceding list of tasks is not exhaustive. For complete information on OAP capabilities, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Functional characteristics of OAP

The following paragraphs briefly describe some of the functional characteristics of OAP.

Protocol version

The protocol version for OAP consists of an OAP release and increment number (X.Y). A given node in the OSSAIN network should be able to support at least 3 releases older than its own release.

For example, if the OAP protocol release of the node is X, then the node should support at least X, X-1, X-2, and X-3 releases. The node also should support all increments within any supported release that are less that the node's release.

The switch will support all increments of any supported releases, which is typically 3 prior releases.

Note: The OAP protocol release number is independent of the switch software release number (such as TOPS20 or SN07).

Protocol version negotiation

Each OAP class header exchanged by OSSAIN nodes contains the following two protocol version numbers:

- the version of the message, known as message protocol version (MPV)
- the maximum protocol version at which the message sender can communicate

Each OSSAIN node in the network (switch or SN) can have a different combination of MPV and maximum protocol version. Because of this complexity, OSSAIN nodes need to keep track of these two protocol versions so they can communicate in the most efficient or *optimal* way.

The optimal MPV has the following conditions:

- The MPV is not higher than the version that the receiving node can process.

- The MPV is not lower than the maximum protocol version of both the sending and receiving nodes.

When a receiving node determines that the MPV is not optimal, then the receiving node and sending node must negotiate the protocol version.

The nodes *cannot* negotiate the protocol version when the sender's maximum protocol version is lower than the lowest version that the receiving node can process. In this event, the sending node must upgrade to a protocol version that is negotiable with the receiving node.

Sequence numbers

Sequence numbers track the order of messages in the message flow and help to track lost messages. The sequence number is contained in all OAP class headers and is included in every message between the switch and SN. Each message contains only one sequence number. The number indicates the number of messages that the message sender sent for this session, including the current one.

Lost messages

The switch can detect that an OAP message has been lost when it receives a message with an out-of-sequence number. When the switch detects a lost message, it pegs an OM register. For information on OMs, refer to Chapter 12: "OSSAIN operational measurements."

Security

Vital data (such as calling card numbers) passed between the switch and SNs are encrypted for protection. However, encryption provides only minimal security. Commercial encryption techniques should be used on the network for more security.

Sanity timers

Sanity timers are timers set by the switch to ensure that resources are not left connected to dead calls. When a sanity timer expires, the switch automatically frees any resources associated with the timer. Such resources include subscriber facilities, connections to external systems (such as SNs), and switch call resources (such as extension blocks). Without sanity timers, manual intervention is required to find and free resources that have not recovered.

Four sanity timers are defined for OSSAIN:

- session begin timer
- on-hook sanity timer
- call sanity timer
- queued sanity timer

Session begin timer operation

The session begin timer ensures that the SN receives the OAP Session Begin message sent by the switch when it routes a call to an OSSAIN function. The timer is set for a particular function in table OAFUNDEF.

Initiated

If the SBTIMOUT subfield in OAFUNDEF is set to Y, the DMS switch starts the session begin timer after it sends an OAP Session Begin message to the SN. The SBTIMER subfield specifies the value of the timer in seconds.

Note: When the SBTIMOUT subfield is set to N, the switch does not start a session begin timer. The switch assumes that the SN received the OAP Session Begin message and has control of the call. If the SN did *not* receive the message, the call remains connected to the SN session until the on-hook timer or call sanity timer takes effect (or until the SN is taken out of service).

Reset

The session begin timer is never reset.

Cleared

If the switch receives an OAP operation from the SN before the session begin timer expires, the timer is cleared.

Action upon timer expiration

When the session begin timer expires, the call is routed based on datafill for an origination failure in table OAFNDISP.

On-hook sanity timer operation

The on-hook sanity timer ensures that a call does not remain hung indefinitely after subscriber on hook or release. Note that this timer may not apply to standard ISUP, because ISUP facilities can automatically be cleared upon disconnect.

Initiated

The on-hook timer for a particular subscriber is initiated when the switch detects that the subscriber has gone on hook. An office-wide default value determines the duration of the timer. However, this may be overridden by datafilling a timer value against the session pool on a particular node.

OSSAIN provides a table, OAINPARM (OSSAIN Parameter), for datafilling parameters specific to OSSAIN processing. The parameter to be used as an office-wide default for the on-hook timer is called ON_HOOK_TIMER_DURATION.

OSSAIN also provides a table that defines options for a specific session pool—table OASESNPL (OSSAIN Session Pool). This table contains a selector that allows the operating company to override the office-wide default for the on-hook timer. The field is called ONHKTMR. If set to USEDEFLT, the switch uses the office-wide default specified by ON_HOOK_TIMER_DURATION in table OAINPARM. If set to OVERRIDE, a duration time value must also be specified. This value is then used instead of the default.

Reset

An on-hook timer, once initiated, cannot be reset. When a subscriber goes on hook, most call flows dictate that either the call should be freed (for example, when an originating subscriber goes on hook) or the connection to the subscriber should be released. In either case, the subscriber's facility is released, which causes the timer to be cleared instead of reset because it is no longer necessary.

Another scenario is when the subscriber goes back off hook subsequent to the on hook. This does *not* reset the timer, because the on-hook timer expiration is used to set the call sanity timer (refer to the following discussion on call sanity timers). Otherwise, in a situation where the switch and SN are not communicating, the subscriber could go off hook and disable all sanity timers.

Sanity timers implemented for operator services are reset upon receiving a message from the agent in control of the call (for example, the operator position). These timers were implemented for specific reasons associated with service processing.

However, in the context of OSSAIN, specific message flows do not exist. Therefore, OSSAIN does not reset the on-hook timer when a message exchange has occurred between the switch and the SN.

Another example is when the subscriber goes on hook but the SN, having encountered a problem, continues sending messages to the switch. In this case, if the timer is reset due to the receipt of a message and operator hold is in effect, the message has no impact on the call other than to tie up the subscriber's connection indefinitely. So resetting the facility on-hook timer upon the receipt of any message is inappropriate.

Cleared

The on-hook timer is cleared if the subscriber's facility is released or the call is taken down for any reason.

Action upon timer expiration

If the on-hook timer expires, an OAIN600 log is generated and the call sanity timer is initiated, if appropriate. The OSSAIN on-hook timer is different from the earlier AABS on-hook timer, which caused the call to be taken down.

Call sanity timer operation

The call sanity timer ensures that SN connections (such as data and voice), subscriber connections, and switch-related call resources (such as extension blocks) are not left connected to dead calls.

Initiated

The call sanity timer is initiated under the following conditions:

- The originator's on-hook timer has expired and there is no terminating party connection.
- The originator's and terminator's on-hook timers have expired.
- The terminator's on-hook timer has expired and there is no originating party connection.
- There are no originating nor terminating party connections.

The parameter, CALL_SANITY_TIMER_DURATION, in table OAINPARM, is used as an office-wide default for the call sanity timer.

Table OASESNPL contains a selector to override the office-wide default for the call sanity timer. The field is called CALLTMR. If set to USEDEFLT, the switch uses the office-wide default specified by CALL_SANITY_TIMER_DURATION in table OAINPARM. If set to OVERRIDE, a duration time value must also be specified. This value is then used instead of the default.

Reset

The call sanity timer is reset if any operation is received from the SN that is associated with the call, as determined by the call identifier (CALLID). The CALLID is contained in the OAP protocol message header. The incoming message must contain the CALLID for the timer to be reset. If the SN CALLID does not match the switch's version, the protocol violation is logged and the timer continues.

Cleared

If a subscriber is subsequently connected to the call, the call sanity timer is cleared.

Action upon timer expiration

When the timer expires, the call is taken down by the switch and an OAIN601 log is generated. The switch informs the SN that the call has been taken down and releases the data and voice connections to the SN, releases all subscriber connections involved in the call, cancels any outstanding request, and frees all switch related call resources (such as extension blocks).

Note: Refer to Chapter 11: "OSSAIN logs," for more information on OSSAIN logs.

Queued sanity timer operation

Under certain conditions, a call can be queued for a session whose session pools are out of service. The queued sanity timer clears up such a call.

Initiated

The duration of the queued sanity timer is 10 seconds. It is initiated under the following conditions:

- The call is queued with all remaining parties on hook.
- The remaining parties go on hook while queued.

Reset

The queued sanity timer is never reset even if any remaining subscriber goes back off hook.

Cleared

If a session becomes available to serve the call, the queued sanity timer is cleared and the call is presented to the SN.

Action upon timer expiration

When the queued sanity timer expires, the call is taken down by the switch and an OAIN601 log is generated.

Note: Refer to Chapter 11: “OSSAIN logs,” for more information on OSSAIN logs.

Hierarchy of OSSAIN sanity timers

The session begin timer applies only to subscriber-originated sessions. When enabled by datafill in table OAFUNDEF, the session begin timer takes precedence over all the other sanity timers.

The on hook sanity timer and the call sanity timer work in a hierarchical manner. When one or more subscribers are present and have not gone on hook, neither timer is in effect. If a subscriber goes on hook, an on-hook timer is initiated.

The call sanity timer is initiated when all of the on-hook timers expire for all connected subscriber parties or when no parties remain connected to the call. The call sanity timer is cleared if a subscriber is subsequently connected to the call (a subsequent on hook initiates an on-hook timer). Thus an on-hook timer and a call sanity timer never run simultaneously.

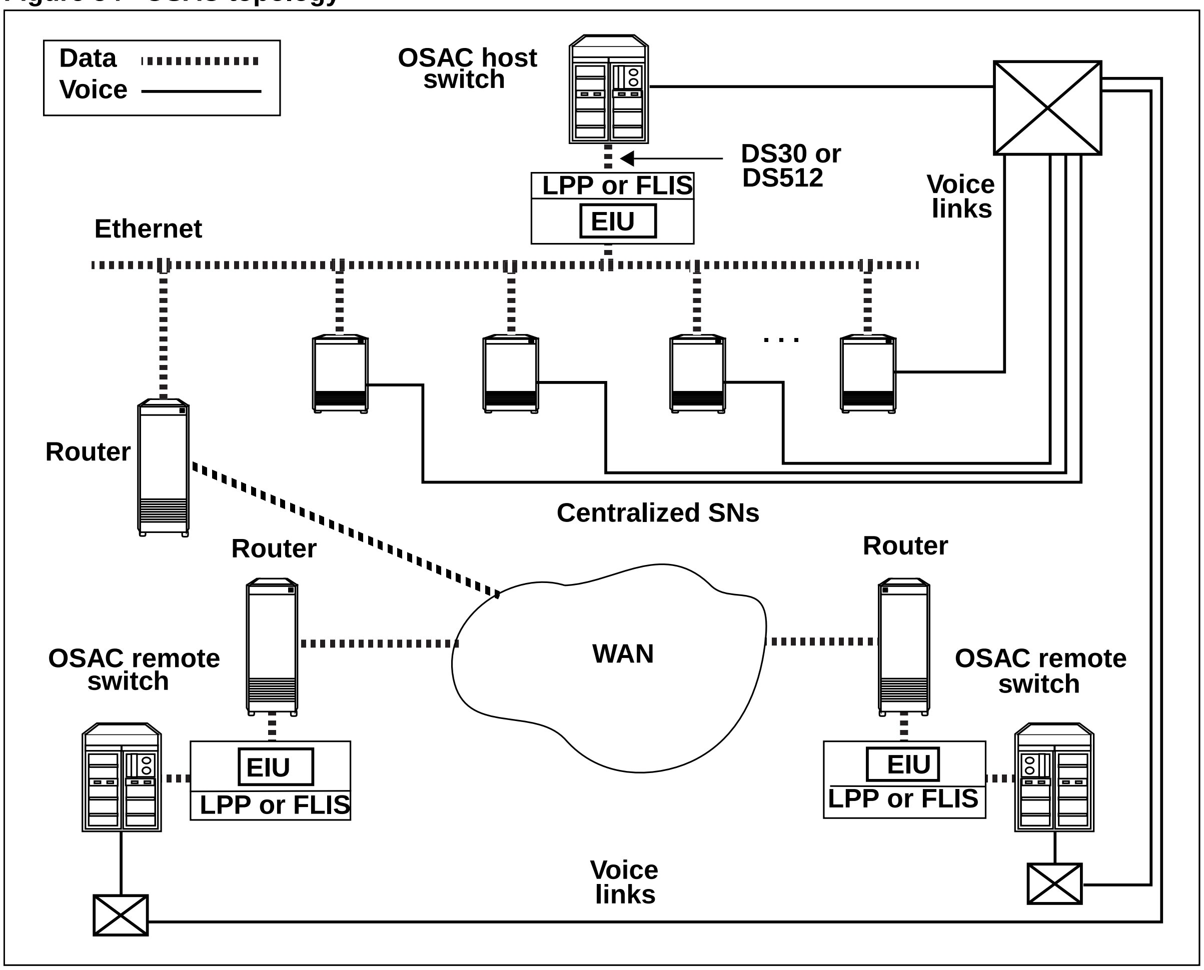
There may be multiple on-hook timers running simultaneously if multiple subscribers are connected and on hook. These timers are independent of each other.

Chapter 3: OSAC call processing

Centralized OSSAIN (OSAC) is a TOPS architecture in which call processing control is distributed among more than one switch and several *centralized* SNs. OSAC allows the services offered by the centralized SNs to be shared by multiple switches.

In the OSAC network, the switches are set up in a host and remote configuration, as shown in the following figure.

Figure 54 OSAC topology



The figure above shows the switches using EIUs to connect to the Ethernet LAN, but alternatively, one or more of the switches could use its XA-Core Ethernet interface.

Depending on the network configuration, a given switch can function as a standalone, an OSAC host, or an OSAC remote for a call. However, for the purpose of understanding OSAC call processing, this chapter discusses a simple OSAC network configuration. For details on an advanced configuration, refer to Appendix B: “Advanced OSAC configuration.”

In the OSAC network, the OSAC host switch centralizes the distribution of calls to SNs and centralizes the voice connections used by the SNs. The following table shows the tasks that each switch in the OSAC network performs.

Table 11 OSAC network tasks

Task	OSAC host switch	OSAC remote switch
Call distribution	X	
Voice connections with SN	X	
Node maintenance	X	
Session pool maintenance	X	
Call control		X
Connectivity monitoring	X	X

Call processing for OSAC is based on the OSSAIN base call processing used in the standalone configuration of the network, which is described in detail in Chapter 2: “OSSAIN software functionality.”

Note: Users need to understand the concepts of OSSAIN base call processing before they proceed to OSAC call processing.

OSAC call processing occurs in both the OSAC remote and OSAC host switches. OSAC call processing distributes calls to centralized SNs and makes centralized voice connections.

This chapter describes in detail how OSAC software functions across the network, focusing on the following major components:

- centralized automatic call distribution
- centralized voice connections
- parallel datafill requirements for OSAC

Note: The OSAC host switch also centralizes the maintenance of SNs. For details, refer to Chapter 10: “OSSAIN maintenance.”

Messaging protocol used between OSAC switches

Messaging is required for call processing and maintenance between an OSAC host switch and an OSAC remote switch. This messaging is provided by a protocol that is proprietary to Nortel. The *OSSAIN User's Guide* does not provide user information on this proprietary protocol.

OSAC call distribution

The distribution of calls is centralized at the OSAC host switch for sessions with a centralized SN. Centralized call distribution allows an OSAC remote switch to have a session to a centralized SN.

If a call at an OSAC remote switch requires a function (service) provided by a centralized SN, the OSAC remote queries the OSAC host for a session to the SN. After the OSAC host obtains the session, the OSAC remote begins the call with the SN and they exchange call control messages.

Types of sessions

Sessions in an OSAC network are of the following two types:

- A *host-remote session* is between an OSAC host switch and an OSAC remote switch. It is used to exchange *OSAC call control messages*.
- An *SN session* is between an SN and an OSAC remote switch. It is used to exchange *OSSAIN call control message*s.

OSAC call distribution flow

The flow of a subscriber-originated call that arrives at the OSAC remote switch consists of the following broad steps:

1. The OSAC remote switch determines that a centralized SN provides the desired function.
2. The OSAC remote switch selects a host-remote session to the OSAC host switch.
3. The OSAC remote switch sends a session request message to the OSAC host switch.
4. The OSAC host switch selects a session to the centralized SN (using OSSAIN routing and queuing).
5. The OSAC host switch sends the session pool ID and SN ID to the OSAC remote switch.

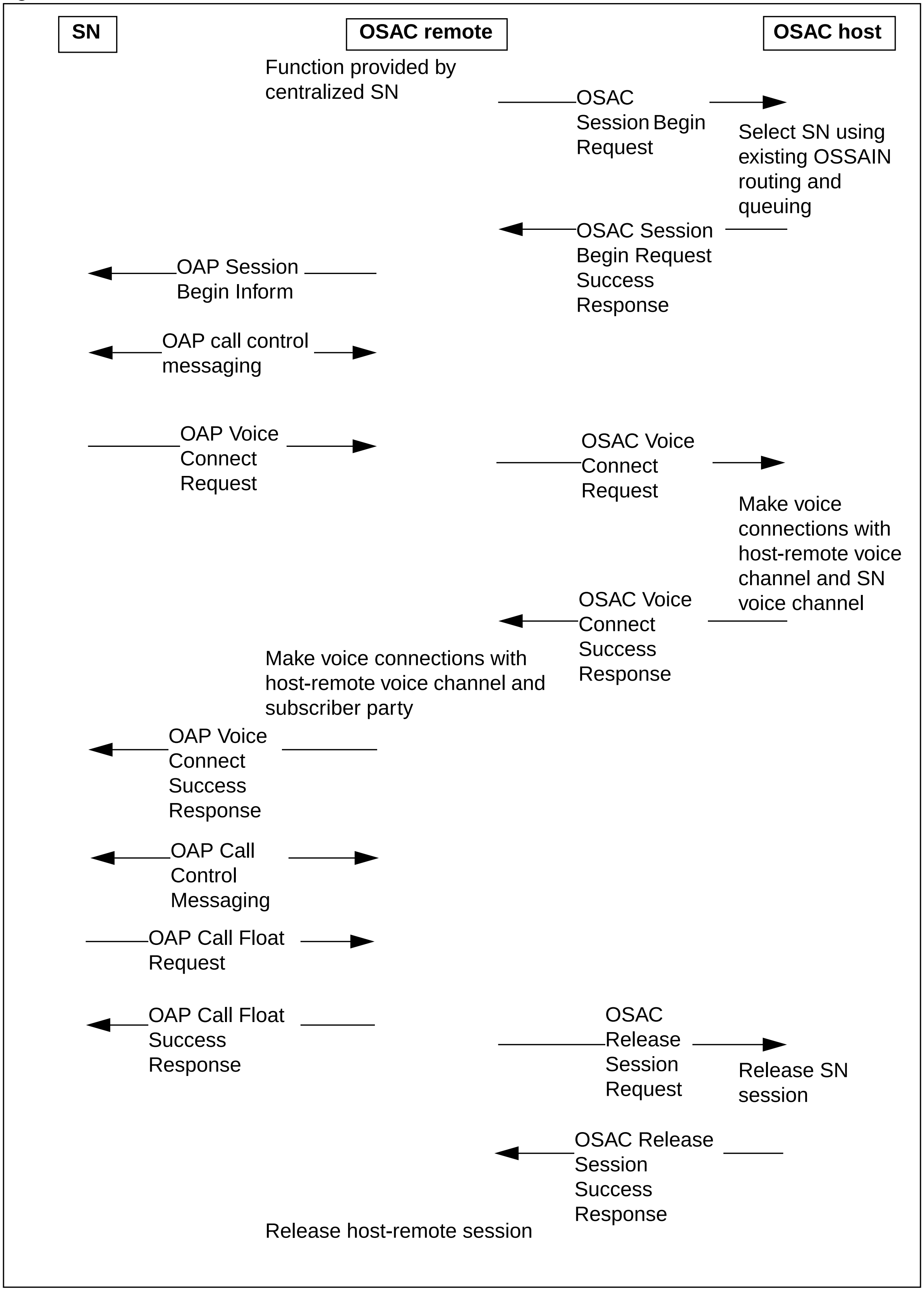
   Note: If an SN session is not immediately available, the call is queued and the OSAC host switch informs the OSAC remote switch. The OSAC remote has the option of canceling the request for an SN session.
6. The OSAC remote switch begins the session and exchanges OSSAIN call control messages with the SN.

   Note: If the SN needs a voice connection, the OSAC remote can send a voice connection request message to the OSAC host during this step.

7 The host-remote session between the OSAC remote switch and the OSAC host switch remains active until the OSAC remote ends the session with the SN.

The following figure shows the message flow for the call.

Figure 55 OSAC call distribution flow



OSAC call distribution datafill

OSAC call distribution requires datafill at both the OSAC host switch and the OSAC remote switch. This section briefly describes the datafill.

OSAC remote datafill

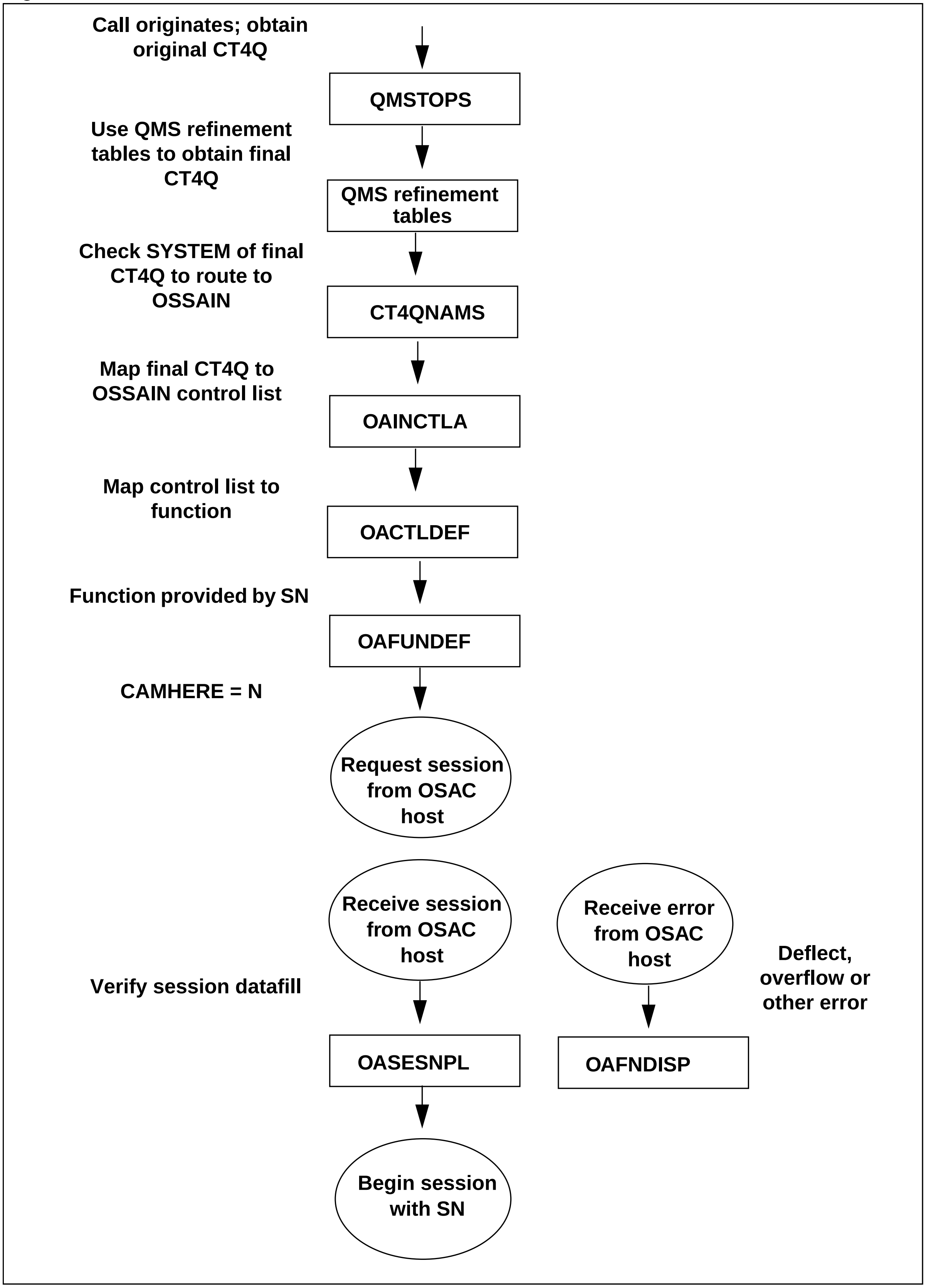
At the OSAC remote switch, an OSAC call routes using QMS datafill and OSSAIN datafill. Table OAFUNDEF specifies that the QMSCAM for the function is located at the OSAC host switch. The OSAC host is responsible for selecting the SN session.

The following tables require datafill at the OSAC remote switch for routing and queuing:

- QMS tables:
  - QMSTOPS
  - CT4QCLAS
  - CT4QREST
  - CT4QPFXT
  - CT4QCAR
  - CT4QCLD
  - CT4QORIG
  - CT4QTIME
  - CT4QLANG
  - CT4QAUTO
  - CT4QNAMS
- OSSAIN tables
  - OAINCTLA
  - OACTLDEF
  - OAFUNDEF
  - OASESNPL
  - OAFNDISP

The following figure shows the table flow at the OSAC remote switch.

Figure 56 OSAC remote switch table flow



Note: For complete information on how to datafill the OSSAIN tables, refer to Chapter 7: “OSSAIN data schema.” For complete information on how to datafill the QMS tables, please refer to the *Customer Data Schema Reference Manual*.

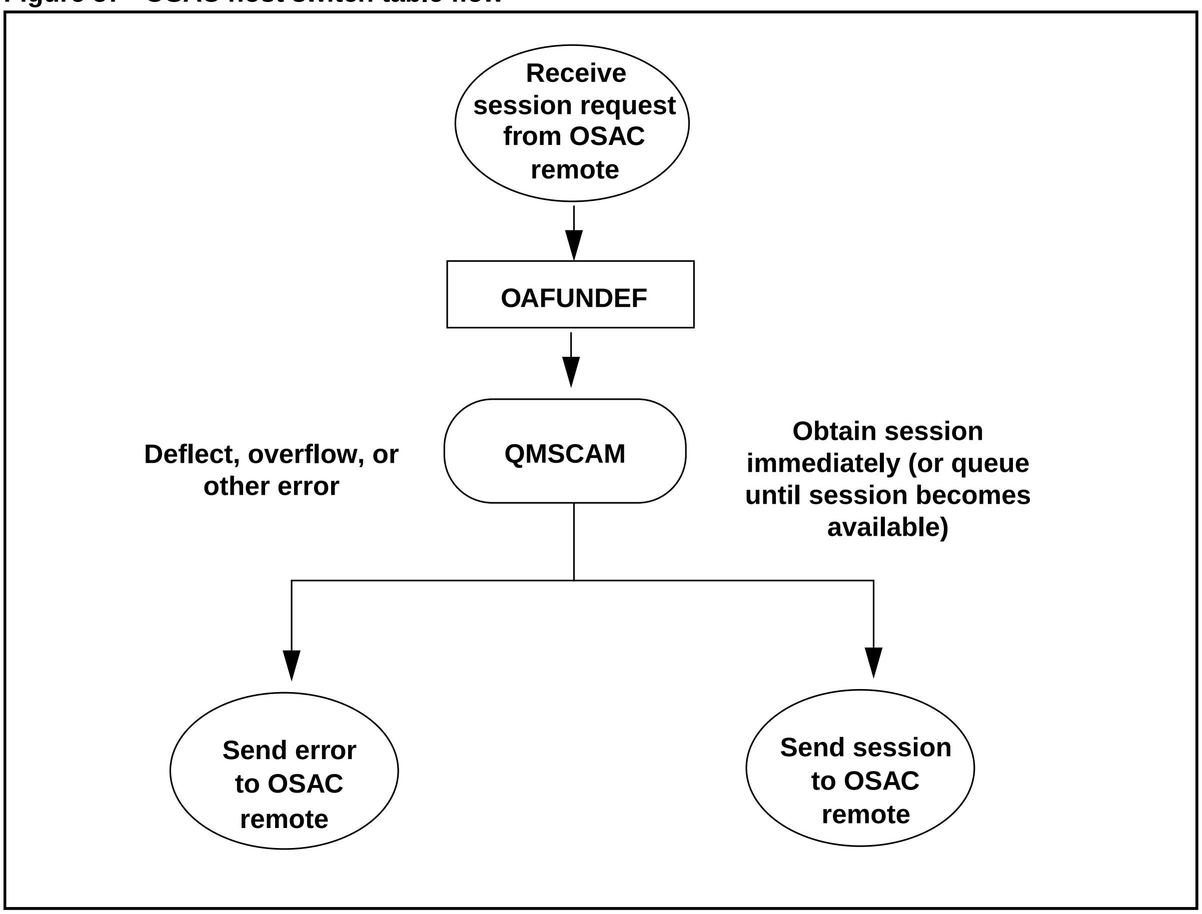
OSAC host datafill

The following tables require datafill at the OSAC host switch for selecting a session with the SN:

- QMS tables:
  - QMSCQDEF
  - QAPLNDEF
- OSSAIN tables
  - OAFUNDEF
  - OASESNPL
  - OQCQPROF

The following figure shows the table flow at the OSAC host switch.

Figure 57 OSAC host switch table flow



Note: Information on OSAC datafill that needs to be parallel in the OSAC switches is covered in “OSAC parallel datafill requirements” on page 143.

SN originations to the OSAC remote switch

A centralized SN can initiate a call to an OSAC remote switch. When the OSAC remote switch receives the request from the SN, the switch verifies the function ID in table OAFUNDEF. If the desired function is provided by a centralized SN, the OSAC remote obtains a host-remote session with the OSAC host.

The OSAC host verifies that the session is available (that is, the session pool or SN is not offline or busy). The OSAC host also verifies the function ID and responds to the OSAC remote with a success message.

The OSAC remote proceeds with the SN session. The host-remote session between the OSAC remote switch and the OSAC host switch remains active until the OSAC remote ends the session with the SN.

Positive assertion on the SN session

If the OSAC host determines that the SN session is already active, the OSAC host takes down the first call (and any resources) and continues the new call with the SN session.

Releasing an SN session

When either the SN or the OSAC remote switch ends the SN session, the OSAC remote requests the OSAC host to end the call. This request applies to both subscriber-originated calls and SN-originated calls.

After receiving the request, the OSAC host releases all resources associated with the host-remote session and the SN session, such as the SN voice channel and the host-remote voice channel. The OSAC host responds to the OSAC remote with a success message.

Note: The SN session ends before the host-remote session in order to prevent the OSAC host switch from selecting the same session for another call before the OSAC remote switch is finished.

Trigger event inform

With a trigger event, the OSAC remote switch must obtain a session with the SN *immediately* instead of requesting the SN session from the OSAC host switch. Obtaining a session this way prevents call data from being lost before another trigger event occurs.

Datafill in table OASESNPL at the OSAC remote switch indicates that the session pool is used only for trigger event inform messages and not for subscriber originations. For session pools datafilled as only for trigger event informs, the QMSCAM is not used and the first available session is selected.

Session pools datafilled as trigger event inform session pools in table OASESNPL must be datafilled against the function associated with the trigger event datafilled in table OAFUNDEF.

Session throttling

The OSAC host switch services a request for an SN session on a first come-first served basis. A poorly engineered network can result in a single OSAC remote switch using most of the SN sessions, which prevents other OSAC remote switches from obtaining a session to the same SN.

Adjusting (throttling) the number of host-remote sessions between the OSAC remote and the OSAC host remedies the problem. Datafill in table OASESNPL controls the maximum number of simultaneous calls handled by the host-remote session pool.

Likewise, an OSAC remote switch that handles a large volume of calls should be provisioned with an appropriate maximum number of host-remote sessions.

Error handling

The OSAC remote handles errors in processing by performing the action datafilled in table OAFNDISP. The following conditions require error handling:

- when the OSAC host deflects or overflows a call
- when the OSAC host returns an incorrect session ID or session pool ID (such as an incorrect origination type)
- when the OSAC remote cannot send a message to the OSAC host

If a failure of an SN or session pool occurs at the OSAC host switch, the OSAC host requests the OSAC remote to take down any active calls (to that SN or session pool).

Note: An OSAC switch generates a log any time it takes a call down.

Positive assertion on the host-remote session

Any time the OSAC host receives a request for a host-remote session that the OSAC host determines is already active, the OSAC host releases the first call (and any resources) and begins a new call.

OSAC voice connections

The OSAC host switch provides the voice connections to the centralized SN. Voice channels are centralized along with the sessions to SNs.

OSAC voice connection flow

The following broad steps comprise the flow of a voice connection at the OSAC remote switch:

1 The SN determines that it needs a voice connection.

2 The SN sends a voice connection request to the OSAC remote switch.

3 The OSAC remote switch forwards the request to the OSAC host switch. The same host-remote session used for the session request is also used for the voice connection request.

4 The OSAC host switch selects a host-remote voice channel between the OSAC host and the OSAC remote.

5 The OSAC host switch makes the network connections between the SN voice channel and the host-remote voice channel.

6 The OSAC host switch provides the host-remote voice connection information to the OSAC remote switch.

7 The OSAC remote switch makes the network connections between the host-remote voice channel and the subscriber.

8 The OSAC remote sends a success response to the SN.

While the OSAC remote and the OSAC host are processing the voice connect request, the SN should not send other requests. The SN should wait for the voice connect response before requesting other actions from the switch. The OSAC remote switch sends an error response if it receives any other request.

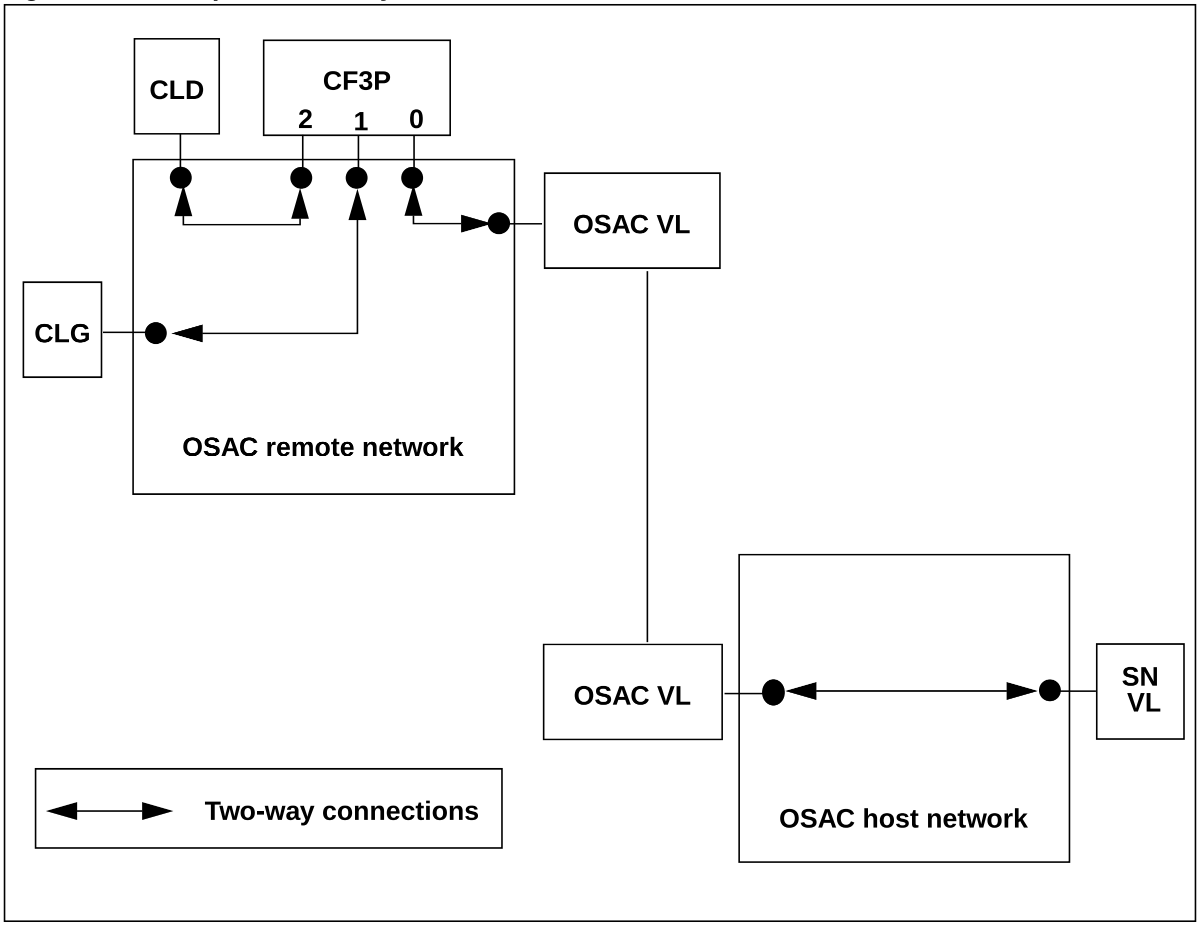
Voice connections with a centralized SN

The OSAC environment supports both two-way non-broadcast voice connections and one-way broadcast voice connections. This subsection describes non-broadcast voice connections. For a description of broadcast voice connections, refer to Chapter 4: “OSSAIN enhancements.”

Non-broadcast voice connections

Conference three ports (CF3P) are allocated only in the OSAC remote and only if needed. Refer to the following figure for an example of the voice connections for a centralized SN.

Figure 58 Example of two-way non-broadcast voice connections with a centralized SN



Note: Two centralized SNs can be connected to the same call at the same time. This is a *simultaneous connection*. Refer to Chapter 2: "OSSAIN software functionality," for information on simultaneous connections.

OSAC voice link datafill

Host-remote voice links require datafill at both the OSAC host switch and the OSAC remote switch. This section describes the datafill.

Note: For complete information on how to datafill the OSSAIN tables, refer to Chapter 7: "OSSAIN data schema." For complete information on how to datafill the trunk tables, please refer to the *Customer Data Schema Reference Manual*.

OSAC remote datafill

The following tables require datafill at the OSAC remote switch for selecting a host-remote voice link to the OSAC host switch:

- CLLI
- TRKGRP
- TRKSGRP

- TRKMEM
- OAVLMAP

OSAC host datafill

The following tables require datafill at the OSAC host switch for selecting an SN voice link to the OSAC host switch:

- CLLI
- TRKGRP
- TRKSGRP
- TRKMEM
- OSCVLGRP
- OAVLMAP

Table OSCVLGRP (OSAC Voice Link Group) is a table used only at the OSAC host switch. It identifies the voice link groups used to connect to another OSAC switch. The most idle selection sequence is used for selecting a voice link member from this group. Datafill in table OSCVLGRP first requires datafill in table CLLI (Common Language Location Identifier) and table OANODINV (OSSAIN Node Inventory).

Once the OSAC host switch selects the host-remote voice link member, it gets the logical voice channel number from table OAVLMAP (OSSAIN Voice Link Mapping). Table OAVLMAP is a table used at both the OSAC host and OSAC remote switches. It maps the logical voice channel to the physical voice circuit datafilled in table TRKMEM (Trunk Members). The OSAC host switch returns the host-remote logical voice channel number to the OSAC remote switch.

Releasing voice connections

After receiving a release voice connection from the SN, the OSAC remote switch releases the connection between the subscriber and the host-remote voice channel. The OSAC remote also sends a success response to the SN.

The OSAC remote forwards the request to the OSAC host. The OSAC host releases the connection between the host-remote voice channel and the SN channel.

Two-way voice link maintenance

Before using a host-remote voice link, both the OSAC host and OSAC remote switches must be on hook on the voice trunk. If one switch detects that the other switch is off hook, the voice trunk is displayed at the MAP (at the OSAC host switch) as remote make busy (RMB). Trunks in the RMB state are not used by the switch for call processing.

When both the SN and the switch are on hook on the voice trunk facility, the trunk can be selected for call processing. After the switch returns a success response to the voice connection request, the SN can use that facility immediately.

Note: For information on broadcast voice link maintenance, refer to Chapter 4: “OSSAIN enhancements.”

Auto voice link selection and connection capability

The OSAC host switch can select the voice link and automatically connect to it when establishing a new session with a centralized SN. Datafill in tables OAFUNDEF and SNVLGRP at the OSAC host switch is used to provide this capability. For details, refer to Chapter 4: “OSSAIN enhancements.”

Error handling

The OSAC host switch returns an error message to the SN for any of the following errors:

- The OSAC host switch cannot use the SN voice channel (for example, the voice channel is in use or the voice channel ID is invalid).
- The OSAC host switch receives a failure when connecting the SN voice channel and the host-remote voice channel. The OSAC host tries two times before notifying the OSAC remote of the failure.
- The host-remote voice channel is not available at the OSAC host switch.
- The voice link request is for a standard two-way connection, but datafill in table OAVLMAP (in the OSAC host) has the BCST_SEL subfield set to Y.
- The voice link request is for a one-way broadcast connection, but datafill in table OAVLMAP has the BCST_SEL subfield set to N.
- The voice link request is for a one-way broadcast connection that would exceed the value for the MAXCONNS subfield in table OAVLMAP.
- The voice link request is for a one-way broadcast connection for which the SN has specified the switch to count cycles, but the CUTTHRU field is set to IMMEDIATE.
- The voice link request is for a one-way broadcast connection, but the OSAN0101 SOC option code is not set to ON at the OSAC host switch.

If a voice connection goes down during a call, the SN session associated with the voice connection is also taken down. If the OSAC host switch detects the voice connection error, it releases the voice connection, sends a message to the OSAC remote, and waits for the OSAC remote to end the SN session. The OSAC remote then takes the session down, which sends a release session request to the OSAC host.

OSAC parallel datafill requirements

This section provides an understanding of the datafill required to set up OSAC nodes and process calls. Some tables require datafill that is *parallel* between the OSAC host switch and the OSAC remote switch.

This parallel datafill ensures that each OSAC switch is aware of the other switches and SNs in the network. This allows the OSAC host switch to distribute centralized SN sessions and make voice connections correctly to OSAC remote switches.

This section discusses the parallel datafill needed at *both switches* in the following five tables:

- OANODNAM
- OANODINV
- OASESNPL
- OAFUNDEF
- OAVLMAP

Note: For more information on how to datafill OSSAIN nodes, refer to Chapter 7: “OSSAIN data schema.”

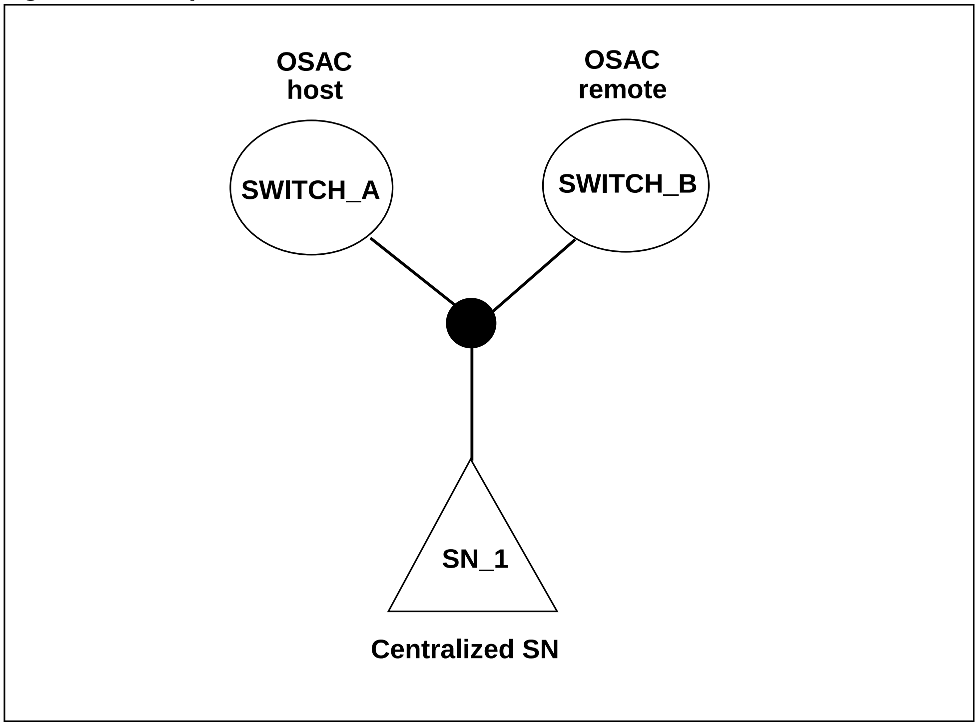
Table OANODNAM

Table OANODNAM (OSSAIN Node Name) uniquely identifies all nodes (SNs and switches) in the network and associates a name with the identifier (ID). Users must datafill table OANODNAM at *both* the host switch and the remote switch. This subsection describes the datafill.

Simple OSAC network

Refer to the following figure of a simple OSAC network. It shows one SN (SN_1) that is centralized at the OSAC host switch (SWITCH_A). The OSAC remote switch (SWITCH_B) also uses the functions provided by the centralized SN. (The lines show data connections only).

Figure 59 Simple OSAC network



OANODNAM example for the OSAC host

The following figure shows tuples in table OANODNAM. In the example, node ID 1 identifies the OSAC host as SWITCH_A, node ID 2 identifies the OSAC remote as SWITCH_B, and node ID 100 identifies the centralized SN SN_1.

Figure 60 MAP display example for table OANODNAM (host)

```
NODEID      NODENAME
---------------------------------------------------------------------
1           SWITCH_A
2           SWITCH_B
100         SN_1
```

OANODNAM example for the OSAC remote

The following figure shows tuples in table OANODNAM. In the example, the same three nodes are datafilled with the same node IDs.

Figure 61 MAP display example for table OANODNAM (remote)

```
NODEID      NODENAME
---------------------------------------------------------------------
1           SWITCH_A
2           SWITCH_B
100         SN_1
```

Verification of parallel datafill

The switches verify the node ID in table OANODNAM during the node RTS between the OSAC nodes (switches) and during the connection RTS between the OSAC remote switch and the OSN node.

Table OANODINV

Table OANODINV specifies to each switch the node type (OSAC, OSN, OSNM) and the relationship it has with the other nodes in the network.

Note: Table OANODINV also contains additional information about the node, such as the protocol, IP address, and maintenance-related settings. Refer to Chapter 7: “OSSAIN data schema,” for details. If an OSAC switch uses XA-Core Ethernet interface for OSAC messaging, the IP address other switches have datafilled for that switch must be consistent with the XA-Core Ethernet interface IP address that switch has datafilled for use with the far-end node.

Users must datafill table OANODINV at *both* the host switch and the remote switch. This subsection describes the distinctions between the datafill at the OSAC host and the datafill at the OSAC remote.

OSAC node type for switches

In the OSAC network, the *OSAC node type* identifies the host switch and the remote switch. Each switch in the network is datafilled in table OANODINV with the node type of OSAC.

OSN and OSNM node types for SNs

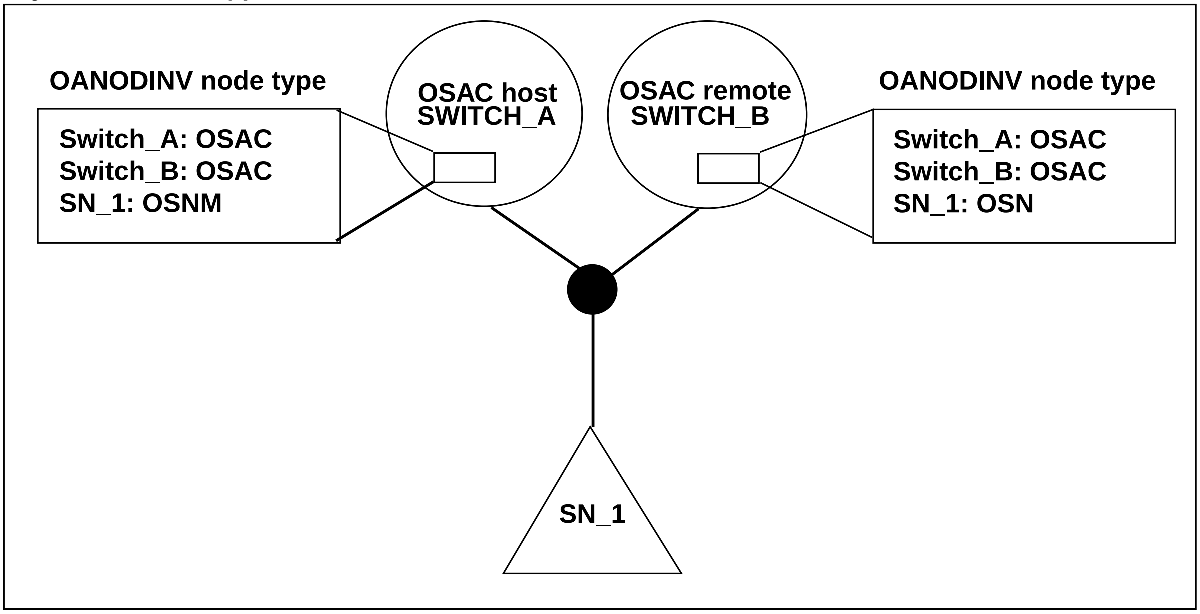
The *OSN node type* identifies the centralized SN at the OSAC remote switch. Likewise, the *OSNM node type* identifies the same SN at the OSAC host switch.

Note: OSNMs have a maintenance relationship with the OSAC host switch.

So the centralized SN is datafilled in table OANODINV at the OSAC remote switch with a node type of OSN and at the OSAC host switch with a node type of OSNM.

The following figure shows how the node types for the SN and switches are datafilled in table OANODINV. (The lines show data connections only).

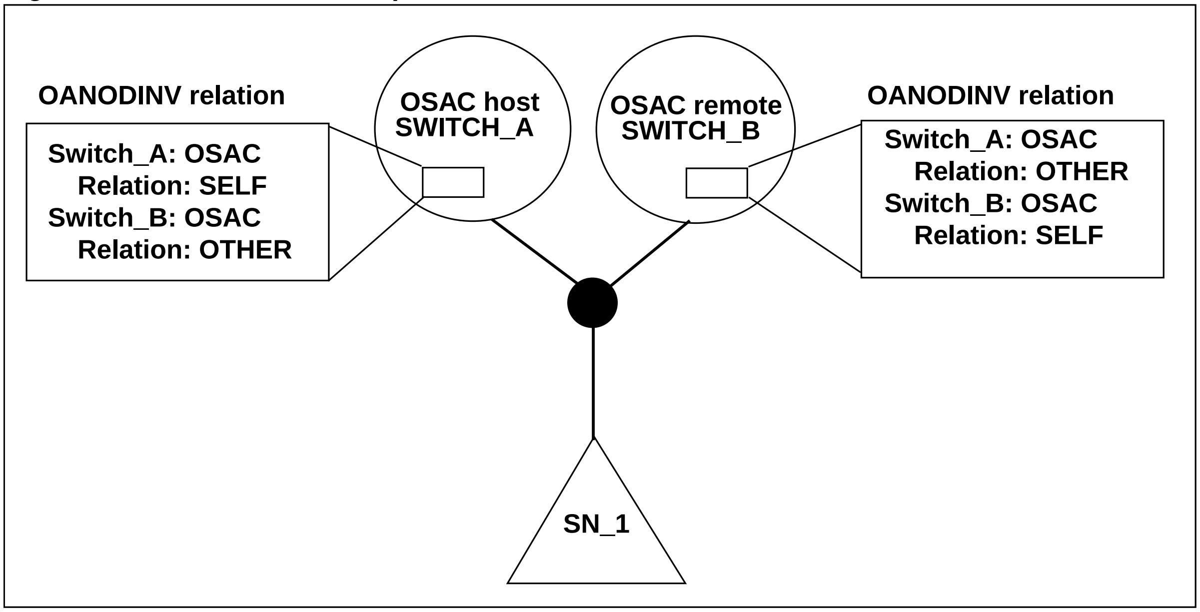
Figure 62 Node types in table OANODINV



In addition to identifying the node type, table OANODINV also provides the relationship of the OSAC switch to the other OSAC switch (or switches) in the network. A relationship of SELF applies to the switch where the datafill exists. A relationship of OTHER applies to all other switches.

In the following figure, table OANODINV at the OSAC host switch has the RELATION field of SWITCH_A datafilled as SELF; and the RELATION field of SWITCH_B datafilled as OTHER. Likewise, at the OSAC remote switch, the RELATION field of SWITCH_A is datafilled as OTHER; and the RELATION field of SWITCH_B is datafilled as SELF. (The lines show data connections only).

Figure 63 SWITCH relationships in table OANODINV



Note: Only one tuple of the OSAC type with a relation of SELF can be datafilled at a given switch.

OANODINV example for the OSAC host

The following example shows tuples in table OANODINV. In the example, the OSAC host SWITCH_A has a RELATION value of SELF, and the OSAC remote SWITCH_B has a RELATION value of OTHER. The SN, which provides the branding function, is centralized at SWITCH_A because it has a node type of OSNM.

Figure 64 MAP display example for table OANODINV (host)

```
NODENAME  NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_A  OSAC SELF  9        8        5
SWITCH_B  OSAC OTHER 0 UDP IPV4 47 187 8 1 EIU SWITCH 2 B 11 CITYB REMOTEOSAC 4 6 20
SN_1      OSNM 0 UDP IPV4 47 245 1 34 7000 EIU Y 2 240 60 SN 4 BB 3 CITYA BRANDING1
    8 6 60
```

OANODINV example for the OSAC remote

The following example shows tuples in table OANODINV. In the example, the OSAC remote SWITCH_B has a value of SELF, and the OSAC host SWITCH_A has a RELATION value of OTHER. The centralized SN has a node type of OSN.

Figure 65 MAP display example for table OANODINV (remote)

```
NODENAME   NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
---------------------------------------------------------------------------
SWITCH_A   OSAC OTHER 0 UDP IPV4 47 245 2 1 EIU SWITCH 4 C 12 CITYA HOSTOSAC 9 8 5
SWITCH_B   OSAC SELF  9        8        5
SN_1       OSN 0 UDP IPV4 47 245 1 34 EIU SWITCH_A Y 2 SN 4 BB 3 CITYA
     BRANDING1 8 6 60
```

Verification of parallel datafill

The switches verify datafill in table OANODINV during the node RTS, TST, and audit between the OSAC nodes (switches) and during the connection RTS, TST, and audit between the OSAC remote switch and the OSN node.

Verification checks the following information:

- PM type
- protocol
- IP address type
- IP address of SN

In addition, verification implicitly checks that the IP addresses the two switches have for each other are consistent with the interfaces they are intended to use. If an OSAC switch receives an OSAC message at an unexpected IP address, it generates an OAIN606 log and discards the message.

Table OASESNPL

Table OASESNPL (OSSAIN Session Pool) identifies the session pools used in OSAC and OSSAIN messaging. These session pools include both host-remote session pools and SN session pools. This datafill specifies to each switch the type of session pool used to send messages between nodes.

Users must datafill table OASESNPL at *both* the host switch and the remote switch. This subsection describes the distinctions between the datafill at the OSAC host and the datafill at the OSAC remote.

OASESNPL example for the OSAC host

The following example shows tuples in table OASESNPL. Session pool ID 1 identifies the SN session pool (SN_01_SP) at the centralized SN that the OSAC host switch sends to the OSAC remote switch. Session pool ID 20 identifies the host-remote session pool (HOST_2_REM) that the OSAC host switch uses to receive messages from the OSAC remote switch. The subfield ORIGTYPE is datafilled as OSACTERM.

Figure 66 MAP display example for table OASESNPL (host)

```
SESNPLID  SESNPLNM   MAXSESN   NODENAME    ORIGAREA
-----------------------------------------------------------------------------
1       SN_01_SP     100   SN_1  SUBSCRIBER SH USEDEFLT USEDEFLT N 5 UDP 7001 7002
20        HOST_2_REM    150   SWITCH_B    OSACTERM
```

OASESNPL example for the OSAC remote

The following example shows tuples in table OASESNPL. Session pool ID 1 identifies the SN session pool at the centralized SN that the OSAC host switch sends to the OSAC remote switch. Session pool ID 20 identifies the host-remote session pool (REM_2_HOST) that the OSAC remote switch uses to originate messages to the OSAC host switch. The subfield ORIGTYPE is datafilled as OSACORIG.

Figure 67 MAP display example for table OASESNPL (remote)

```
SESNPLID   SESNPLNM   MAXSESN  NODENAME    ORIGAREA
-----------------------------------------------------------------------------
1        SN_01_SP     100     SN_1      SUBSCRIBER R USEDEFLT USEDEFLT N UDP 7001
20        REM_2_HOST    150      SWITCH_A   OSACORIG
```

Verification of parallel datafill

The switches verify datafill in table OASESNPL during the host-remote session pool RTS, TST, and audit between the OSAC nodes (switches) and during the OSN session pool RTS, TST, and audit. SN session pool information is also verified during call processing.

Verification checks the following information:

- host-remote session pool:
  - session pool ID
  - maximum sessions
  - node ID
  - protocol
  - origination type

- OSN session pool:
  - — session pool ID
  - — maximum sessions
  - — node ID
  - — switch type
  - — protocol
  - — origination type

Table OAFUNDEF

Table OAFUNDEF (OSSAIN Function Definition) defines the function (service) that is provided by the SN. This datafill specifies to each switch the location of the CAM (call and agent manager) of the function.

For example, if the function is provided by a centralized SN, then table OAFUNDEF is datafilled at the OSAC remote switch with CAMHERE set to N, and has the host-remote session pool also datafilled. Likewise, at the OSAC host switch, CAMHERE is set to Y, and has the call queue also datafilled.

Users must datafill table OAFUNDEF at *both* the host switch and the remote switch. This subsection describes the distinctions between the datafill at the OSAC host and the datafill at the OSAC remote.

OAFUNDEF example for the OSAC host

The following example shows a tuple in table OAFUNDEF. Function ID 1 identifies the call queue (CQ1) that the OSAC host switch sends to the OSAC remote switch for the function. The subfield CAMHERE is datafilled as Y.

Figure 68 MAP display example for table OAFUNDEF (host)

```
FUNCID        FUNCNAME        FUNCAREA
-------------------------------------------------------------------------------
1             FUNC_01         SN  TASERV N N Y 7 N Y Y CQ1 Y Y
```

OAFUNDEF example for the OSAC remote

The following example shows a tuple in table OAFUNDEF. Function ID 1 identifies the host-remote session pool (REM_2_HOST) the OSAC remote switch uses to send a request for the SN that provides the function. The subfield CAMHERE is datafilled as N.

Figure 69 MAP display example for table OAFUNDEF (remote)

```
FUNCID        FUNCNAME        FUNCAREA
-------------------------------------------------------------------------------
1             FUNC_01         SN  TASERV N N Y 7 N Y N REM_2_HOST
```

Verification of parallel datafill

The switches verify the function ID in table OAFUNDEF only during call processing.

Table OAVLMAP

Table OAVLMAP (OSSAIN Voice Link Mapping) maps the logical voice channel to the physical voice circuit datafilled in table TRKMEM. The OSAC host switch returns the host-remote logical voice channel number to the OSAC remote switch.

Note 1: Only nodes of the types OSAC and OSNM are datafilled in table OAVLMAP.

Note 2: The voice circuits must first be datafilled in table TRKMEM before they can be used in table OAVLMAP. A host-remote voice link tuple cannot be deleted from table TRKMEM until all references to that circuit are removed from table OAVLMAP.

Note 3: Datafill in BCSTAREA specifies whether an SN voice link is used for broadcasting. Host-remote voice links cannot be used for broadcasting. For details on BCSTAREA, refer to the *Customer Data Schema Reference Manual*.

Users must datafill table OAVLMAP at *both* the host switch and the remote switch. This subsection describes the distinctions between the datafill at the OSAC host and the datafill at the OSAC remote.

OAVLMAP example for the OSAC host

The following example shows tuples in table OAVLMAP. In the example, the OSAC host switch makes voice connections for the OSAC remote switch (SWITCH_B).

The OSAC host selects trunk group OSACVL_B. It then selects the most idle voice link from that trunk group (for example, trunk member 2). The OSAC host maps the actual group and trunk number (OSACVL_B 2) to the logical voice channel number for the OSAC remote switch (channel 2). The OSAC host sends this channel number to the OSAC remote.

Figure 70 MAP display example for table OAVLMAP (host)

```
NDANCH                 CLLI        EXTRKNUM   BCSTAREA
----------------------------------------------------------------------
SWITCH_B   1           OSACVL_B     1           N
SWITCH_B   2           OSACVL_B     2           N
SWITCH_B   3           OSACVL_B     3           N
SWITCH_B   4           OSACVL_B     4           N
SN_01      1           SNVL_01      1           Y IMMEDIATE 80 OCTO
SN_01      2           SNVL_01      2           N
SN_01      3           SNVL_01      3           N
SN_01      4           SNVL_01      4           N
```

Note 1: This example shows an SN voice link used for broadcasting at the OSAC host switch.

Note 2: At the OSAC host switch, host-remote voice trunk groups must have their direction set to OG (outgoing) in table TRKGRP.

OAVLMAP example for the OSAC remote

The following example shows tuples in table OAVLMAP. In the example, the OSAC remote switch receives the logical voice channel 2 from the OSAC host switch (SWITCH_A). The OSAC remote then maps this channel to the actual group and trunk number (OSACVL_A 2).

Figure 71 MAP display example for table OAVLMAP (remote)

```
NDANCH                  CLLI         EXTRKNUM    BCSTAREA
----------------------------------------------------------------------
SWITCH_A   1            OSACVL_A       1         N
SWITCH_A   2            OSACVL_A       2         N
SWITCH_A   3            OSACVL_A       3         N
SWITCH_A   4            OSACVL_A       4         N
```

Note: At the OSAC remote switch, host-remote voice trunk groups must have their direction set to IC (incoming) in table TRKGRP.

Verification of parallel datafill

Call processing verifies only a limited amount of voice link data.

Chapter 4: OSSAIN enhancements

This chapter describes enhancements to the core OSSAIN software functions discussed in Chapter 2 and Chapter 3. OSSAIN enhancements provide greater flexibility in OSSAIN call processing and greater support of other TOPS features.

The following table lists each OSSAIN enhancement according to the product computing module load (PCL) where it was introduced. In cases where the enhancement was modified in a later PCL release, the latest PCL appears in parentheses. Any restrictions that apply to OSSAIN functions, including enhancements, are discussed in Chapter 5: "OSSAIN interactions."

Table 12 OSSAIN enhancements

Enhancement name	PCL or load	Page number
OSSAIN preprocessing	LET0007	page 154
Simultaneous connections	LET0007 (SN07)	page 156
Operator handoff	LET0007 (LET0012)	page 160
Service provider identifier (SPID)	LET0007	page 161
TOPS Local Number Portability (LNP)	LET0007	page 163
Release link trunking (RLT)	LET0007 (SN07)	page 163
Call timing	LET0007 (LET0012)	page 167
Outgoing ISUP	LET0009	page 169
Voice link broadcasting (VLB)	LET0009	page 170
OSSAIN conferencing	LET0009 (LET0011)	page 176
Country direct	LET0010	page 185
OSSAIN QMS MIS	LET0010	page 189
OSSAIN alternate routing	LET0011	page 196
Additional QMS CT4Q refinements	LET0011 (SN07)	page 199

Table 12 OSSAIN enhancements

Enhancement name	PCL or load	Page number
Auto voice link selection and connection	LET0011	page 201
Commercial credit card sales report	LET0011	page 203
Authorization Codes	SN07	page 203
Automatic carrier selection	LET0011	page 204
Estimate of charges	LET0011	page 204
Global support	LET0011	page 204
Called DNSCRN screening	LET0011	page 205
Special location routing number (SLRN)	LET0012	page 205
OSSAIN Support for DA Automation	LET0013	page 206
Short Message Service	SN07	page 206
XA-Core Ethernet interface	SN07	Chapter 6: “OSSAIN engineering” on page 237

OSSAIN preprocessing

TOPS QMS allows calls that are assigned to a TOPS operator or automated system to receive OSSAIN *preprocessing* from an SN. With preprocessing, calls that typically are denied OSSAIN processing can have a limited session with an SN.

For example, an SN can perform branding of an Automated Coin Toll Service (ACTS) call in OSSAIN preprocessing. Preprocessing occurs before the call is connected to the operator or automated system. After preprocessing, the SN can request that the switch either terminate the call (for example, if the calling party goes on hook) or resume TOPS processing to an operator or automated system.

The following additional CO types are allowed for OSSAIN preprocessing calls:

- alarm
- mobile
- attended pay station
- CAMA
- 121, 131, 141, 151, 161, 171, 181, 191

- 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162

Two tables require datafill to allow OSSAIN preprocessing: table CT4QNAMS and table OAINPRE. After the switch performs CT4Q refinement, a call that meets the criteria for OSSAIN preprocessing uses the final CT4Q assignment to index table OAINPRE. From OAINPRE, a function provided by an SN is determined and a session with the SN begins.

Preprocessing selector in table CT4QNAMS

The SYSAREA field in table CT4QNAMS specifies whether OSSAIN preprocessing occurs for certain TOPS calls. When the SYSTEM subfield is set to TOPSOPR and the OAINPRE subfield is set to Y, the CT4Q is eligible for OSSAIN preprocessing. Datafill is also required in the OPRSYS subfield, which lists the set of TOPS operator systems (OPR, MCCS, ACTS, ADAS, ALL, NONE) that can receive OSSAIN preprocessing.

The following figure shows sample tuples in table CT4QNAMS. In the example, calls refining to index 0 (1+COIN) can receive OSSAIN preprocessing for ACTS calls or operator calls. Calls refining to index 1 (DA_411) can receive OSSAIN preprocessing for operator calls.

Figure 72 MAP display example for table CT4QNAMS

```
CT4QNUM CT4QNAM  OAMA  ITRIDIDX SYSAREA
-----------------------------------------------------------------------
0       1+COIN    N       N        TOPSOPR   N         N  Y ACTS OPR $ N
1        DA_411    N       N         TOPSOPR   Y 411_OPR  N  Y OPR $ N
```

Table OAINPRE

Table OAINPRE defines an OSSAIN function for each CT4Q name assigned to TOPS calls that receive OSSAIN preprocessing. Each CT4Q datafilled in table OAINPRE first must be assigned in table CT4QNAMS. The following figure shows sample tuples in table OAINPRE.

Figure 73 MAP display example for table OAINPRE

```
CT4Q                          NETWRKID       OAFUNCNM
-------------------------------------------------------------------
1+COIN                        200            SN_TOLL_BRAND
DA_411                        201            SN_DA_BRAND
1+HOTEL                       202            HOTEL_OAINPRE
```

OSSAIN preprocessing call control

During the call control stage, the switch performs requests from the SN. The SN can perform only a subset of OAP operations, as follows:

- connect and release a voice circuit

- assign a service provider identifier (SPID)
- send an OAP context data block when the SN releases the call
- terminate or resume the call

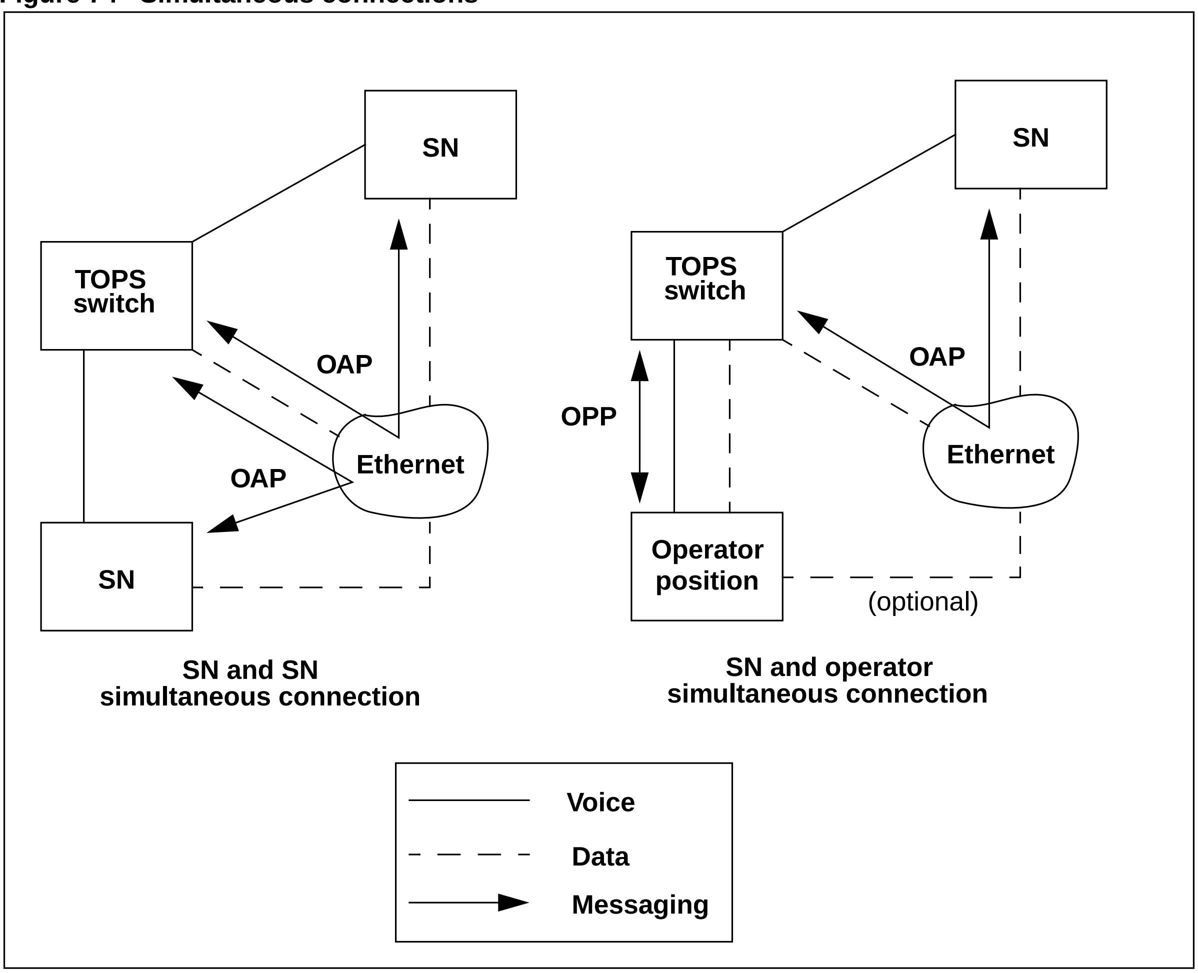
Simultaneous connections

In a simultaneous connection, two function providers are connected to a single call at the same time. OSSAIN allows an SN to connect to a call simultaneously with another SN or with an operator. (This is similar to the connection of an ADAS node and an operator in an ADAS call.)

Note: Although an existing TOPS automated system is considered a function provider in the OSSAIN environment, it is *not allowed* to be an agent in a simultaneous connection.

The following figure shows two different simultaneous connection scenarios: between two SNs and between an SN and an operator.

Figure 74 Simultaneous connections



Sample interactions with two service providers

The following examples of interactions between service providers show how a simultaneous connection could be used.

SN and operator interaction

A DA call is routed to an SN for automated directory assistance service. The SN prompts the subscriber for city and name, and records the information. The SN initiates a simultaneous interaction with an operator. Once the operator is connected, the SN plays the recorded message and releases itself from the call. The operator provides the service.

SN and SN interaction

A call arrives at an SN. The SN offers the subscriber a menu of services, all of which are provided by another SN. The subscriber selects one of the services. The SN initiates a simultaneous interaction with the second SN. While the first SN waits to connect to the second SN, it plays an announcement to the subscriber. Once the second SN is connected, the first SN releases itself from the call. The second SN provides the service.

Agents in a simultaneous connection

In a simultaneous connection, the function providers are referred to as agents. While two agents are connected to the call, only one agent may control the flow of the call. The controlling agent is the *active* agent; the other agent is the *passive* agent.

Although either an SN or an operator can be an agent in a simultaneous connection, the active agent *always* must be an SN. An operator may not take control of the call while an SN is attached. (However, an operator is allowed to *release* an active SN and become the sole agent on a call. Refer to "Passive agent operations.")

Active agent operations

The active agent can perform the following OAP operations:

- Attaching a passive agent to the call, which allows the active SN to initiate a simultaneous connection by requesting that an operator or second SN be attached to the call.
- Detaching a passive agent from the call, which allows the active SN to end a simultaneous connection by requesting that the operator or second SN be detached from the call. This action also cancels a queued request for a passive agent.
- Passing control of the call to a passive agent, which allows the active SN to pass control of the call to another SN (but not to an operator). The SN that passes control becomes the passive agent.
- Releasing itself from the call, which allows the active SN to end a simultaneous connection by detaching itself from the call. The formerly passive operator or SN now controls the call.

- Exchanging messages with the switch, known as *pass-through messaging*, which allows agents to communicate using messages passed through the switch. (See “Pass-through messaging” on page 158.)
- Requesting a call disposition, which allows the active SN to request a transfer or call float. Before the disposition is performed, the switch releases the passive agent from the call.

Passive agent operations

The passive agent in a simultaneous connection can be either an SN or an operator. The passive agent does not control the flow of the call and can perform only a restricted set of operations.

If the passive agent is an SN, the following OAP operations are available:

- Connecting a voice link, which allows the passive SN to add a voice connection on the call.
- Releasing the voice link, which allows the passive SN to release its voice connection on the call.
- Releasing from the call, which allows the passive SN to drop off the call and end the simultaneous connection.
- Performing pass-through messaging

If the passive agent is an operator, the following actions are available:

- Releasing from the call, which allows the operator to drop off the call and end the simultaneous connection.
- Releasing the active SN, which allows the operator to obtain control of the call by ending the simultaneous connection.
- Performing pass-through messaging

Other than calling and called hook status, the passive agent in a simultaneous connection is not informed of changes in the status of the call. However, if the active agent passes control to the passive agent, the passive agent receives an update at that time. This update may consist of only the call information that has changed, or it may consist of complete call details.

Pass-through messaging

Both active and passive agents can use pass-through messaging. Pass-through messages are blocks of data sent in the OPP message (between the switch and an OPP-compatible position) or the OAP message (between the switch and an SN).

These two-way messages allow agents in the simultaneous connection to exchange *small* amounts of information while on the call. For example, the information could include updated context data or IP addresses.

Note 2: For details on OAP, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1; for details on OPP, please refer to *Open Position Protocol Specification*, Q214-1.

Simultaneous connection flow

The simultaneous connection comprises four steps:

1 Initiating—an SN requests a passive agent

2 Connecting—the passive agent is connected to the call

3 Processing—the two service providers perform processing simultaneously

4 Terminating—the simultaneous connection ends

Initiating

When the SN servicing a call determines that a simultaneous connection is needed, the SN sends a request for a passive agent to the switch in an OAP message. This message, a Function Provider Request, contains the name of the function that the passive agent must provide (as defined in table OAFUNDEF).

If the passive agent is available immediately, the switch sends it an OAP session begin message (if the agent is an SN) or an OPP call begin message (if the agent is an operator).

If the passive agent is not available immediately, the request is queued and the switch notifies the active SN. While the request is queued, the active SN can still perform operations, including canceling the request. When the passive agent becomes available, the switch notifies the active SN and the simultaneous connection begins.

If a passive SN cannot be obtained because of a deflection, overflow, or invalid agent type (such as an automated system), the switch sends the active SN an error message.

If a passive operator cannot be obtained because of deflection, overflow, or datalink failure, then alternate hosting may be utilized to select an operator from a different OC host switch. This can be setup in table OCHOSTQ as provided by feature NC0152. The active SN is not informed when alternate hosting is occurring; it is only informed of the final result. If a passive operator cannot be obtained from either the initial or the alternate host, then the switch sends the active SN an error message. However, if the initial request fails but the alternate host request succeeds, then the active SN will receive only the success response.

Note: If using alternate hosting, ensure any appropriate SN timers are increased to account for the two requests.

Connecting

When a passive operator is connected to a call, all parties by default have two-way speech. With OAP 9.0, however, an SN's request for a passive agent can include instructions for the switch to alter the speech path when the passive agent is connected. This eliminates any time delay between connecting a passive operator and processing a speech path change request. An SN may use this functionality to split the calling party and provide an announcement to the operator at call arrival.

Processing

During this step, the two agents provide the required service to the call. The agents can perform any of the following operations during simultaneous processing:

- The active and passive agent can use pass-through messaging.
- The active agent can pass control to the passive agent (SN only).
- The SN can connect or release a voice link.

Terminating

During this step, the simultaneous connection ends and only one agent remains to control the call. The agents can perform any of the following operations to terminate the connection:

- The active agent can release itself from the call.
- The passive agent can release itself from the call.
- The active agent can release the passive agent from the call.
- The passive agent (operator only) can release the active agent.
- The active agent can request a call disposition.
- The switch can end the call.

Note: Prior to SN07, the switch always sent complete call details to an operator position when an SN was released from a simultaneous connection with the position. Most of the call details were unchanged from the time when the position was initially connected to the call. As of SN07, the switch attempts to optimize messaging bandwidth utilization by not sending complete call details to the position if the switch is able to determine that the call details have not changed.

Operator handoff

Calls that are eligible for handoff (for automated alternate billing) by an operator can be sent to an SN to obtain billing acceptance. The SN or operator can specify the number of operator-to-SN handoffs allowed on a per call basis.

ALT_BILL_HANDOFF_METHOD parameter

The ALT_BILL_HANDOFF_METHOD parameter in table OAINPARM specifies the system to which the operator hands off the call. Valid values are AABS (the default, but not supported) or OSSAIN. Calls that are handed off to an OSSAIN SN are routed using the CT4Q specified in the parameter.

This parameter does not change the checks for eligibility of a call for handoff. Also, it does not change any screen displays or the way the operator marks a call for handoff.

Note 1: If an operator successfully marks a call for handoff and also marks the call for transfer through an OGT keying action, the handoff takes priority.

Note 2: Refer to Chapter 7: "OSSAIN data schema" for an example of table OAINPARM.

VSNOPT parameters

The VSNOPT parameter NUM_AUTO_LANGS_REQD_FOR_HANDOFF applies equally to attempts to handoff to AABS and to OSSAIN. For either AABS or OSSAIN handoffs, the operator receives the same error display when a check for eligibility fails.

Note: For an example of table VSNOPT, refer to the *Translations Guide*.

Service provider identifier (SPID)

A service provider identifier (SPID) is a code that uniquely identifies the service provider of the originating party. The TOPS switch processes two types of SPIDs, as follows:

- The account owner (AO) SPID identifies the service provider of the subscriber. The AO SPID can be assigned using switch datafill.
- The billing service provider (BSP) SPID identifies the provider of billing services (such as AMA records) for the AO service provider. The BSP SPID cannot be assigned using switch datafill.

SPID processing includes assigning the calling AO SPID, branding the call using the AO SPID, providing the SPID on operator screen displays, and recording the SPID on the AMA record.

TOPS call processing obtains SPIDs in the following ways:

- through OAP messaging with an OSSAIN SN
- through an OLNS query
- through switch datafill (calling AO SPID only)

Note 1: A SPID obtained through OAP messaging takes precedence over a SPID obtained from an OLNS query or from switch datafill.

Note 2: For more information on the switch datafill for SPID as well as restrictions for SPID, please refer to *TOPS Unbundling User's Guide*, 297-8403-903.

Obtaining the SPID through OAP

OAP messaging allows the switch and SN to exchange SPID information, if available. When the switch initiates a session with an SN, the switch sends SPID information to the SN. The switch also sends SPID information when the SN requests call details.

Likewise, when the SN initiates a session with the switch, the SN sends SPID information to the switch. The SN can send SPID information at any time during a session.

Note 1: The OM group OAPCALP5 includes registers that count SPID assignment requests, successes, and errors. Refer to Chapter 12: “OSSAIN operational measurements.”

Note 2: For complete information on OAP operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

TOPS Local Number Portability (LNP)

Local Number Portability (LNP) is a circuit switched network capability that allows telephone subscribers to keep their directory number (DN) when they change service providers. The subscriber keeps the same DN when the DN is moved, or *ported*, to a different end office. Other subscribers can connect to the ported DN without changing their dialing procedure.

LNP changes the way that TOPS translates, routes, and signals numbers. In LNP processing, the switch determines whether a calling, called, or billing DN is portable. If it is, the switch may have to send an LNP query to the SCP database to determine whether the DN has been ported, and if it has, the location routing number (LRN) of the recipient switch. The switch needs the LRN to route the call and to provide information about the DN for the AMA billing record.

The OSSAIN environment supports LNP processing. OSSAIN does not change the criteria TOPS uses to make LNP queries. The SN can request and receive LNP information (using OAP) from the switch for the calling, called, or billing numbers in an OSSAIN call.

Note 1: The OM group OAPCALP5 includes registers that count LNP requests, successes, and errors. Refer to Chapter 12: “OSSAIN operational measurements.”

Note 2: For complete information on the TOPS LNP capability, please refer to *TOPS LNP User’s Guide*, 297-8403-902.

Release link trunking (RLT)

Release link trunking (RLT) increases the capacity of ISUP trunks by releasing ISUP connections between a previous DMS-250 switch and a TOPS switch. After RLT is performed, ISUP connections are released, which makes circuits available for additional traffic. Without RLT, ISUP connections must be maintained between the TOPS switch and the previous switch for the duration of the call.

RLT phased deployment

RLT between the TOPS switch and the DMS-250 switch is deployed in three phases. These phases can function simultaneously in the TOPS switch. Trunk datafill in table ISUPTRK (ISUP Trunk) determines which phase of RLT is performed on each trunk.

This section provides a background on how TOPS and OSSAIN implement the phases of RLT. For complete information on how to setup RLT, please refer to the *Translations Guide*.

RLT phase I

RLT phase I requires a software load of UCS06 or greater in the DMS-250 switch. At the TOPS switch, the SOC option code OSEA0102 controls RLT phase I. Phase I applies to the following set of call types:

- collect billing
- third number billing
- station paid billing

RLT phase II

RLT phase II implements ISUP Universal Carrier Protocol (UCP) functionality and requires a software load of UCS08 or greater in the DMS-250. At the TOPS switch, the SOC option code OSEA0102 controls RLT phase II. Phase II applies to the following set of call types:

- collect billing
- third number billing
- station paid billing
- calling card billing

Phase II also allows trigger analysis and call sequencing.

RLT Wireless

RLT Wireless can be applied in table ISUPTRK via options RLT_IS41 or RLT_GSM for calls originating from corresponding offices. At the TOPS switch, the SOC option code OSEA0104 controls RLT Wireless along with table TOPSFTR parameter RLT_FOR_IS41. OSSAIN supports informing a SN if a call is eligible for wireless RLT and will process a wireless RLT request for DACC. In addition to the following eligibility checks, wireless RLT requires the original called digits be TOPS TLDN digits.

Eligibility checks

To be eligible for RLT, a call must pass a set of eligibility checks. All eligibility checks in the following list must pass for *all* phases of RLT:

- The appropriate SOC option is turned ON.
- Valid calling and called numbers are present.
- The incoming trunk supports the correct variant of RLT.
- If an outgoing party is connected, the outgoing trunk supports the correct variant and version (phase) of RLT.
- If an outgoing party is connected, the previous DMS-250 switch contains a connection for both call legs. (This is verified by checking the point codes of incoming and outgoing trunks.)
- A conference ID is not present for OSSAIN calls.
- The class charges are not unspecified.
- The call is not a hotel or coin call.
- The call is not an intercept call.

- The call is not a delay call.
- The call is not headed to treatment.

Version screening

Because all phases of RLT can exist in the same office, the TOPS switch must screen the version to ensure stability between the phases. The following eligibility checks need version screening:

- Trunks that are datafilled with RLT phase II (version 2 in table ISUPTRK) have the OSEA0102 SOC option turned ON.
- If the call is in a bridging scenario, both the incoming and outgoing trunks support the correct version.
- If phase I is enabled, RLT cannot be used for calling cards. If phase II is enabled, RLT can be used for calling cards.
- If phase I is enabled and a trigger profile index is present in the call, trigger analysis is not performed. If phase II is enabled and a trigger profile index is present, trigger analysis is performed.
- RLT wireless as defined by RLT_IS41 and RLT_GSM in table ISUPTRK require the original called digits be TOPS TLDN digits. It also requires SOC option OSEA0104 and TOPSFTR RLT_FOR_IS41.

Types of RLT

The following types of RLT are supported for TOPS and OSSAIN calls:

- *RLT bridging* occurs after the TOPS switch establishes a forward connection to the previous DMS-250 switch. Before requesting RLT, if a forward party is present, the TOPS switch checks the point codes of the trunks to ensure that both call legs have a connection in the previous switch. After the point codes are validated, the TOPS switch requests the previous switch to bridge the two calls together (as in a collect call).
- *RLT transfer* occurs before the TOPS switch establishes a forward connection. The TOPS switch signals routing information to the DMS-250, and the DMS-250 completes the call to the called party (as in a calling card call).

ISUP messages supporting RLT

The following ISUP messages support RLT:

- The Facility Request (FAR) message is sent by the TOPS switch and requests that the previous switch perform RLT.

  Note: In Phase II, the FAR message is also used to signal a 100-byte UCP context block to the DMS-250. This block stores information needed for sequencing.
- The Facility Accept (FAA) message is sent by the previous switch and indicates that it successfully processed the RLT request.

- The Facility Reject (FRJ) message is sent by the previous switch and indicates that it rejected the TOPS request to perform RLT.

ISUP messaging exchange

If RLT applies to the call, the TOPS switch sends an ISUP FAR message to the DMS-250 switch, which requests a transfer or bridge connection. If the DMS-250 switch can provide the appropriate connection, it returns an ISUP FAA message to indicate RLT success. Once the TOPS switch receives the FAA message, the ISUP connections are released between the two switches.

If the DMS-250 switch is unable to perform RLT, it returns an ISUP FRJ to indicate RLT failure. In this case, the TOPS switch maintains the connection and connects the forward party if needed.

In phase II of RLT, a sequencing capability exists. If an OSSAIN SN requests RLT, the eligibility checks determine whether or not the call can sequence. If it can, an additional ISUP FAR/FAA exchange is required to signal the 100-byte UCP context block to the DMS-250 switch. This exchange occurs *before* the TOPS switch sends the ISUP FAR message requesting RLT.

Upon re-origination, another ISUP FAR/FAA exchange retrieves the context block from the DMS-250 switch. The TOPS switch retrieves any trigger information from the context block, and routes the call according to trigger datafill.

ISUP 100-byte UCP context block

The following information is not changed on the re-origination of a sequenced RLT call:

- class charge
- account code
- custom PIN length
- billing number
- billing number type
- billing number digit count
- sequence number
- trigger profile index
- branding status
- language information
- OSSAIN context block

Call timing

The switch performs a calculation of call timing and records it on the AMA billing record generated for the call. Call timing includes the answer time and conversation time for the call. The switch uses OAP messaging to inform the SN of call timing values so that the SN can perform precise billing activities for the call.

If calls are floated prior to answer, the switch takes a time stamp when answer occurs (the parties are connected). This time stamp indicates the start of conversation. If calls are floated after answer and conversation time begins, the switch continues to accumulate conversation time for the call. The only exception to this rule are calls that are billed station collect or person collect. In these collect cases, the switch overwrites the answer time stamp at the point the call is floated.

Conversation time is the elapsed time calculated for a call. It begins at answer time and ends when one or both parties disconnect. Conversation time also can be reported during an event that interrupts call timing during the talking phase of the call. The switch provides these call timing values in a single data block to the SN.

Note: For information on how the talking state is affected by trigger events, refer to a description of the talking trigger profile in Chapter 2: “OSSAIN software functionality.” Also refer to Chapter 7: “OSSAIN data schema,” for information on the stop conversation (STOPCNVT) option in table OATLKPRF.

GEN AMA with called party connected

Calls in the talking state at the SN can have an AMA record generated while the called party is still connected. When AMA is generated at the SN with the called party connected, the time of the request is used as the disconnect time for the AMA record. The time of the request is also used as the connect time on the subsequent AMA record.

Example OAP message flow

Prepaid card service is an example of an SN application that may use call timing information. This service allows billing to a calling card that has a preset amount of conversation time already paid by the subscriber. The card account is debited by the amount of conversation time used in each call.

The OAP message flow between the switch and SN follows these basic steps:

- The SN providing the prepaid service sends the prepaid card number to the switch as a custom card type.
- The SN validates the card number.

  Note: The switch does not validate prepaid cards through a calling card validation (CCV) query.

- At answer, the switch sends the answer time stamp to the SN.
- At disconnect, the switch sends the answer time stamp and the conversation time to the SN.

Note: For complete information on OAP operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

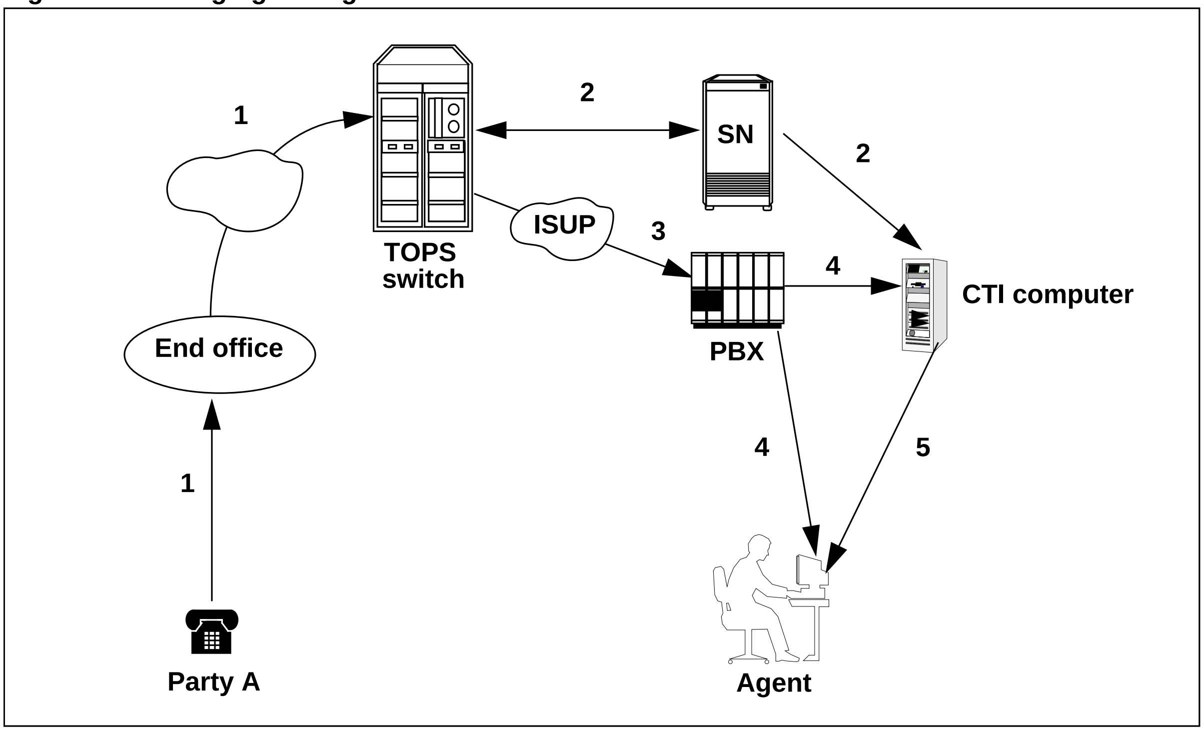
Outgoing ISUP

An SN can change the DN in the Calling Party Number (CgPN) parameter in an outgoing ISUP initial address message (IAM). The SN can use this capability to coordinate network level and node level processing.

Sample scenario

The following figure shows a simple network call center environment. In the scenario, the SN collects call context information at the network level for a call that is forwarded to a call center agent at a private branch exchange (PBX). The steps are described after the figure.

Figure 75 Changing the CgPN in a network call center environment



1 Party A dials a number for a call center service or agent. The call is routed to the TOPS switch.

2 The SN is brought into the call to collect customer information. After collecting the information, the SN requests the switch to use a particular DN in the CgPN parameter of the outgoing ISUP IAM to the call center PBX. At the same time, the SN sends the customer information, which includes the new CgPN, to the CTI server associated with the agent that handles the call.

3 The switch makes the forward connection by sending the IAM over the ISUP trunk to the PBX. The CgPN parameter in the IAM contains the DN supplied by the SN.

4 The PBX routes the call to the appropriate agent and at the same time sends the calling number it extracted from the IAM to the CTI server.

5 The CTI server coordinates the calling number with the information it received earlier from the SN. This information is forwarded to the agent to use in serving Party A.

Voice link broadcasting (VLB)

In OSSAIN voice processing, the SN establishes a unique, two-way network voice connection whenever it needs to obtain voice or DTMF input from the parties in a call. However, if the SN does not need to collect any information from the parties but merely needs to play a generic announcement or background music, the SN can establish a *broadcast* voice link instead.

With voice link broadcasting (VLB), the SN can use a single one-way broadcast voice link that can connect up to 1023 different calls at the same time. This capability saves voice link facilities between the DMS switch and the SN.

This section discusses the following topics:

- example VLB scenarios
- datafilling broadcast voice links
- maintenance of broadcast voice links
- provisioning broadcast voice links

Example VLB scenarios

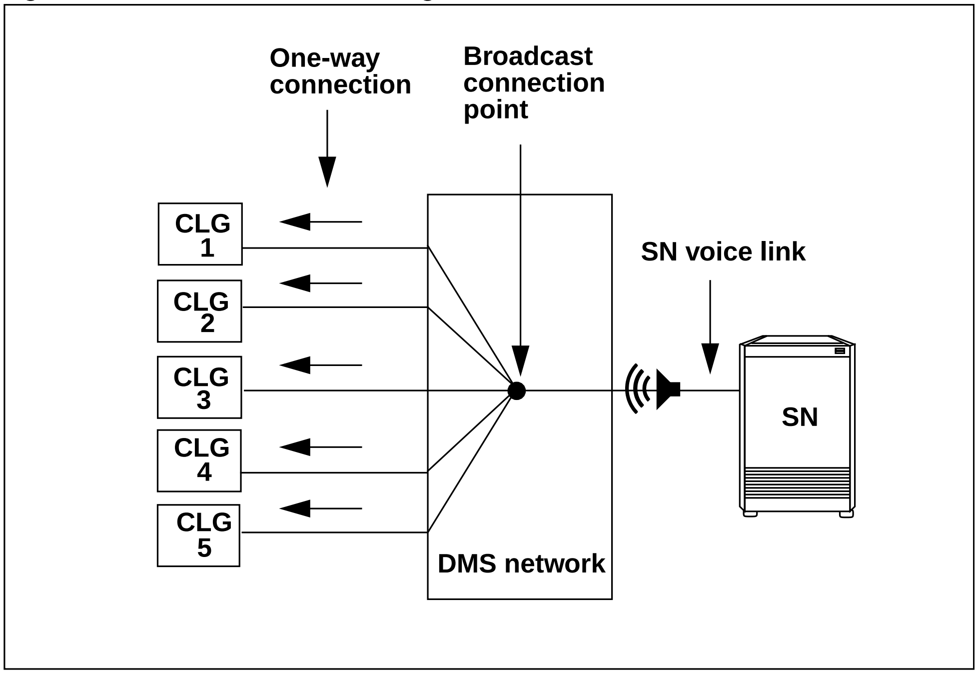
This section describes VLB scenarios for both standalone OSSAIN and for centralized OSSAIN (OSAC).

Standalone broadcast voice connections

Consider the following scenario: A call that has a data connection to an SN is waiting for an available agent. During this wait, the SN requests the call to be connected to a voice link that continuously plays music. With VLB, the switch makes a one-way connection between the SN voice link and the calling party, effectively providing music while the call is on hold.

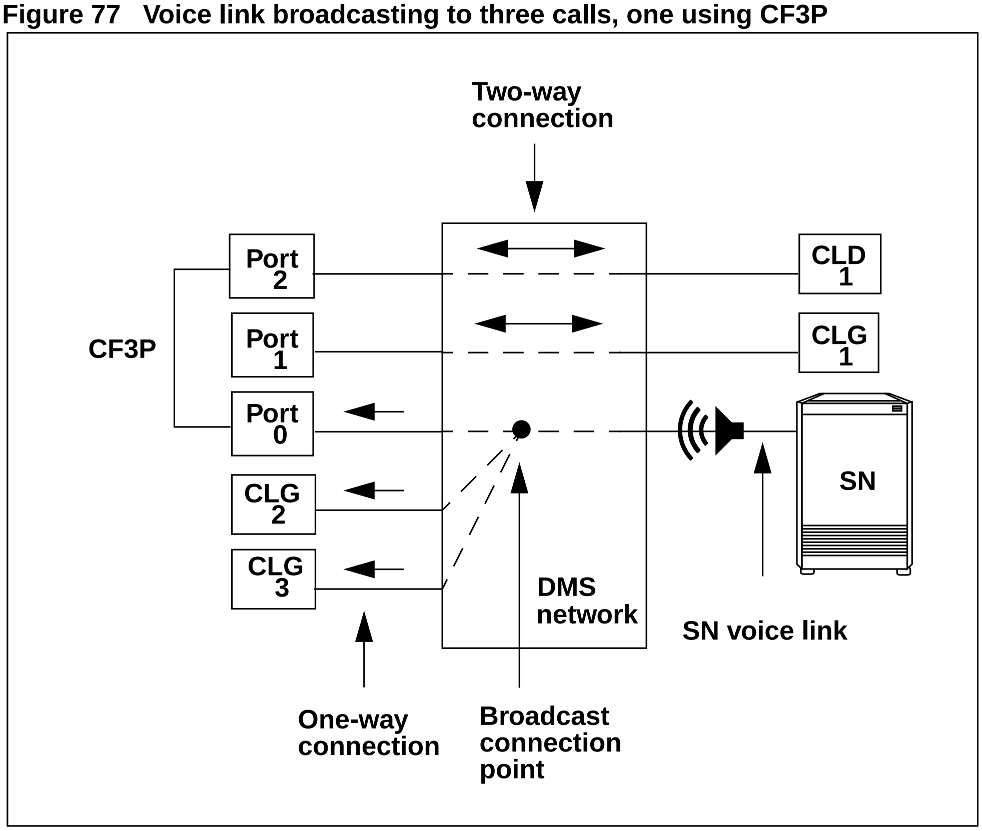
If several calls are waiting for an agent, every call can be connected to the same broadcast voice link simultaneously, instead of to separate voice links. The following figure shows an example of how the network connections might look in this scenario. In the example, a single SN broadcast voice link provides music to five different callers at the same time. Each call has a one-way connection to the voice link.

Figure 76 Voice link broadcasting to five calls



The following figure shows how three calls are connected to a single broadcast voice link. One of the calls uses a conference three port (CF3P) circuit.

Figure 77 Voice link broadcasting to three calls, one using CF3P

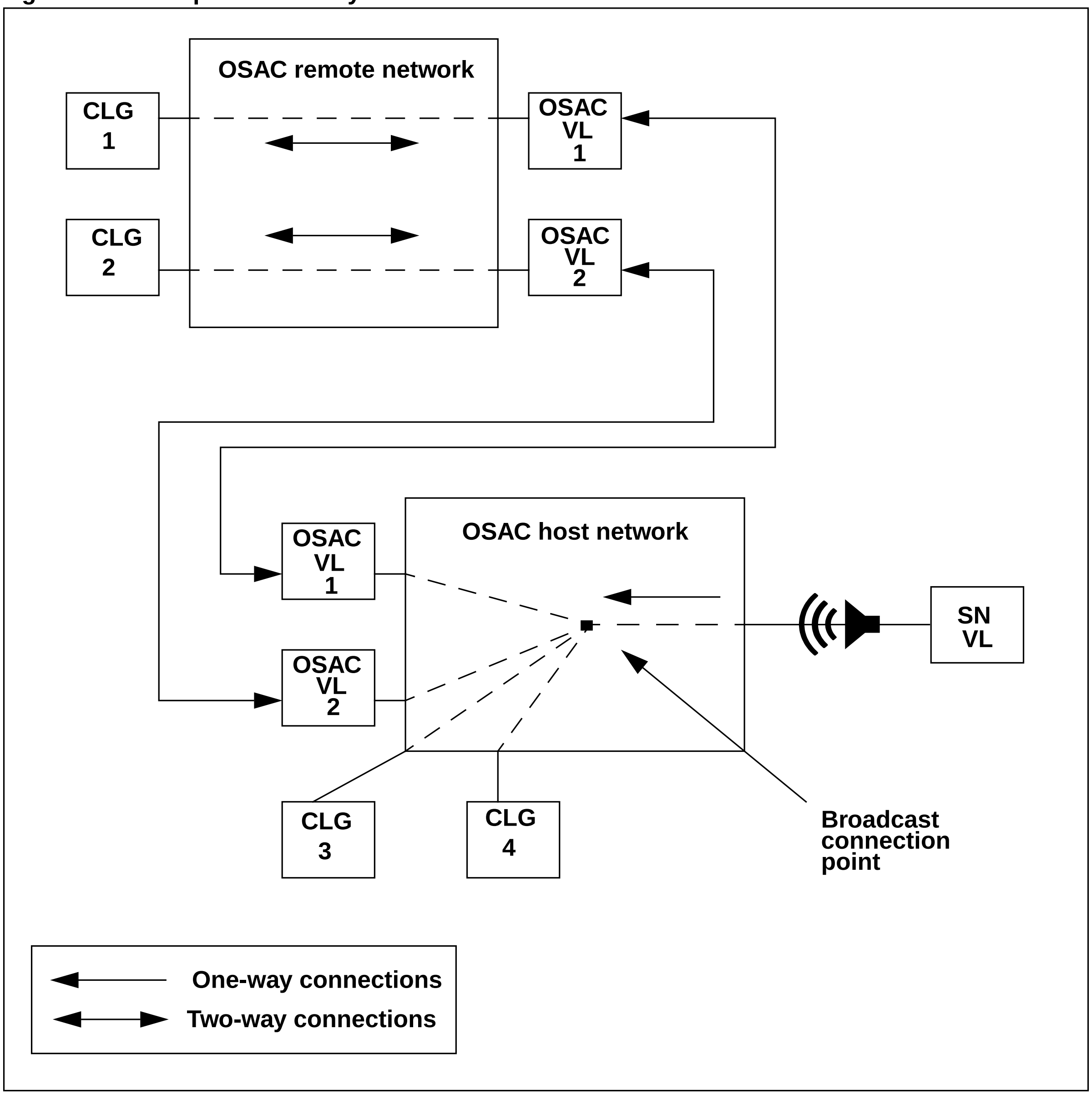


OSAC broadcast voice connections

Broadcast voice connections are supported in the OSAC environment. Only the voice links between a standalone or OSAC host switch and an SN (SN voice links) can be set up for broadcasting in table OAVLMAP. Broadcasting is *not* allowed for voice links between the OSAC host and remote switches (OSAC voice links).

The following figure shows how voice link broadcasting works in OSAC.

Figure 78 Example of one-way broadcast voice connections with a centralized SN



In Figure 78, the OSAC host switch also functions as a standalone switch. Four separate calls are shown connected to a single broadcasting SN voice link. Two of these calls originated in the OSAC remote switch (calls 1 and 2) and two originated as standalone calls in the OSAC host switch (calls 3 and 4).

Note: Broadcasting for the OSAC remote still requires an individual OSAC voice link for each call.

A request for a voice connection of a call in the OSAC remote to an SN voice link in the OSAC host is sent from the SN to the remote. This request includes the logical voice circuit to be used (which is the logical voice circuit between the SN and the OSAC host). The OSAC remote forwards the message, including the SN ID, to the OSAC host. The OSAC host checks the information against datafill in table OAVLMAP to obtain the physical trunk group member of the voice link.

Successful requests result in the OSAC host selecting an outgoing OSAC voice link and making the appropriate one-way network connection between it and the SN voice link. At the same time, the OSAC host tells the OSAC remote which incoming OSAC voice link is to be used. The OSAC remote then makes a connection (always two-way) in its network between the OSAC voice link and the party or parties involved in the call.

Datafilling broadcast voice links

The SN is responsible for selecting and supplying the logical voice channel number and the node ID for the voice connection. The DMS switch uses datafill in table OAVLMAP (OSSAIN Voice Link Mapping) to associate the logical voice channel number with an index in table TRKMEM (Trunk Members).

For broadcast voice links, datafill in the BCSTAREA field in table OAVLMAP specifies the method to establish the voice path, the maximum number of simultaneous connections to a particular broadcast voice link, and whether specialized tone receiver (STR) supervision applies. This section describes the subfields and refinements in the BCSTAREA field.

Note: A voice link can function as only one type: either non-broadcast or broadcast.

BCST_SEL

When the BCST_SEL subfield is set to Y (yes), the trunk is used for broadcasting one-way voice communication from an external announcement device to multiple calls simultaneously. If BCST_SEL is set to Y, datafill in the CUTTHRU, MAXCONNS, and STRCLG refinements is also required. When the BCST_SEL field is set to N (no), the trunk is used for non-broadcast, two-way voice communication.

Note: For OSAC voice links in both the host and the remote, the BCST_SEL field in table OAVLMAP is set to N. This restriction is enforced by DMS switch table control. Only the SN voice links in the OSAC host switch can have the BCST_SEL field set to Y.

CUTTHRU

The CUTTHRU refinement determines the method for pulse code modulated (PCM) cut-through (voice path). After the switch makes the network voice connection, it enables the transmission PCM samples of voice signals from the broadcast voice link. The CUTTHRU refinement specifies one of two methods, as follows:

- IMMEDIATE—PCM cut-through is enabled when the voice connection is made. This method works well for playing background music, where the caller does not need to hear the beginning of the recording.
- HKCHG (hook change)—PCM cut-through is disabled until the switch detects an off hook on the voice link of at least 10 milliseconds. This method works well for playing announcements, where the caller needs to hear the beginning of the recording.

MAXCONNS

The MAXCONNS refinement specifies the maximum number of simultaneous connections that can be made to a voice link. For Junctored Network (JNET) offices, the range is from 2 to 255. For Enhanced Network (ENET) offices, the range is from 2 to 1023.

Note: DMS switch table control enforces the JNET limit of 255. If CUTTHRU is set to HKCHG, an ENET limit of 255 is also enforced.

STRCLG

The STRCLG refinement specifies whether STR supervision is performed on the calling party while the call is connected to the broadcast voice link. If the STRSEL subfield is set to Y, datafill in the STRDIGIT subfield is also required. STRDIGIT must be datafilled with either STAR (*) or OCTO (#).

The following figure shows example datafill in the BCSTAREA.

Figure 79 MAP display example for table OAVLMAP (standalone)

```
NDANCH              CLLI          EXTRKNUM    BCSTAREA
----------------------------------------------------------------------
BILLING_SN 1        BILLING_VL    1           N
BRANDING_SN 2       BRANDING_VL   2           Y  HKCHG 50 Y STAR
CALL_CENTER_SN 5    CALL_CTR_VL   5           Y  IMMEDIATE 800 N
```

Note: For details on all the fields in table OAVLMAP, refer to Chapter 7: “OSSAIN data schema.”

Maintenance of broadcast voice links

With broadcast voice links, the SN may use the hook status of a voice channel to indicate to the switch when to enable PCM cut-through. An off-hook received from an idle broadcast voice link causes the link to be taken out of service. (This operation is the same as for non-broadcast voice links.)

A broadcast voice link must return to the on-hook state when not in use, regardless of the last state it was in while connected to a call. If a broadcast voice link remains off hook when it has been released from a call, the switch takes it out of service.

Provisioning broadcast voice links

The requirements for provisioning broadcast voice links depend on whether the office is equipped with JNET or ENET.

ENET offices

In ENET offices, broadcast voice links are assigned by the same rules used for standard, non-broadcast voice links. At most, 1023 simultaneous connections can be made to a broadcast voice link in an ENET office.

Note: In table OAVLMAP, if CUTTHRU is set to HKCHG, the limit for MAXCONN is 255 simultaneous connections.

JNET offices

In JNET offices, 255 simultaneous connections at most can be made to a broadcast voice link. Also, certain engineering guidelines must be followed to ensure that no more than 256 (for NT0X48 networks) or 512 (for NT5X13 or NT8X11 networks) simultaneous connections (broadcast and non-broadcast) occur on a network subgroup (NSG).

Note: Refer to Appendix C: “Provisioning broadcast voice links in a JNET office,” for provisioning details.

OSSAIN conferencing

An SN can use one of two methods to configure a conference call at the DMS switch:

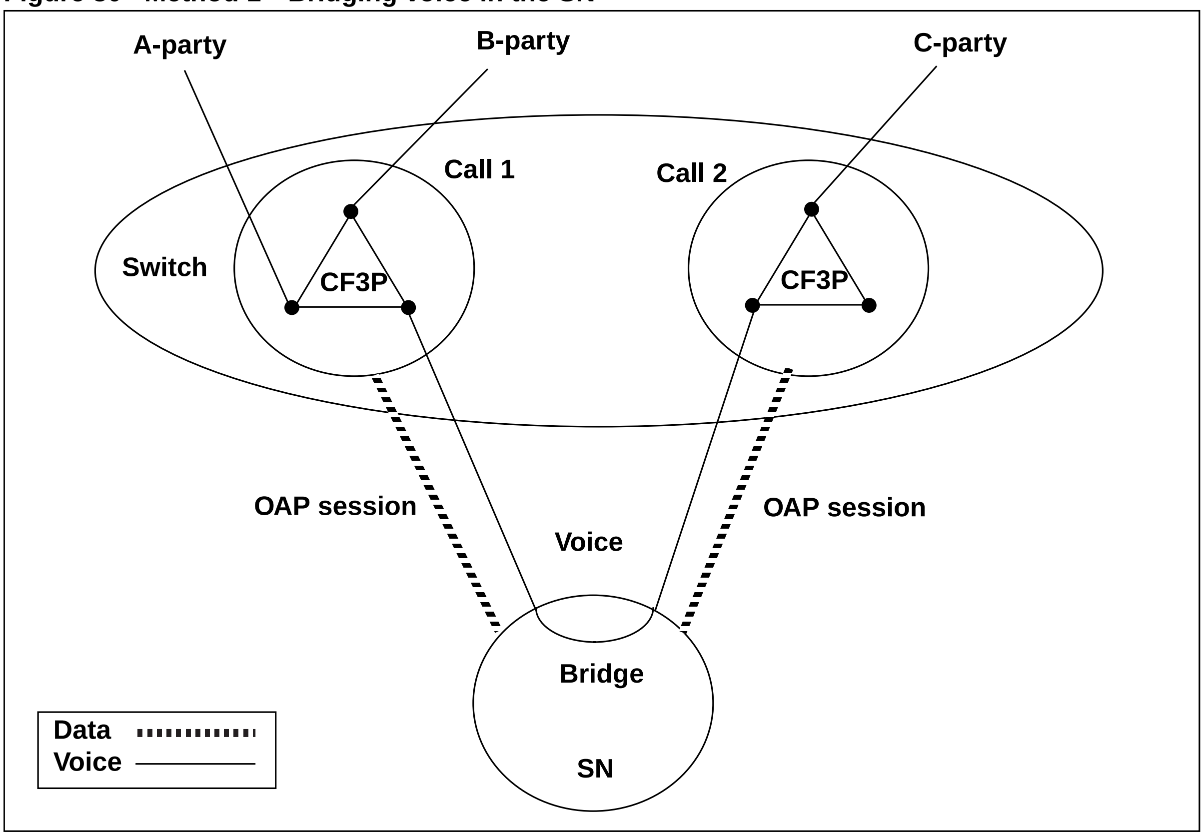
- Method 1—bridging the voice in the SN
- Method 2—bridging the voice in the switch

Method 1

With Method 1, the SN establishes a voice link to each call and bridges the voice. This method uses two voice links and two OAP sessions for the duration of the call. It also uses two conference three port (CF3P) circuits.

The following figure illustrates Method 1.

Figure 80 Method 1—Bridging voice in the SN

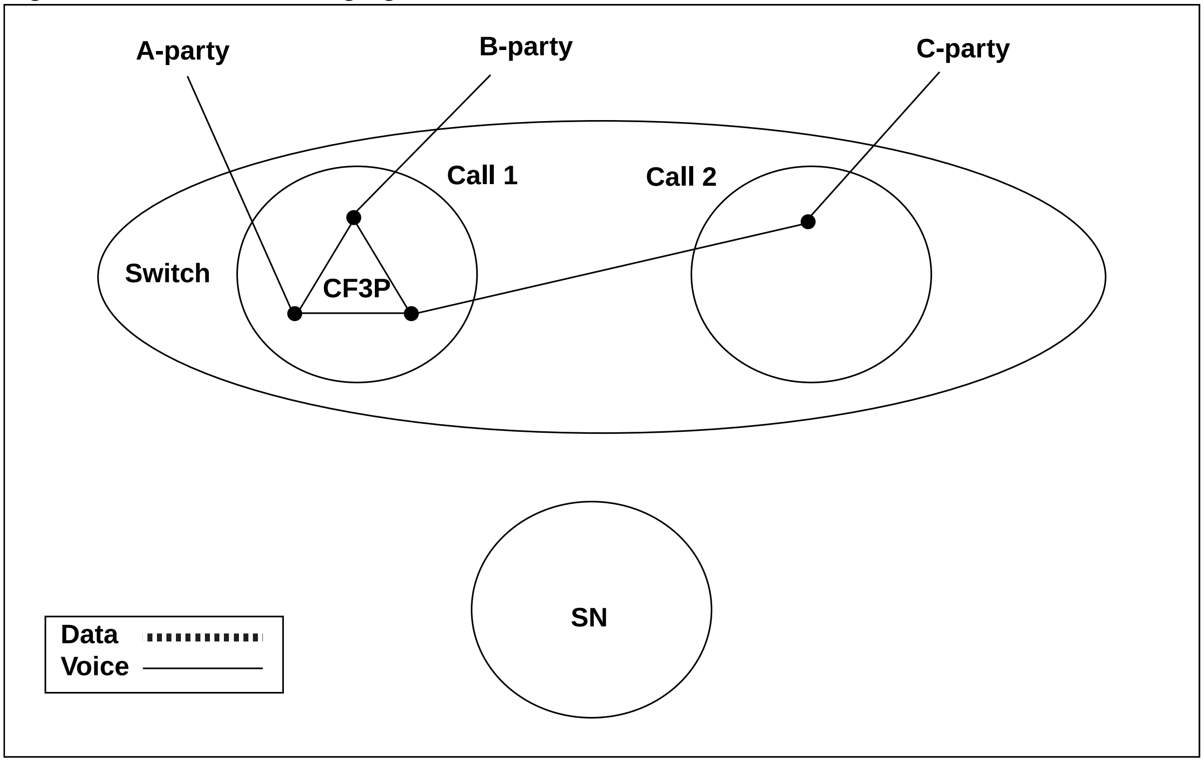


Method 2

OAP messaging allows the SN to create conferences and move parties in and out of conferences. With Method 2, the SN can connect three subscribers simultaneously and float both calls, which releases the voice link and OAP sessions. This method requires *only one* CF3P circuit.

The following figure illustrates Method 2.

Figure 81 Method 2—Bridging voice in the switch



Note: For complete information on OAP messaging, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

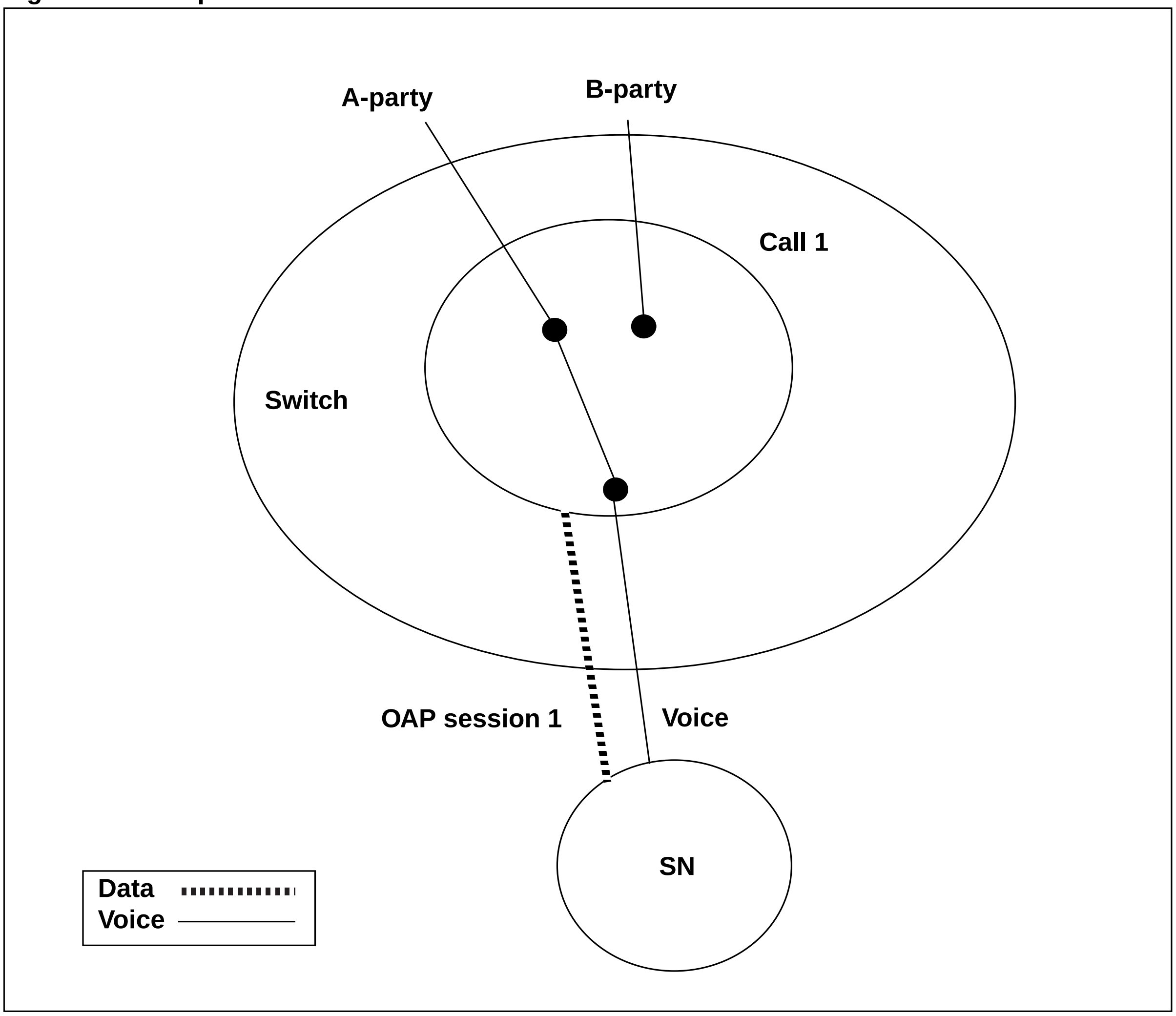
This section shows how the SN creates a three-way call. The scenario begins with a two-party (A-party and B-party) floated call. The following four steps are described:

- setup
- third leg
- consult
- conference and float

Setup

In the conference setup, the A-party uses the telephone keypad to trigger an event. This trigger causes the switch to send a Session Begin Inform message to the SN. The SN responds by placing the B-party on hold and connecting a voice link to the A-party to collect digits for the consult part of the scenario. The following figure shows the setup step.

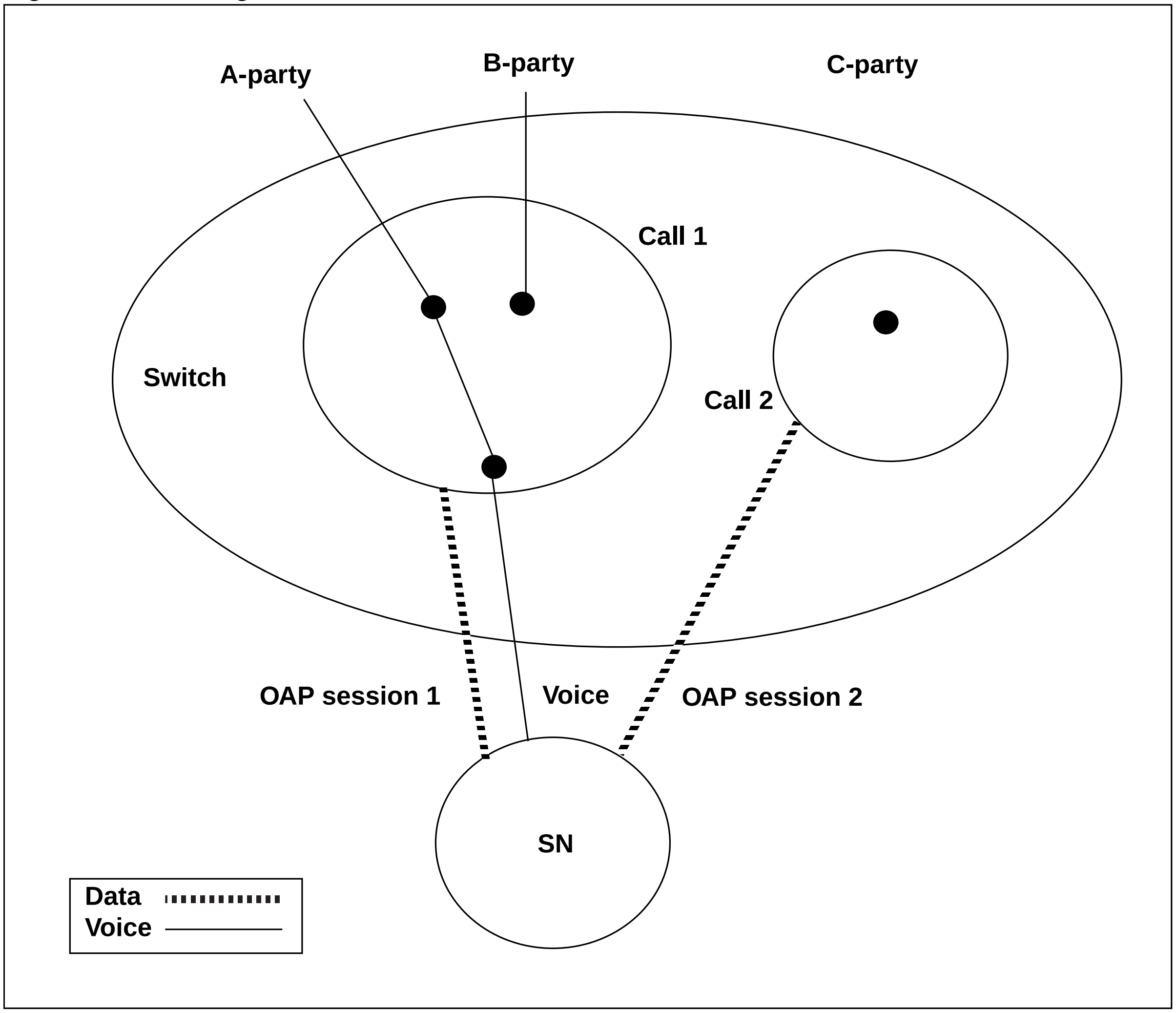
Figure 82 Setup



Third leg

After the SN collects enough digits from the A-party, it begins a new session to call 2. The following figure shows the third leg step.

Figure 83 Third leg

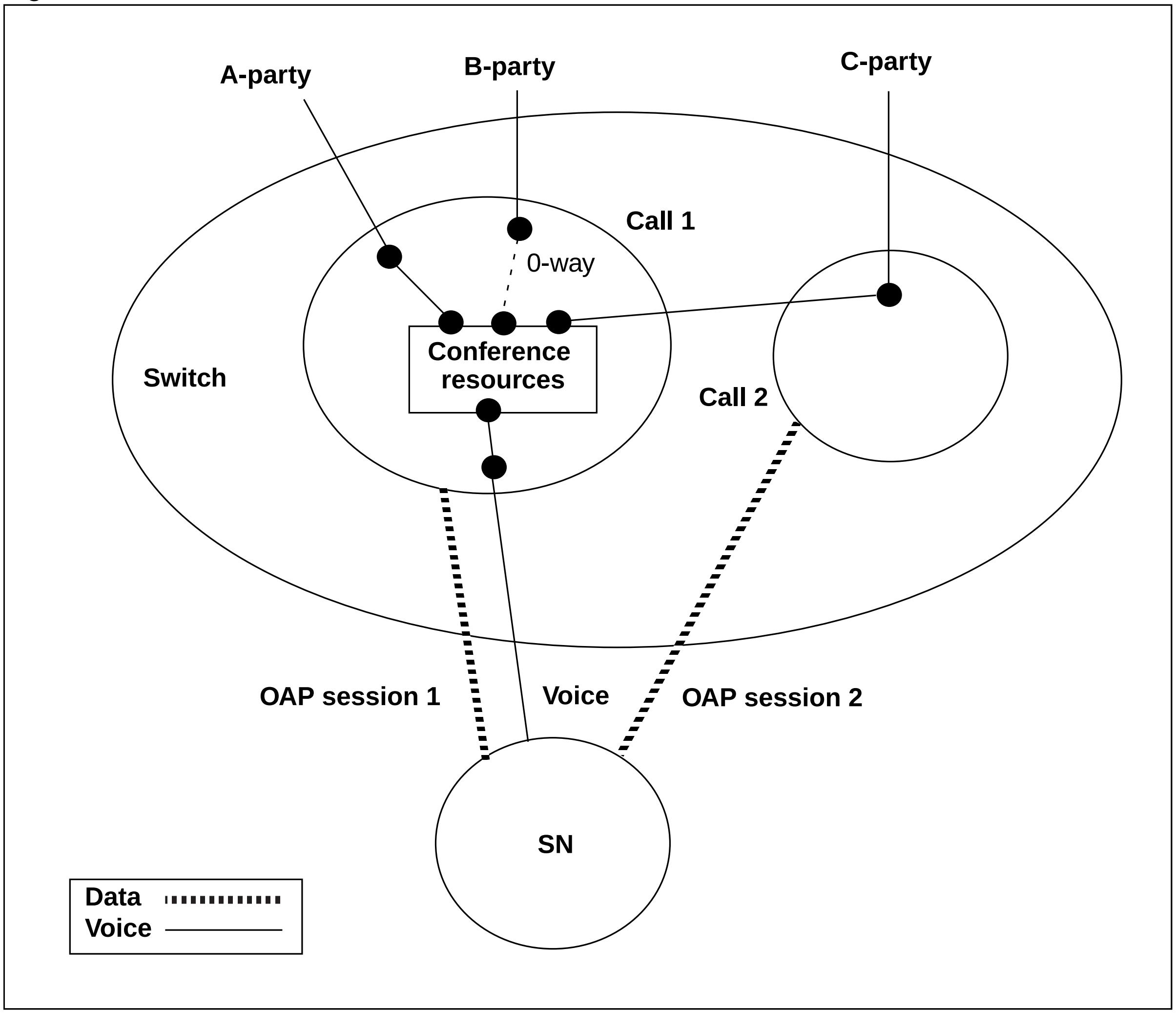


Consult

Using OAP session 1, the SN creates a conference, adding the A-party, the B-party, and the SN voice link to the new conference. Using session 2, the SN requests the DMS switch to outpulse to the C-party (which is the B-party of call 2).

The OAP message sent by the SN supplies an optional data block that indicates the C-party should be added to the conference. The A-party and C-party can now talk to each other. The following figure shows the consult step.

Figure 84 Consult

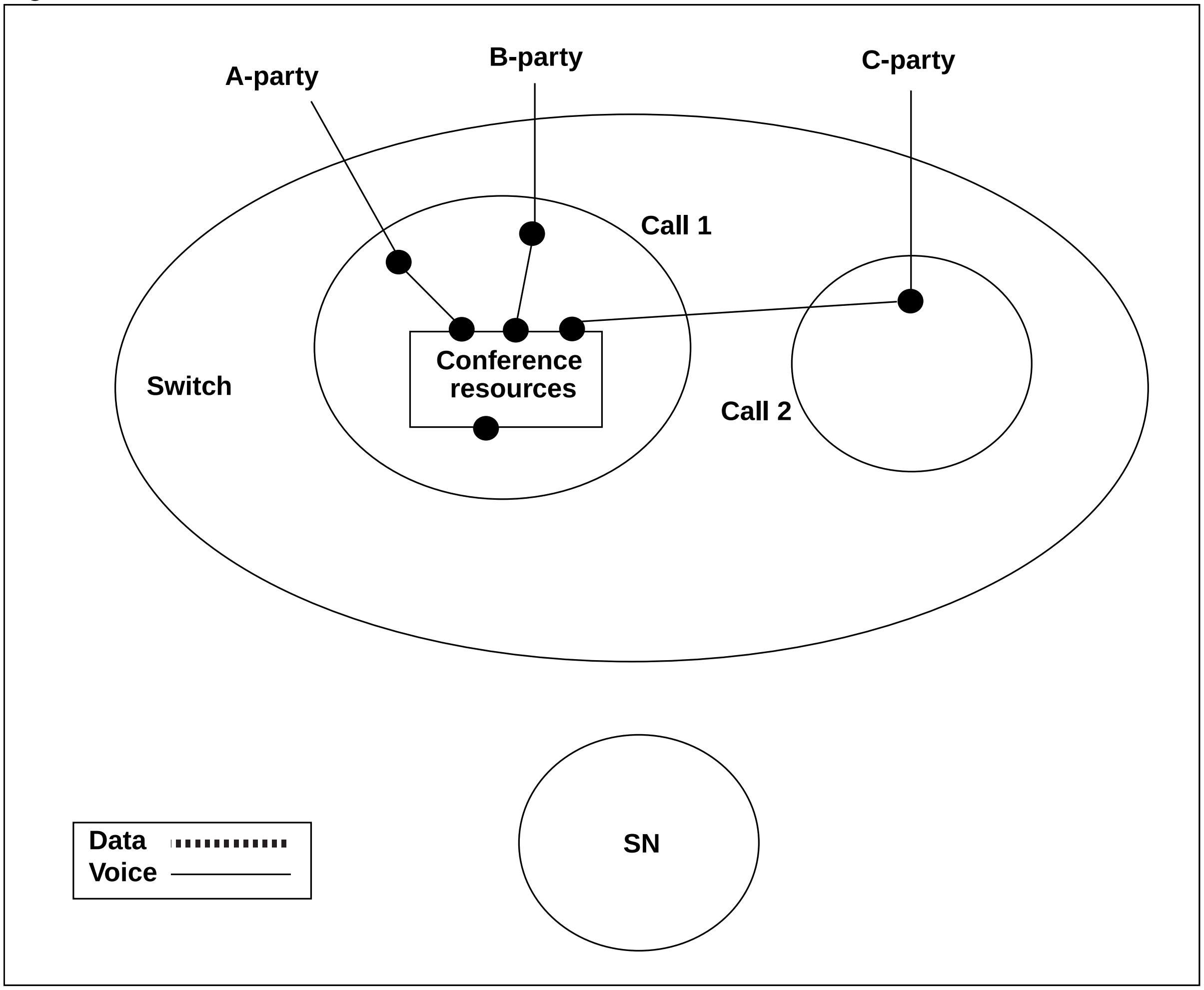


Note: *Conference resources* refers to the generic conference resources that the DMS switch appropriately allocates and deallocates.

Conference and float

The SN allows the B-party to participate in the conference. The SN requests to float call 1 on OAP session 1 and to float call 2 on OAP session 2, specifying two-way speech for all parties. The following figure shows the conference and float step.

Figure 85 Conference and float



Four-way conference

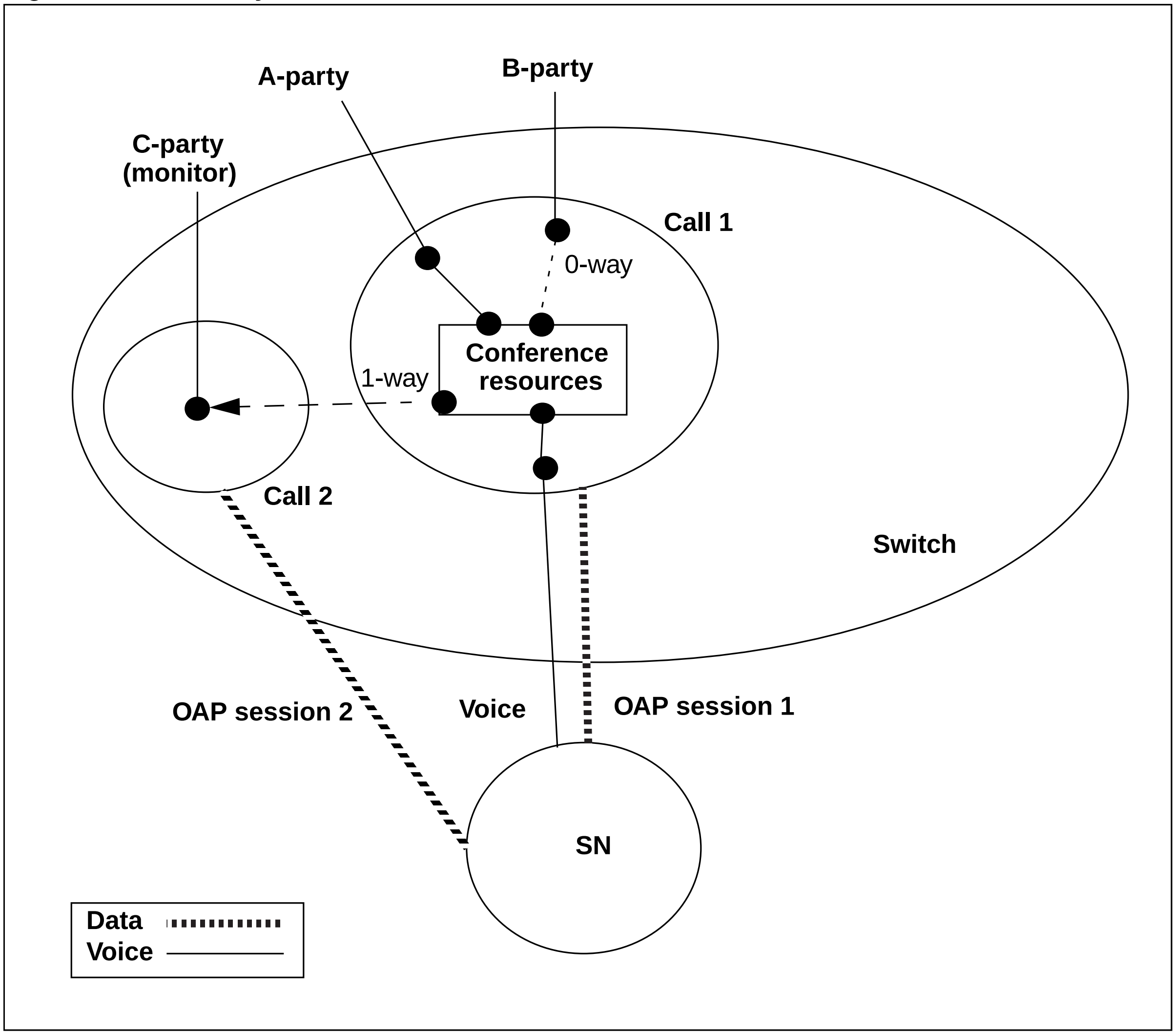
Four subscribers can be connected to an OSSAIN conference simultaneously when no SN voice link is connected. There are many ways that a four-way conference can be established. This section shows one example scenario where a monitoring agent has a one-way connection to a conference for the purpose of monitoring a call center agent. The following three steps are described:

- three-way conference
- fourth leg
- four-way conference

Three-way conference

The A-party, B-party, and C-party (monitor) are in the same conference. The B-party is on hold, while the C-party has a one-way connection (listen only) and is monitoring the call. The following figure shows the three-way conference step.

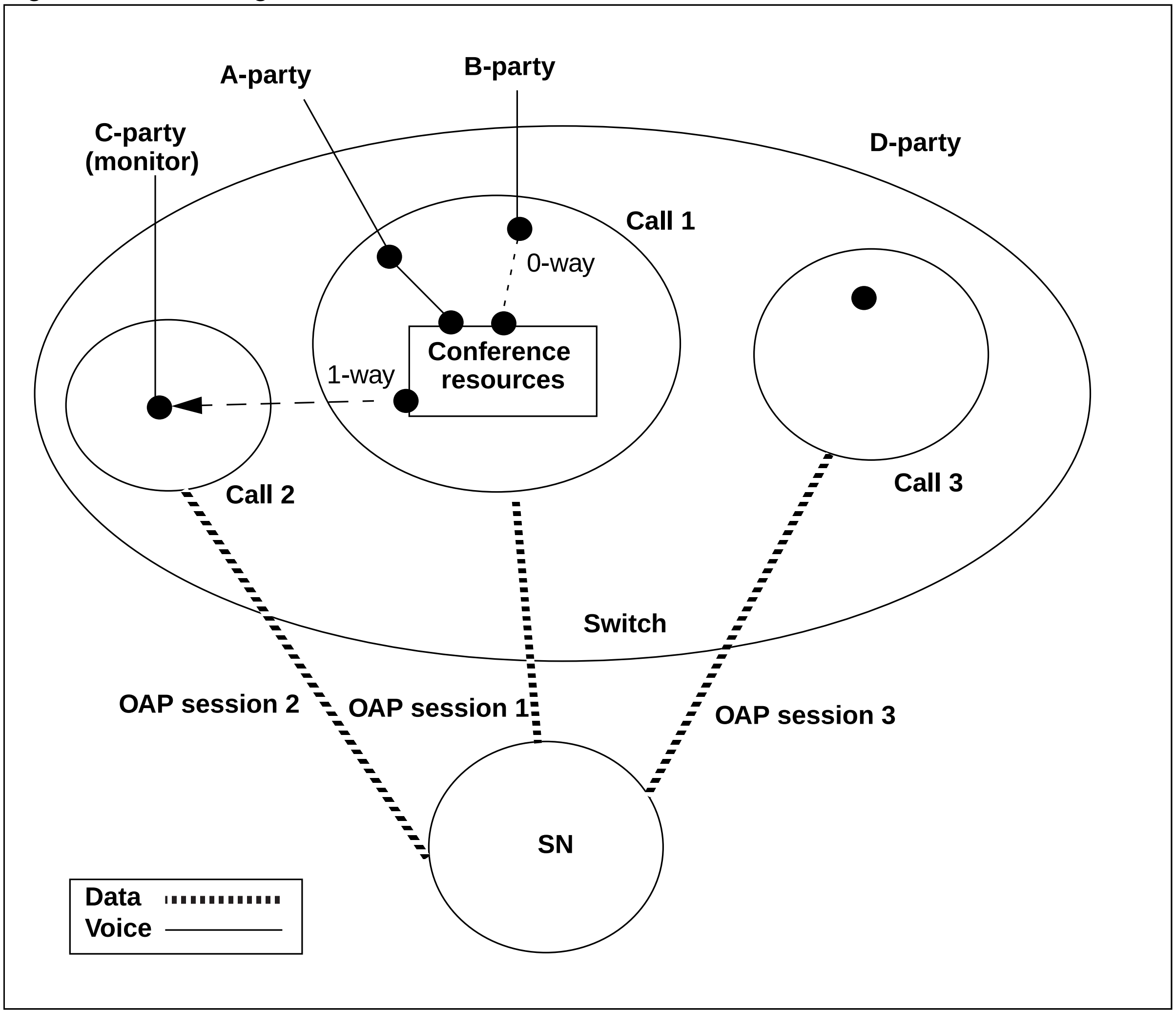
Figure 86 Three-way conference with monitor



Fourth leg

After the SN collects enough digits from the A-party, it removes the voice link from the conference and begins a new session to call 3. The following figure shows the fourth leg step.

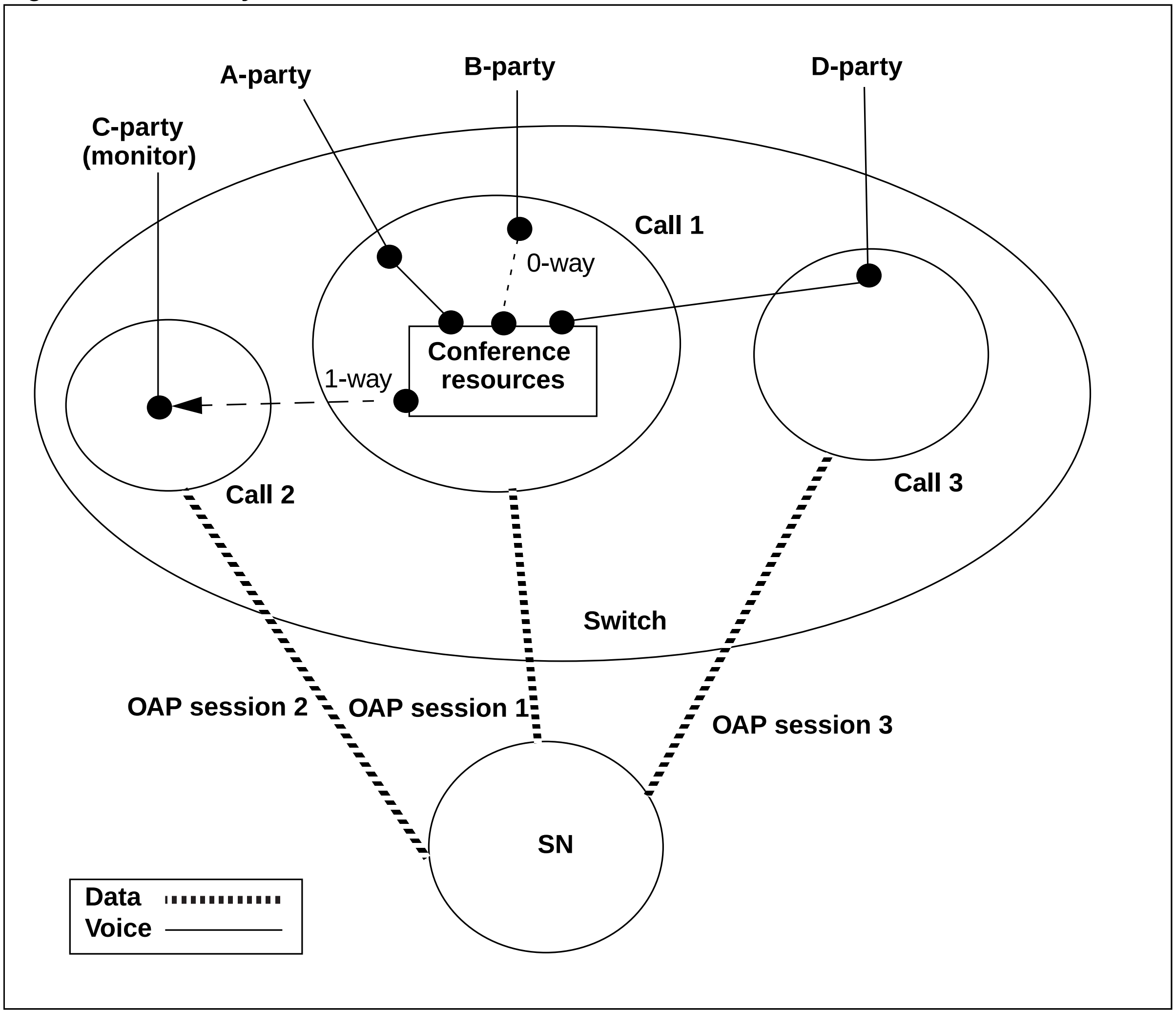
Figure 87 Fourth leg



Four-way conference

Using OAP session 3, the SN requests the DMS switch to outpulse to the D-party (which is the B-party of call 3). The OAP message sent by the SN supplies an optional data block that indicates the D-party should be added to the conference. The A-party and D-party can now talk to each other. The C-party remains in the conference to monitor the call. The following figure shows the four-way conference step.

Figure 88 Four-way conference with monitor



Country direct

OSSAIN supports international inbound country direct functionality. In the OSSAIN environment, country direct calls bypass the operator to be serviced by an OSSAIN SN. The SN collects the terminating number and billing information, then releases the call to the TOPS switch for standard call completion.

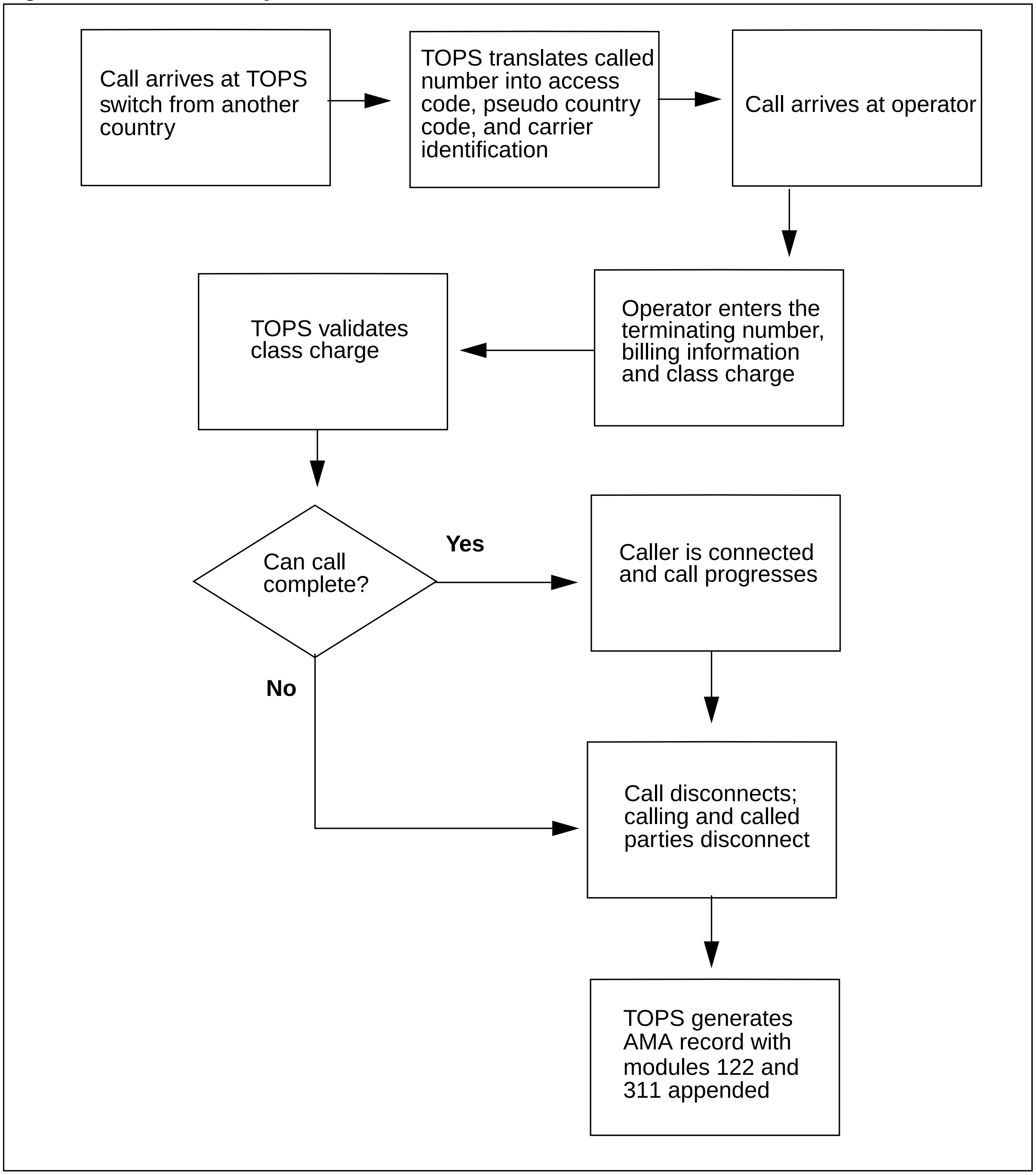
This section briefly describes TOPS country direct functionality and shows how country direct works in OSSAIN.

Note: For complete details on TOPS country direct, please refer to the *Translations Guide*.

TOPS country direct

With TOPS country direct, a caller in a foreign country can dial an access code to reach an operator in the country in which the call is to be billed (usually the home country). The following figure shows the flow of a TOPS country direct call.

Figure 89 TOPS country direct flow



The flow of TOPS country direct follows these steps:

1 Screening—After the digits are received, they are parsed as defined by table datafill. For example, assume that 171-123-0003 is a country direct number. In this case, 171 is treated as an access code; 123 is treated as the pseudo country code; and 0003 is treated as the carrier identification. The country direct digit stream is sent to TOPS as a called number.

2 Call presentation—After the call is marked as country direct (CDIR) and the digit stream has been parsed, it is routed to the operator for call completion. The calling number is obtained from the country direct digit stream, or from a parameter in table TOPSPARM.

3 Call handling—After the call arrives at the operator position, the operator can process the call.

4 AMA—After the call disconnects, module codes 122 (Country Direct) and 311 (Originating Call Type) are appended to the AMA record, which shows the country of origination, terminating domestic number or country of termination, and billing information.

Related TOPS country direct datafill

Please refer to the *Translations Guide*, for details on the following datafill needed for TOPS country direct:

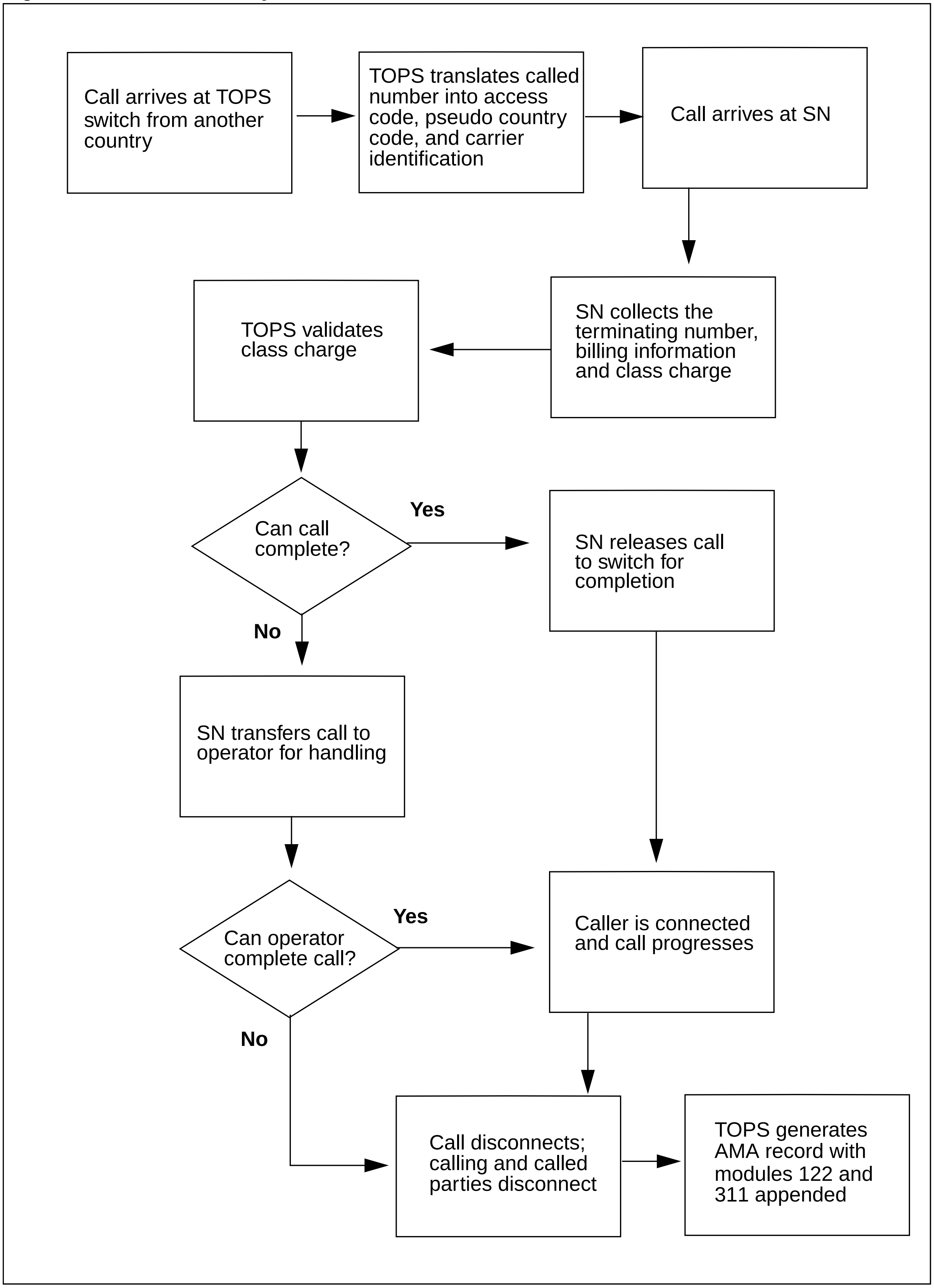
- TOPS R1 signaling in a North American environment
- TOPS R1 signaling in an Open Numbering environment
- TOPS R2 signaling
- parsing the access code, country code, and carrier of origin
- optional parameter CDIR_DEDICATED_DN in table TOPSPARM

OSSAIN country direct

With OSSAIN, country direct calls can go directly to the SN, using standard QMS queuing. The SN collects the terminating number and billing information. If the call can complete, the SN releases the call to the switch for completion. In case of an invalid entry and re-prompt, the SN can transfer the call to an operator, indicating that it is a country direct call.

The following figure shows the flow of an OSSAIN country direct call.

Figure 90 OSSAIN country direct flow



OSSAIN QMS MIS

The OSSAIN Queue Management System Management Information System (QMS MIS) is a switch application that collects event-driven data about OSSAIN calls and sends this data to an external reporting facility, or *MIS node*. The data is used to report statistics on the functioning of OSSAIN call queues and sessions.

This description focuses on the following OSSAIN QMS MIS areas:

- MIS node connectivity
- OSSAIN QMS MIS application
- MIS node maintenance

Note: OSSAIN QMS MIS is not enhanced to support SN07 OAP changes. It is slated for end of life in a future release.

MIS node connectivity

The OSSAIN QMS MIS application composes, buffers, and sends a stream of MIS messages to the MIS node. Communication between the DMS switch and the MIS node is over an IP network using the OSSAIN Open Automated Protocol (OAP). Data communications processing in the switch encapsulates the OAP MIS operation and data block buffer within an MIS class header in a TCP message.

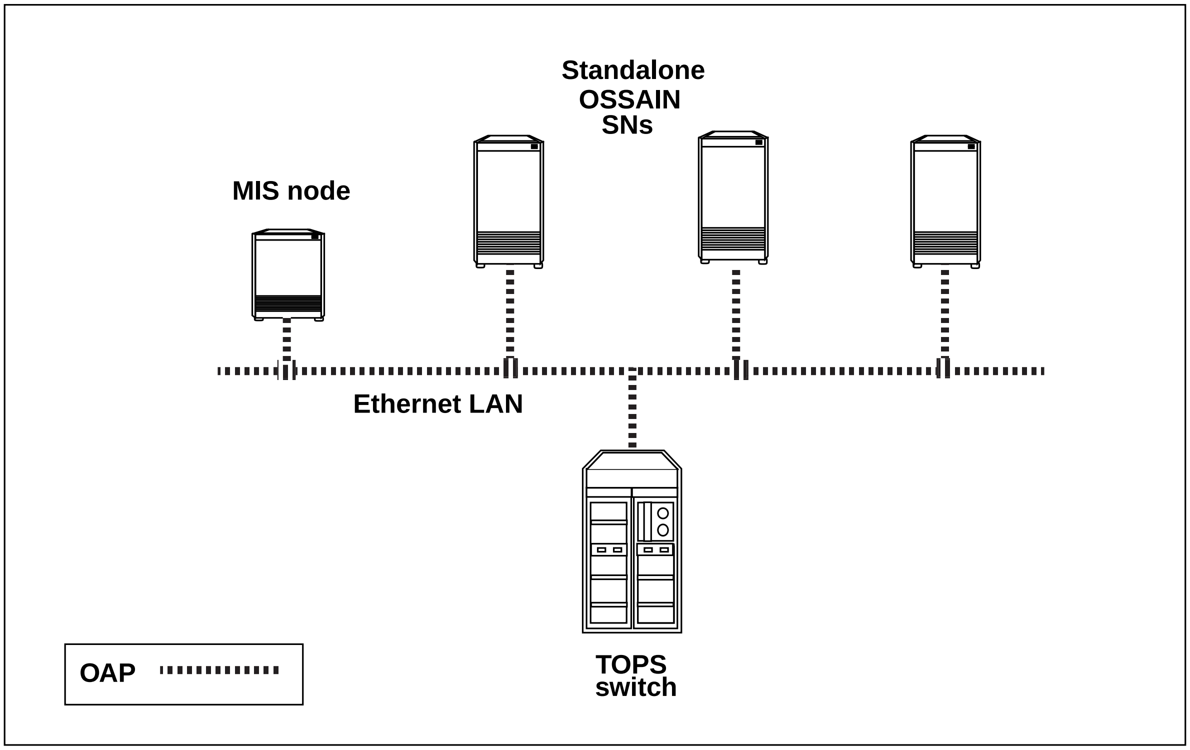
Note 1: For complete details on OAP and specific information on how it supports QMS MIS messaging, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Note 2: For more details on data connectivity, refer to Appendix A: “OSSAIN data communications.”

Standalone OSSAIN

For calls that originate in the standalone OSSAIN environment, the switch collects the MIS data and sends it to the MIS node. The following figure shows an MIS node in a simple standalone OSSAIN topology.

Figure 91 Simple standalone OSSAIN topology



OSAC

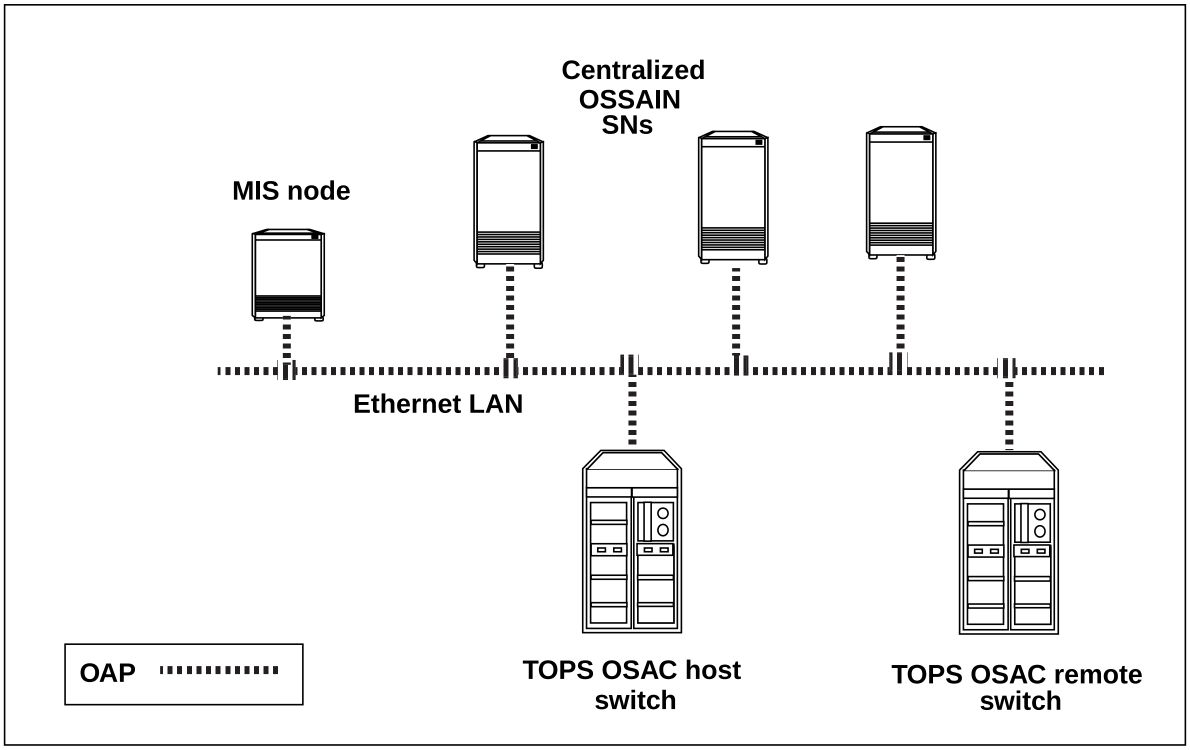
In a centralized OSSAIN (OSAC) environment, the QMS Call and Agent Manager (CAM) is located *only* at the OSAC host switch. So all QMS MIS messages about OSSAIN calls are composed, buffered, and sent by the OSAC host switch and *not* by the OSAC remote switch.

For calls that originate in the OSAC remote, the call information is communicated to the OSAC host from the OSAC remote. The OSAC host collects the data and sends it to the MIS node.

Note: Messaging between the OSAC remote switch and OSAC host switch uses the OSAC protocol, which is proprietary to Nortel. The *OSSAIN User's Guide* does not provide user information on the OSAC protocol.

The following figure shows an MIS node in a simple OSAC topology.

Figure 92 Simple OSAC topology



OSSAIN QMS MIS application

The existing TOPS QMS MIS application is unchanged and supports the QMS MIS position event and queue event messages for calls handled by operators.

Likewise, the OSSAIN QMS MIS application supports the QMS MIS *session* (agent) event and queue event messages for calls handled by SNs. Both applications are defined in table QMSMIS.

This section describes the following areas of the application:

- QMS MIS statistics
- pegging work volume and idle time
- buffering messages

QMS MIS statistics

The OSSAIN QMS MIS collects data on session events and queue events, as follows:

- Session events are generated to capture the state changes of each OSSAIN session. Session events are recorded for both call and non-call (such as maintenance) session transitions. Session events include:
  - the number of times an SN is presented with a new call in a given period
  - the number of times an SN initiates a new call in a given period
  - the number of times an SN is presented with a call as a result of a trigger event in a given period
  - the number of times an SN is presented with a call transferred from an operator or another SN
  - the number of times a call is released from an SN
- Queue events are captured as part of the processing performed when OSSAIN exchanges a message with the QMS CAM. Queue events include:
  - the number of calls placed in a queue
  - the number of calls removed from a queue
  - the number of calls presented to an SN
  - the number of queue deflections
  - the number of queue overflows

Pegging work volume and idle time

The QMS MIS session event message is used to record the availability of a session as it transitions between states. The session event message contains a timestamp that can be used to calculate the duration that a session was in a particular state.

Idle time and work volume pegs are reported in both call and non-call session events, as follows:

- Call Busy Work Volume (CBWV) is the amount of time from call arrival to a session to call release from the session.
- Non-call Work Volume (NCWV) is the amount of time a session is in service but not available to handle calls. Pegging of NCWV occurs for sessions that are drained or throttled.
- Idle Time (IDLT) is the amount of time a session is in service and available to handle calls.

So that the DMS switch can provide timestamps for session events, the events are reported *after* the event has been completed. For example, CBWV is pegged at the point the call is being released from a session. IDLT is pegged at the point an idle session is being assigned to a call.

Note: Peg events are reported only on in-service sessions.

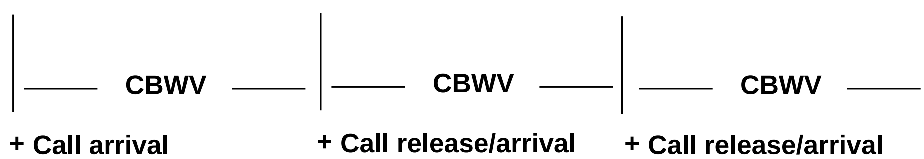
The following call flows illustrate work volume and time pegging for call arrival to an idle session, continuous call arrival, and draining or throttling a session pool.

- Call arrival to an idle session—When a call is presented to an idle session, the system begins accumulating CBWV, stops accumulating IDLT, and sends a request to peg IDLT. Likewise, when the call is released from the session, the system begins accumulating IDLT, stops accumulating CBWV, and sends a request to peg CBWV.

Figure 93 Call arrival to an idle session

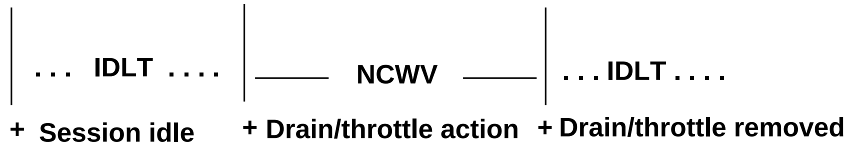
- Continuous call arrival—Continuous call arrival occurs when multiple calls are queued for a session. As a call is released, CBWV is pegged for that call. No IDLT is pegged.

Figure 94 Continuous call arrival



- Draining or throttling a session pool—NCWV begins accumulating when a session pool is drained or throttled and the session can no longer accept calls. NCWV is pegged as soon as the drain or throttling action is removed.

Figure 95 Draining or throttling a session pool



Buffering messages

The DMS switch maintains a single 4K message buffer for the OSSAIN QMS MIS application. The message buffer fills up with session event and queue event messages as they occur. After the buffer is filled, it is sent across the appropriate connection to the MIS node.

Buffering MIS messages saves real-time costs of sending data across the Ethernet connection as individual messages. During periods of high traffic, the buffers also reduce the risk of messages being discarded. However, during periods of *low* traffic volume, a significant amount of time may pass before a buffer becomes full. To keep the MIS system informed in a timely manner, the switch uses a timer to initiate the sending of the MIS message buffer even when it is not full. The QMS_MIS_OAIN_XMIT_TIMEOUT parameter in table OAINPARM determines the maximum amount of time before an OSSAIN OAP MIS buffer is sent. In periods of very low traffic (when no session event or queue event messages have been buffered), only an information message is sent to the MIS node.

Most OSSAIN call scenarios generate three OAP MIS data blocks, as follows:

- one call queue event data block, 44 bytes in length
- one session event data block, 16 bytes in length
- one non-call session event data block, 56 bytes in length

The following calculation can be used to determine the buffer transfer rate:

(average message bytes per call) * (calls per minute) / buffer size

Using these numbers, the average number of bytes per call is 116 (44+16+56). If the DMS switch services approximately 1000 calls per minute, an approximate buffer transfer rate is one buffer every 2.1 seconds, as shown in the following calculation:

116 * 1000 / 4K = 29 buffers per minute, or 1 buffer every 2.1 seconds

MIS node maintenance

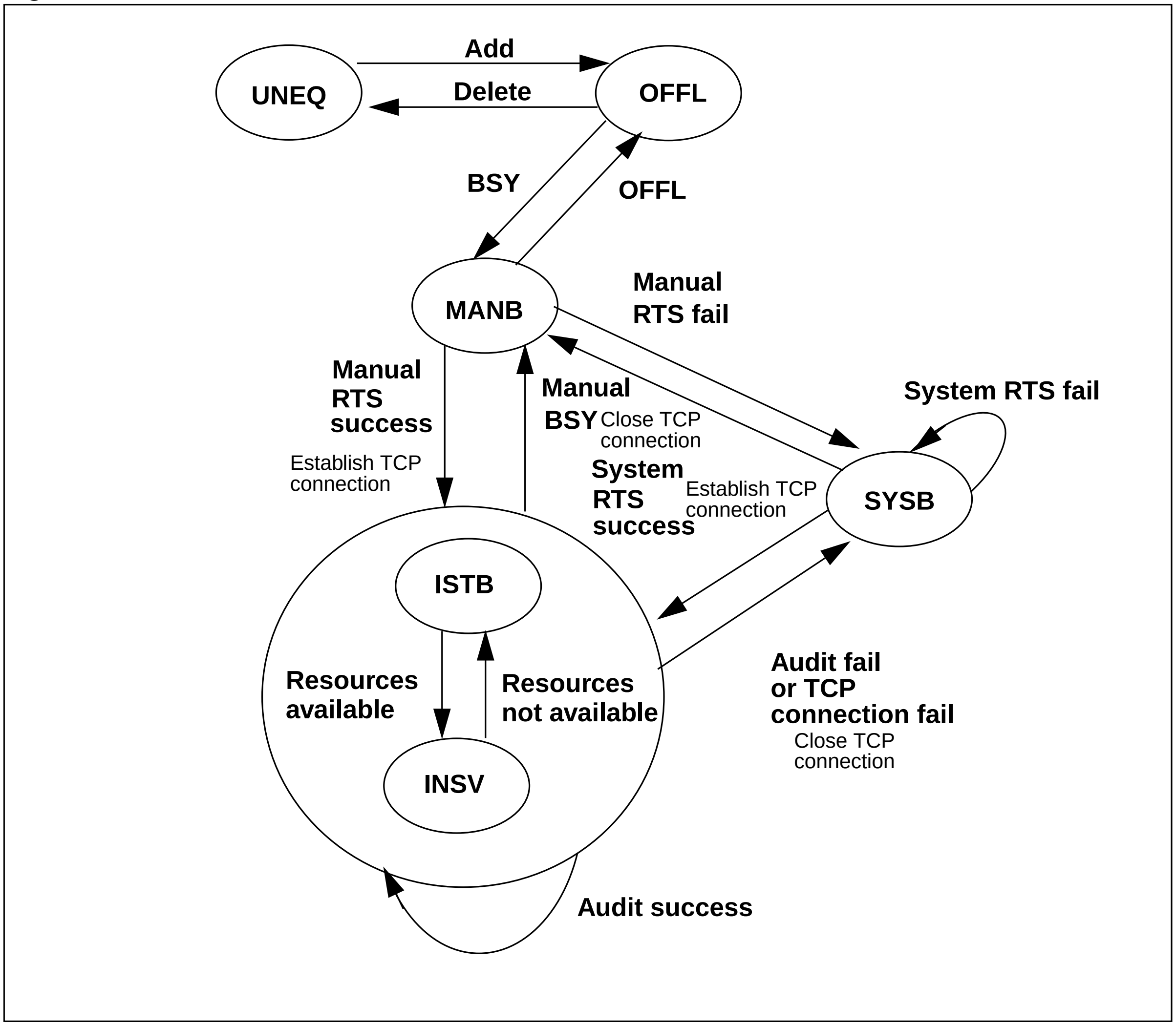
The maintenance for MIS nodes is similar to OSNM nodes, with the following two exceptions:

- Consideration is given to the state of the TCP connection between the switch and the MIS node.
- No session pools can be datafilled on the MIS node.

This section briefly describes the maintenance of MIS nodes. For complete details on the maintenance of OSSAIN nodes, refer to Chapter 10: “OSSAIN maintenance.”

The following figure shows the state transitions of an MIS node. Each transition type is described after the figure.

Figure 96 MIS node state transitions



UNEQ

Unequipped. The node has not been datafilled in table OANODINV.

OFFL

Offlined. The node has been datafilled, but has not been manually busied for the first time or has been offlined from the MANB state. No TCP connection has been established.

MANB

Manual busy. The node has been manually busied from the MAP. No TCP connection has been established, and any previously established connection for the MIS node is released by transition into the MANB state.

SYSB

System busy. The MIS node has been placed in a busy state by the system. No TCP connection is established while the node is in the SYSB state. This occurs under one of the following conditions:

- when the TCP connection fails
- when the TCP connection is closed by the node
- when the node goes out of service
- when an automated audit fails

When an MIS node goes SYSB, all service processing provided by that node is stopped and a major alarm is posted at the MAP.

INSV

In service. The TCP connection has been established for the MIS node to transition into the INSV state.

ISTB

In service trouble. The node is in this state if the connection is still up, but TCP resources are temporarily unavailable. A minor alarm is posted at the MAP. When resources become available, the MIS node transitions to the INSV state.

OSSAIN alternate routing

OSSAIN processing can perform alternate routing. Alternate routing allows an SN to *override* the translations route selected by the DMS switch for calling (A-party), called (B-party), or third (billing number) connections. This capability gives the SN more control in selecting the outgoing route for a call.

With alternate routing, the actual DN of the call is *unchanged*. The DN is still used in digit manipulation (based on the new route) before it is outpulsed.

Two methods of alternate routing

Before the switch connects to the party, the SN can use one of two methods to override switch translations and screening, as follows:

- The SN can send a *routing index* (RI) in an OAP message to the switch. The RI is used to directly override the route that was determined by DN (or LRN) translations and screening. The RI maps to table OAINRTE, which specifies an office route index into table OFRT. Using datafill in OFRT, the switch selects the new route for the given connection (calling, called, or third) and uses this route in any digit manipulation of the DN (or LRN) before outpulsing.
- The SN can send a *routing number* (RN) in an OAP message to the switch. The RN is a directory number that the switch translates to select a new route for the connection. This route is used in any digit manipulation of the DN before outpulsing.

Note 1: In this description of OSSAIN alternate routing, DN refers to the actual directory number associated with the A-party, B-party, or third party. RN refers to a number supplied by the SN to use for routing to the associated A-party, B-party, or third party.

Note 2: For complete details on OAP operations that contain the RI and RN, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Using a routing index—OAINRTE

The RI sent in an OAP message to the switch is an index into table OAINRTE (OSSAIN Route). This table provides a bridge between the SN and the switch routing tables—OFRT, OFR2, OFR3, and OFR4. Using datafill in OAINRTE, the operating company can determine the routes that correspond to RIs sent by the SN.

OAINRTE example

The following figure shows example datafill in table OAINRTE. In the example, four RIs are specified. Two RIs map to different indexes in table OFRT, and two RIs map to the same index in table OFR2.

Figure 97 MAP display example for table OAINRTE

```
RTEIDX       RTETABID        OFRTIDX
---------------------------------------------------------------------
0            OFRT            32
3            OFRT            1000
16           OFR2            50
235          OFR2            50
```

Note: For details on all the fields and valid values in table OAINRTE, refer to Chapter 7: “OSSAIN data schema.”

OFRT example

The following figure shows example datafill in table OFRT. In the example, office route 32 (from routing index 0) results in an outgoing trunk group, and office route 1000 (from routing index 3) results in a treatment. Both office routes are valid.

Figure 98 MAP display example for table OFRT

```
RTE      RTELIST
---------------------------------------------------------------------
5        (N D ISUP2WATC11 15 D088 N) $
32       (N D ISUP2WIT 0 N N) $
1000     (S D VCA) $
1005     (N D ISUP2WIT 0 N N) (N D MF2WIT 0 N N) $
```

Note: For details on all the fields and valid values in table OFRT, refer to the *Translations Guide*.

After determining the new route from datafill, the switch treats this route as it would any valid route obtained from translating the DN. For example, on B-party calls, the switch still applies standard and equal access (EA) translations and screening to the call, based on the actual called DN. This allows the switch to determine the called directory number type. The switch still performs digit manipulation on the DN, but it is based on the *new route* specified by the RI.

The RI remains with the connection for the life of the DN. If the associated DN is ever cleared or changed, the RI is also cleared. For details, see “Clearing and replacing RIs and RNs” on page 198.

If the SN sends an RI that is not datafilled in table OAINRTE, the switch-selected route is *not* overridden. The switch informs the SN of the error. The SN may or may not attempt to send a different RI (or RN) for the connection. There is no maximum number of times the SN can send an RI on a given call.

Using a routing number

The RN sent in an OAP message to the switch is translated and screened by the switch to determine a valid route. The DN (not the RN) is used to determine the called directory number type and local call area (LCA) and EA information. The DN is used to perform digit manipulation based on the new route.

The RN remains with the connection for the life of the DN. If the associated DN is ever cleared or changed, the RN is also cleared. For details, see “Clearing and replacing RIs and RNs.”

If the SN sends an RN that is incorrectly formatted, the switch does not store the RN. The switch informs the SN of the error. The SN may or may not attempt to send a different RN (or RI) for the connection.

Note: The switch can store an RN up to 18 digits long.

Clearing and replacing RIs and RNs

The SN can clear (remove) an RI associated with a connection by sending the switch a NIL (blank) value of either 2046 or #7FE. Likewise, the SN can clear an RN associated with a connection by sending the switch a value of zero digits in the RN count field. When the switch receives the zero value, it clears the RI or RN, but leaves all other DN-related information unchanged.

The SN can change the RI or RN for a given connection at any time, assuming there is a valid DN associated with the connection (see Note). Sending a new RI or RN replaces the existing one. If the SN does not send another RI or RN after clearing the existing one, then standard DMS switch translations on the DN (or LRN) is used to determine the route for the connection.

Note: The only time a DN may not be associated with the connection is during an OAP transfer to carrier request. In this case, the SN can send a B-party RI (not an RN) for a call being transferred to a carrier regardless of the presence of a B-party DN.

Interactions between RIs and RNs

RIs and RNs apply to *connections* not to calls. A call can have a maximum of three connections (calling, called, third). Each connection may or may not have an RI or RN applied to it. Also, the routing information associated with a call does not have to be of the same form. For example, a call can have an RN applied to the calling DN, an RI applied to the third DN, and no alternate routing applied to the called DN.

RIs and RNs are *mutually exclusive* on a given connection. Only one can apply at a time. The switch stores the most recent RI or RN received, as long as it is valid. For example, suppose the switch receives the following sequence of alternate routing values:

```
RI 0, RN 2012201234, RI 235, RI 3
```

The switch stores each value as it is received, replacing the existing value with the new one until the last value, RI 3, is used to route the connection.

If an RI or RN sent by the SN is invalid, the previously stored value (if any) is not replaced. For example, if an RI datafilled in table OAINRTE indexes a non-existent tuple in table OFRT, the previous RI or RN stored for the connection remains.

Note: The SN cannot send both an RI and an RN during the same operation. If it does, the switch does not process the operation, and informs the SN of the error. (One exception to this is the Session Initiation Request operation. If the SN sends multiple RN/RIs for a given connection, the last one processed is used.)

Additional QMS CT4Q refinements

Additional TOPS Queue Management System (QMS) call type for queuing (CT4Q) refinements can be applied to OSSAIN calls that route to an operator through table OAFUNDEF (OSSAIN Function Definition). By allowing CT4Q refinements, the existing QMS capability of segregating operator traffic is extended to OSSAIN calls.

Instead of having to datafill multiple functions to segregate OSSAIN operator traffic, operating companies can datafill *one* function to funnel the calls to an operator CT4Q that is refined based on the queuing criteria (such as class of service, language, time of day, and so on).

Note: QMS CT4Q refinements as described here also apply to the TOPSAUTO DAS function defined in table OAFUNDEF. This allows the option to utilize the same refinements to the various DA services to reach the desired DAS vendor service.

OSSAIN calls routed to operators—OAFUNDEF

In table OAFUNDEF, the FUNCTYPE subfield is set to TOPSOPER when the function provider is an operator. The QREFINMT and CQORDER subfields specify whether CT4Q refinements are applied and if so, which CT4Q order to use for the TOPSOPER function.

OAFUNDEF example

The following figure shows example datafill. In the example, the first tuple has the QREFINMT subfield set to Y and the CQORDER subfield set to POSTAUTO. So if a call is routed to the operator using the 0_MINUS_OPER function, CT4Q refinements are applied according to the POSTAUTO ordering with the 0_MINUS CT4Q as the starting call queue.

The second tuple has the QREFINMT subfield set to N, so if a call is routed to the operator using the DA_OPER function, the OPER_DA CT4Q is used for final call queue assignment (index in table TQMSFCQA).

Figure 99 MAP display example for table OAFUNDEF

```
FUNCID      FUNCNAME              FUNCAREA
---------------------------------------------------------------------------------
2           0_MINUS_OPER          TOPSOPER  0_MINUS Y POSTAUTO
4           DA_OPER               TOPSOPER  N OPER_DA N
```

Note 1: For details on all the fields and valid values in table OAFUNDEF, refer to Chapter 7: “OSSAIN data schema.”

Note 2: For complete information on TOPS QMS queuing and related datafill, refer to the *Translations Guide*.

OSSAIN calls that route to the operator as a result of one of the following methods can have additional CT4Q refinements applied using datafill in table OAFUNDEF:

- Trigger processing—These are floated calls that use OSSAIN trigger datafill to route to a TOPSOPER function.
- SN transfer to operator—These are calls that are transferred to the operator by the SN, using an OAP Transfer to Control List operation. The control list datafill (table OACTLDEF) points to a TOPSOPER function.
- Disposition routing processing—These are calls that failed to reach the SN. The disposition routing datafill (table OAFNDISP) points to a TOPSOPER function.
- Initial call presentation using OSSAIN—These are calls that route to an OSSAIN CT4Q that points to a TOPSOPER function, instead of routing directly to an operator using an operator CT4Q (during the initial call setup stage).

Auto voice link selection and connection

The DMS switch can select the voice link and automatically connect to it when establishing a new session with the SN. Also, the SN can request a voice connection *without* having to specify the logical voice channel number in the OAP message to the switch. The OAP request also allows the switch to automatically release any previous voice link connection before processing the request.

Note: For details on the other method for making OSSAIN voice link connections, refer to “Voice link connections between the switch and SN” in Chapter 2: “OSSAIN software functionality.” For complete information on OAP operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Using the auto voice link selection and connection capability saves OAP messaging between the SN and switch, which improves use of real time. This capability can be used in the following types of OSSAIN processing:

- during subscriber originations to the SN
- when a call triggers back to the SN
- when a passive SN is brought into the call
- when the SN initiates a session with the switch
- when an SN transfers a call to another SN
- when an operator transfers a call to an SN
- when a session is recalled to an SN
- during an OSSAIN preprocessing session with the SN

This section describes the two OSSAIN data tables that provide this optional capability:

- Table OAFUNDEF (OSSAIN Function Definition) specifies whether or not the switch should establish a voice link connection before beginning an SN session. It also specifies whether or not the switch should perform disposition routing when the voice link connection fails.
- Table SNVLGRP (SN Voice Link Group) provides the SN trunk group CLLI to use in the automatic voice link connection.

Note: Table OAVLMAP (OSSAIN Voice Link Mapping) does not require SN voice links to be datafilled when using auto voice link selection and connection. For details on OSSAIN voice link datafill, refer to Chapter 7: “OSSAIN data schema.”

Connecting the voice link—OAFUNDEF

Table OAFUNDEF defines the functions used in processing OSSAIN calls. The auto connect capability applies only to function types of SN that have the CAMHERE subfield set to Y.

The CONVOICE refinement specifies whether or not the switch should establish a voice connection before routing the call to the SN. When set to Y, the switch uses datafill in table SNVLGRP to select the voice link trunk and make the connection. When set to N, the switch routes the call to the SN without establishing a voice connection.

When CONVOICE is set to Y, the DISPROUT refinement specifies whether or not the switch should perform disposition routing if the auto voice connection attempt fails. When set to Y, the switch uses datafill in table OAFNDISP for origination failures. When set to N, the switch routes the call to the SN without a voice connection.

OAFUNDEF example

The following figure shows example datafill. In the example, the branding function has automatic voice connection and disposition routing enabled. The Yellow Pages function does not have automatic voice connection enabled.

Figure 100 MAP display example for table OAFUNDEF

```
FUNCID      FUNCNAME              FUNCAREA
------------------------------------------------------------------------------------------
1           BRANDING              SN  TASERV N N N Y CQ0 Y Y
2           YELLOW_PAGES          SN  DASERV N N N Y CQ1 N
```

Note 1: For details on all the fields and valid values in table OAFUNDEF, refer to Chapter 7: “OSSAIN data schema.”

Note 2: For details on disposition routing, refer to Chapter 2: “OSSAIN software functionality.”

Selecting the voice link—SNVLGRP

Table SNVLGRP specifies a voice link for a particular SN and function pair. This is the voice link selected by the switch when automatic SN voice connections are required (CONVOICE field set to Y in table OAFUNDEF) or when the SN makes an OAP request to the switch for a voice connection.

The table has a two-part key comprised of NODENAME and FUNCNAME. The CLLI field specifies the voice link used to make the connection for the SN and function pair.

The most idle selection sequence is used for selecting a voice link member from the group. After the voice link is successfully connected, the switch informs the SN of the trunk group and member number in an OAP message.

SNVLGRP example

The following figure shows example datafill. In the example, both SN_01 and SN_02 provide the calling card function. SN_01 also provides the AABS function. When routing a call to SN_01 for the calling card function, the voice link from voice link group VL_01 is used to serve the call.

When routing a call to SN_02 for the calling card function, the voice link from voice link group VL_02 is used to serve the call. Similarly, the voice link group VL_05 is used for the AABS function on SN_01.

Figure 101 MAP display example for table SNVLGRP

```
NODEFUNC              CLLI
---------------------------------------------------------------------
SN_01 CALLING_CARD    VL_01
SN_01 AABS_SVC        VL_05
SN_02 CALLING_CARD    VL_02
```

Note: For details on all the fields and valid values in table SNVLGRP, refer to Chapter 7: “OSSAIN data schema.”

Commercial credit card sales report

For calls that are billed to a commercial credit card, the commercial credit card sales report feature allows the TOPS switch to send charge information to a LIDB at the end of a call. This charge information is used by the supplier of the commercial credit card to adjust the credit limit available to the owner of the credit card.

With OSSAIN, an SN retrieves the charge information from the calling subscriber. Using OAP messaging, this information is sent to the TOPS switch, which then sends a calling card validation query to the LIDB. If validation is successful, the SN floats the call, allowing the switch to connect the calling and called parties. If validation is unsuccessful, the switch sends an error message to the SN.

Note: For more information on commercial credit card sales report, refer to the *Translations Guide*.

Authorization Codes

SNs capable of validating calling cards, commercial credit cards, and/or authcodes may send the approved authorization code to TOPS for placement in the AMA record. The SN can send the authorization code along with the supported type and request ‘no validation’. The switch will accept the value and store it appropriately for the AMA system.

Calling card authorization codes are recorded in AMA module code 611. Commercial credit card authorization codes are recorded in AMA module code 305. Authcode authorization codes are recorded in AMA module code 102.

Automatic carrier selection

Automatic carrier selection allows the TOPS switch to select the outgoing carrier for a call based on call characteristics. If the call is a country direct call type or uses an international calling card for billing an outbound call, then automatic carrier selection is performed.

For country direct calls, outgoing carrier selection is based on the terminating country, the originating country, and the incoming carrier. For calling card calls, the outgoing carrier selection is based on the terminating country and the calling card issuer ID.

Note: For more information on automatic carrier selection, refer to the *Translations Guide*.

Estimate of charges

A TOPS operator can enter hypothetical call information and provide the subscriber with an estimate of the charges for the call. Optional call information that affects charges includes date, time, and duration. For an estimate of a no-duration call, charges are provided for an initial period and a subsequent period. In a single call, an operator can perform multiple estimate requests, with each estimate billed at a fixed rate.

With OSSAIN, the SN retrieves details of the call from the subscriber. The SN then sends an estimate request operation to the switch. The switch processes the request and calculates the estimate of charges for the call. If processing is successful, the switch sends the charge information to the SN. If processing is unsuccessful, the switch sends an error message to the SN.

Note: For more information on estimate of charges, refer to the *Translations Guide*.

Global support

Global support of OSSAIN functionality is provided by the following enhancements:

- signaling interworkings with R1, R2, GOSS7 ANSI ISUP, and GOSS7 ETSI ISUP
- support of OAP operations
- support of the core OSSAIN functions:
  - datafill for defining queuing, functions, control lists, triggers, maintenance, and connectivity
  - trigger processing
  - transfers
  - voice links
  - data links
- support for current OSSAIN services in a global environment:

- — 0+ (operator assisted) alternate billing (collect, third, calling card, account)
- — 0- (operator handled) automation
- — 1-800 toll-free access
- — transfer to operator (operator fallback)

Called DNSCRN screening

When called DNSCRN screening is active, the called number in table DNSCRN is screened for the BLCKCALL, UNPAID, or CCR attribute, or any combination of these attributes. If screening determines that an attribute is datafilled against the given called number, the switch sets an indication and sends it to the SN. The SN may still complete the call even if an attribute is set against the called number.

Special location routing number (SLRN)

A special LRN (SLRN) can be associated with an incoming ISUP call when the call routes to the TOPS switch based on the LRN. The TOPS switch stores the SLRN for use in QMS routing, which may send the call to an SN for servicing. After the interaction with the SN, the call continues on to the terminating party.

The OSSAIN SLRN enhancement provides the following changes:

- a call origination (CO) type of SLRN
- optional answer on voice link connection to an SN
- additional QMS refinement tables for the SLRN:
  - — table TQSRNNAM defines the name of the SLRN criteria
  - — table TQSRNDIG associates the SLRN name with the SLRN digits
  - — table CT4QSLRN maps the actual QMS refinement to identify the new CT4Q
- additional OAP operations to support processing of the SLRN

Note 1: For more information on the QMS refinement tables, please refer to the *Customer Data Schema Reference Manual*.

Note 2: For details on OAP operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

Optional answer on voice link connection capability

Logic is added to the switch to determine whether to return an ISUP answer message (ANM) at the time a voice link is connected to an SN. Previously, the switch always returned an ANM at the time a voice link was connected.

Datafill in the ANSONVLC subfield in table OAFUNDEF provides this capability. If set to Y, the switch returns an ANM at the time of the connection. If set to N, the switch does not return an ANM; instead, the SN determines whether to send an ANM.

OSSAIN support for DA automation

OSSAIN supports the deployment of the database independent Directory Services (DA) automation services. Information pertinent to DA calls is provided to the SN for interpretation and call handling. Through OAP, this information is provided to the SN for initial call presentation.

For additional information, please refer to the *OSSAIN Open Automated Protocol Specification*, Q235-1.

Short Message Service

For wireless callers, a SN can send a short message to the switch to be forwarded to the Message Center for the caller. This functionality requires TOPSFTR parameters SHORT_MESSAGE_SERVICE and OSSAIN_RELEASE_20 be set. It also requires the SN send a valid wireless DN along with the text message. TOPS currently only supports wireless calling number in the North American dialling plan and thus the SN must forward the wireless calling number in the SMS request.

The SN will be informed if the request fails validation or forwarding to the Message Center; however, it will not receive a response indicator from the Message Center.

Part 3: Interactions

Part 3: Interactions includes the following chapter:

Chapter 5: “OSSAIN interactions,” beginning on page 209.

Chapter 5: OSSAIN interactions

This chapter discusses interactions and restrictions that apply to the OSSAIN functions described in Chapters 2, 3, and 4. The following table lists the functions in alphabetical order, and the page in this chapter where each description begins.

Table 13 OSSAIN interactions and restrictions

Feature	Page number
Additional QMS CT4Q refinements	page 210
Auto voice link selection and connection	page 211
Branding	page 212
Call timing	page 212
Changing the base service	page 213
Country direct	page 213
Equal access	page 214
Global support	page 215
OSSAIN alternate routing	page 215
OSSAIN conferencing	page 217
OSSAIN preprocessing	page 218
OSSAIN QMS MIS	page 219
OSSAIN support for DA automation	page 220
OSSAIN support for toll-free calls	page 221
Outgoing ISUP	page 224
Release link trunking	page 224
Routing and queuing	page 225
Simultaneous connections	page 226
Special Location Routing Number	page 226

Table 13 OSSAIN interactions and restrictions

Feature	Page number
Transitions	page 226
Triggers	page 229
Voice link broadcasting	page 232

Note: For a list of OSSAIN system limitations, refer to Chapter 6: “OSSAIN engineering.” That chapter also includes all the limitations and restrictions related to IP addressing, routing, capacity, and security.Chapter 6:

Additional QMS CT4Q refinements

This section discusses interactions and restrictions that apply to OSSAIN QMS CT4Q refinements for calls that route to the operator or the DAS autosys.

CT4QAUTO refinement

The CT4QAUTO CT4Q refinement table refines the CT4Q by the automated service a call has received. This table allows all the automated services TOPS provides (ACTS, ACCS, MCCS, AABS, ADACC, and ADAS) to be used as a criterion for queuing. It also allows a NO_AUTO criterion to be used for calls that have not received any automated services.

When applying the CT4QAUTO refinement for an OSSAIN call, the automated service criterion is set to NO_AUTO, *regardless* of whether the call has received any TOPS automated services prior to OSSAIN.

ASST refinement

The ASST CT4Q refinement ordering is a function of TOPS CASE (Customer Assistance Service Enhancements), which is provided by SOC option OSB00001. To receive CT4Q refinements based on ASST ordering, OSB00001 must be turned ON.

If the call attempts to use ASST refinement ordering and OSB00001 is not turned on, an OAIN302 log is produced to indicate a SOC problem. The call is routed using the CT4Q from table OAFUNDEF as the final operator CT4Q for the call.

Invalid CT4Q refinement

When applying CT4Q refinements to a TOPSOPER/TOPSAUTO CT4Q for an OSSAIN call, if the result of the refinements leads to an OSSAIN CT4Q instead of to an operator CT4Q, an OAIN206 log is produced. This log indicates invalid CT4Q refinement, and the call is taken down. This action avoids any potential recursive refinement conditions due to incorrect QMS datafill.

Applying CT4Q refinements

For calls that route through an OSSAIN CT4Q to the operator for initial call presentation, the switch applies QMS CT4Q refinements one time to reach the OSSAIN CT4Q during the call setup stage. If this OSSAIN CT4Q routes to a TOPSOPER function that has the QREFINMT set to Y, the switch applies QMS CT4Q refinements a *second* time.

Auto voice link selection and connection

This section discusses interactions and restrictions that apply to the auto voice link selection and connection capability.

OSAC

Auto voice link selection and connection is supported in both the standalone and centralized OSSAIN (OSAC) configurations. In the OSAC configuration, the OSAC host switch is responsible for making the voice link connection for the centralized SN.

Datafill in tables OAFUNDEF and SNVLGRP at the OSAC *host* switch is used to provide auto voice link selection and connection. If the voice connection attempt fails and the DISPROUT field (table OAFUNDEF) is set to Y, then datafill in table OAFNDISP at the OSAC *remote* switch is used to route the call.

Note: Refer to Chapter 3: “OSAC call processing” for more information on OSAC.

Simultaneous connections

Auto voice link selection and connection is supported in an OSSAIN simultaneous connection scenario. When a passive SN is brought into the call, the switch can automatically connect a voice link for the passive SN, assuming there is no broadcast voice link already connected to the call.

If the voice link connection attempt fails, the DISPROUT field (table OAFUNDEF) is checked, as follows:

- If DISPROUT is set to N, the passive SN is brought into the call.
- If DISPROUT is set to Y, the passive SN is *not* brought into the call. In this case, OSSAIN disposition routing is not performed.

OSSAIN preprocessing

Auto voice link selection and connection is supported for calls that receive preprocessing. If the voice link connection attempt fails, the DISPROUT field is checked, as follows:

- If DISPROUT is set to N, the call is routed to the SN.
- If DISPROUT is set to Y, the call continues to TOPS processing. In this case, OSSAIN disposition routing is not performed.

SN initiated calls and recalls

Auto voice link selection and connection is supported for SN initiated calls and for calls that recall to the SN. OSSAIN disposition routing does *not* apply for these calls. If the voice link connection attempt fails, the DISPROUT field is not checked and the call remains at the SN.

Trigger events

Auto voice link selection and connection is supported for an OSSAIN call that triggers to the SN. However, it is not supported for a trigger event for which the SN does not get control of the call, but is only informed of the trigger event (that is, when the SNCONTRL subfield is set to N in the trigger profile).

Broadcast voice links

Auto voice link selection and connection is *not* supported for broadcast voice links.

Branding

The following interactions apply to branding:

- The OAP allows the SN and the switch to exchange information about the current branding status of a call.
- The TOPS_BRAND_DISPLAY parameter in table OFCENG determines how branding information is displayed at the operator position for both OSSAIN and TOPS calls.

Call timing

The following restrictions apply to call timing:

- When a GEN AMA is performed as part of switch trigger datafill and processing (for example, in a sequence call), then, in effect, a new call is started and presented to the SN with no applicable call timing information.

  Therefore, if the SN requires call timing information for a call just prior to a trigger event, the GENAMA option should *not* be datafilled against that trigger event. Refer to Chapter 7: “OSSAIN data schema,” for information on the GENAMA option in tables OATLKPRF and OADTFPRF.
- Conversation time on the DMS switch must pass a minimum charge duration threshold before billable conversation time is accumulated. Calls that disconnect prior to the minimum charge duration have a conversation time of zero. This duration is specified in the MINIMUM_CHARGE_DURATION parameter in table OFCENG.
- When AMA is generated on a call with the forward party connected, the charge status of the call for the next period is always set to “billable.” This is the case even if the charge status at the time of the request is “not billable.”

- Multiple GEN AMA requests when the forward party is connected may affect the correlation of AMA records for RLT calls. The existence of multiple AMA records for the same call may make it difficult to match AMA records between the TOPS switch and the previous switch involved in RLT.

Changing the base service

An SN can change the base service of a call from TASERV (toll and assist) to DASERV (directory assistance), or from DASERV to TASERV. After processing the service change, the DMS switch sends updated call details to the SN.

The following restrictions apply to changing the base service:

- A service change from TASERV to DASERV is not allowed on a call if a forward party is currently connected. However, the service change is allowed if a forward *number* exists but is *not* currently connected. After the change, the forward number is cleared (replaced with the NIL value).
- An AMA record is not automatically generated as a result of a service change. If an AMA record is needed, the SN must specifically request it using OAP AMA operations. To provide billing for the old service, the SN should request AMA *before* requesting to change the base service.
- The class charge of a call is reset to unspecified (no class charge) after processing the request to change the base service. The call must be class charged again in the new service to satisfy billing.
- For TASERV to DASERV, the call origination type remains what it was prior to the service change. For DASERV to TASERV, the call origination type is reset to Zero Minus.
- For DASERV to TASERV, the charge status of the call is reset to billable.
- Changing the base service to INTCSERV is not allowed while the call is at the SN.

Country direct

The following interactions apply to OSSAIN country direct:

- The SN is allowed to hand off a country direct call to a function datafilled for AABS instead of completing the call. Since AABS is no longer supported, however, such a call will route to MCCS/ACCS if eligible, and otherwise to a TOPS operator.
- Using trigger mechanisms, sequence calls are supported.
- A TOPS operator can transfer the country direct call to an SN.
- OSSAIN country direct is supported in a centralized OSSAIN (OSAC) environment.

- The OM group CTRYDIR contains a register to count the number of country direct calls that are handed off to an SN. (Refer to Chapter 12: “OSSAIN operational measurements,” for an example.)
- OSSAIN country direct is enabled with SOC option OSAN0102. If OSAN0102 and SOC option ENSV0101 (TOPS Country Direct) are *both* ON and a country direct call is routed to an SN whose OAP version is less than 5.0, the call is presented to the SN with a CO type of UNSPEC instead of CDIR.

The following restrictions apply to OSSAIN country direct:

- Calls are alternately billed only. Station paid and person paid are not allowed.
- The SN cannot change the service to DA.
- It is recommended that the SN *not* change the calling number, release the calling party, or connect back to the calling party for country direct calls.
- OSSAIN country direct is supported for TOPS trunks that support OSSAIN. It is also supported for ISUP IT and ISUP ATC trunks if SOC option OSEA0102 (ISUP to TOPS Enhancements) is ON. (Refer to Chapter 8: “OSSAIN software optionality control,” for details on the SOC option for OSSAIN country direct.)

Equal access

An OSSAIN SN has the same equal access functionality as that of an operator. So when a call is at an SN, OSSAIN processing has the following capabilities:

- determining whether a query is needed for billing validation
- determining whether queries are blocked for a particular carrier
- determining whether the class charge is valid for the station class of the calling party
- marking a call to be transferred to an interLATA carrier

Note: A call marked for transfer is routed to the carrier switch and the call is denied OSSAIN triggering.

This section discusses interactions and restrictions that apply to equal access calls.

COMFGD and EAOSS signaling types

The following limitation applies to calls received on a TOPS trunk with either the COMFGD or EAOSS signaling type (table TRKGRP). For these calls, the POS subfield in table STDPRTCT (subtable STDPRT) must be set to TOPS so that translations can mark a DD-signaled call as OA. This value routes the call to table QMSTOPS for QMS refinement, and later selection of OSSAIN. Otherwise, the call tandems through the TOPS office without checking for OSSAIN eligibility. For additional information on signaling types, please refer to the *Translations Guide*.

STDPRTCT.STDPRT example

The following figure shows example datafill in subtable STDPRTCT.STDPRT. In the example, the DD-signaled call received with digits 2122205555 is marked as OA and is eligible for QMS refinement.

Figure 102 MAP display example for subtable STDPRTCT. STDPRT

```
FROMDIGS      TODIGS        PRETRTE
----------------------------------------------------------------------
2122205555    2122205555    T  OA 0 TOPS DD 10 10 TOPS
```

Note: Refer to the *Translations Guide*, for details on table STDPRTCT.

Global support

The following restrictions apply to global support of OSSAIN:

- SOC code OSAN0101 provides global support. However, some OSSAIN functions are controlled by higher OSAN SOC codes that must also be enabled for the function to work in the global environment. For details, refer to Chapter 8: “OSSAIN software optionality control.”
- global support does not include the following functions:
  - — attendant pay station
  - — service analysis center (SNAC) logs
  - — global competitive access
  - — TOPS intelligent network (IN) context
  - — logical terminal identifier (TID) for time and charges recall
  - — music and announcement in queue

OSSAIN alternate routing

This section discusses interactions and restrictions that apply to OSSAIN alternate routing.

Equal access

On EA calls, standard translations first are performed on the DN and then on the RN. The *called* DN is used for local access and transport area (LATA) screening, which determines if the call is interLATA or intraLATA. *Calling* party information is used to determine the carrier (unless the SN overrides the carrier with another). If an RN is present on a call, it is used for EA screening, which determines the EA route. The RN does not affect the EA status or the carrier assignment of the call.

If an RI is present on the B-party connection, the route specified by the RI is taken regardless of the EA status or carrier assignment of the call; however, the EA status and carrier assignment remain unchanged.

Note: At the point of routing a call, the SN knows whether the call is a carrier call, and if so, its carrier assignment. If the SN does not want to use the RN or RI, it can clear the RN or RI. This causes translations to be performed again on the DN to determine the route.

Local number portability (LNP)

With LNP enabled, a calling, called, or third DN may be ported. LNP queries are done on *all* numbers regardless of any alternate routing status, if the conditions for LNP queries are met.

If a called number is ported, translations first are performed on the DN. The called number type remains from the initial translations on the DN. If an RI is associated with the connection, the route specified by it *replaces* the route obtained from the LRN. If the ported call is routed to treatment, standard logs on the DN or LRN (or both) are generated. If the call is routed to treatment by an RN or RI, an additional log (TOPS132) that lists the RN or RI is generated.

Note: If an RN or RI is on the call, a TOPS132 log is generated regardless of how the call routed to treatment.

On B-party connections, the SN knows the LNP status (and the LRN, if present) of the call before it releases the call. (On A-party and third connections, the SN must specifically request the LNP information.) If the SN wants to remove or change the alternate routing based on the LNP status of the call, the SN can send another RN or RI. This causes the switch to perform a complete set of translations (DN, LRN, RN) again, using the new alternate routing information.

TOPS translations group (XLAGRP) scheme

Alternate routing does not affect the TOPS translations group (XLAGRP) translations scheme. The XLAGRP translations scheme allows RNs to be processed, and it allows RIs to override the translations route obtained from table TOPSDP (TOPS Dial Plan).

Note: For details on the XLAGRP translations scheme, refer to *TOPS Translations and Screening User's Guide*, 297-8403-905.

The following restrictions apply to OSSAIN alternate routing:

- Alternate routing works only in North American (NA) markets; it does not affect translations in non-NA environments. Universal translations and open numbering *do not* support alternate routing.
- Alternate routing information *does not* remain with the call when it transfers to a TOPS automated system (such as MCCS/ACCS). If alternate routing information is needed for a service, the SN should not transfer the call to an automated system.
- Alternate routing information *does not* remain with the call when it transfers to the operator. If alternate routing information is needed for a service, the operator should transfer the call back to the SN.
- LCA screening is performed on a called DN to determine if the call is a local call. It is performed regardless of any alternate routing associated with the call. If LCA screening determines the call is local, then toll charges may not apply.
- Alternate routing does not affect rating. If an RN that is applied to a called connection has a different rate center than the DN, the rate obtained from the DN is used.
- OSSAIN alternate routing is only a portion of the functionality provided by SOC option OSAN0102. Disabling alternate routing also disables the other features. Refer to Chapter 8: “OSSAIN software optionality control” for details on OSAN0102.
- Table OAINRTE supports only 2046 tuples. So, the SN can send a maximum of 2046 alternate RIs to the switch.
- Table OAINRTE supports only OFRx tables (OFRT, OFR2, OFR3, and OFR4). OSSAIN alternate routing does not support IBN tables and IBN trunks.
- If users change a tuple in table OFRx, the switch *does not* ensure that references in table OAINRTE (RIs) have been deleted or changed first. When an SN sends an invalid RI, the switch responds with an error message to the SN. The error message indicates the invalid routing index.

OSSAIN conferencing

The following restrictions apply to OSSAIN conferencing:

- Only one conference is allowed per session.
- Conference size is limited to four subscribers.
- A call may not be floated without a B-party.
- Release link trunking cannot be used on conference calls.
- A simultaneous connection cannot be initiated, nor can one be in progress when the conference is created or when a party is added to an existing conference.

- A call participating in a conference cannot be transferred to an operator position.
- A call participating in a conference cannot be transferred to a carrier.
- A call participating in a conference cannot be routed to treatment through an explicit operation from the SN. However, a DN may still route to treatment.
- Calls in different DMS switches cannot be conferenced together. For example, a call in an OSAC remote switch cannot be conferenced with a call in an OSAC host switch.

OSSAIN preprocessing

The following restrictions apply to OSSAIN preprocessing:

- OSSAIN preprocessing is allowed only at the PREOPR and POSTAUTO stages in QMS refinement. PREOPR refining is performed before a call receives any type of operator processing (TOPS or OSSAIN). POSTAUTO refining is performed if a call leaves a TOPS automated system and requires further queuing. (However, a call refined after completing the automated portion of ADAS or ADASPLUS is not eligible for OSSAIN preprocessing.)
- Calls must route only to functions provided by an SN. If the translation results in a function type (field FUNCTYPE in table OAFUNDEF) of either TOPSOPER or TOPSAUTO, the call is not allowed to enter OSSAIN preprocessing. An OAIN305 log is generated and the call continues to an operator or automated system.
- Calls are limited to one OSSAIN preprocessing session. The only exception is an Automated Directory Assistance Call Completion (ADACC) call. ADACC calls are allowed one interaction during each of the DA and TA portions of the call.
- Calls that receive SN branding are no longer eligible for switch branding, except in the case of ADACC. ADACC calls are allowed one branding function for each portion of the call, whether provided by the switch or the SN.
- Calls that receive switch branding at the PREOPR refinement stage are still eligible for an SN preprocessing session at the POSTAUTO refinement stage. The SN is notified that the call has been branded and it is up to the SN to ensure that it is not branded again, unless necessary.
- If no branding function is provided by the SN during an OSSAIN preprocessing session, switch branding cannot be performed when the call resumes. However, if the call reaches another queue refinement stage, such as POSTAUTO, it is eligible for switch branding.
- Preprocessing is not allowed at the RECALL or ASST queue refinement stages.

- Switch branding is not allowed for calls using standard OSSAIN processing.
- An SN must inform the switch if it has provided branding. The switch needs this information to properly display the branding status of a call at an operator position. This restriction applies to both standard and preprocessed OSSAIN calls.

OSSAIN QMS MIS

As of SN07, OSSAIN QMS MIS capability is no longer being updated to include new OAP or OSSAIN functionality. OSSAIN QMS MIS is slated for end of life in the near future.

- Preprocessing is not supported for POSTAUTO-refined ADAS or ADASPLUS calls. This limitation applies only to calls that are successfully handled by the ADAS or ADASPLUS system. These calls are still connected to voice recording and processing equipment, waiting to play back the subscriber’s requested city and listing to an operator. A call in this condition is not interrupted with an OSSAIN preprocessing session.

  However, preprocessing is allowed for calls that are considered unsuccessful at recording the city and listing and which need an operator. Also, calls are eligible to receive preprocessing prior to connecting to the ADAS or ADASPLUS systems.
- Disposition routing using table OAFNDISP does not take place for OSSAIN preprocessing. This applies to all dispositions (deflections, overflow, function blocking, and origination failures). Calls that are unable to obtain a session immediately, or queue for a session, bypass preprocessing and continue to a TOPS operator or automated system.
- Authcode calls are eligible for OSSAIN preprocessing. MCCS is the operator type for authcode calls. In order for these calls to receive preprocessing, the CT4Q must have MCCS in the OPRSYS field of table CT4QNAMS.
- Simultaneous connections are not allowed during an OSSAIN preprocessing session.
- Preprocessing is not allowed on intercept or delay call origination types.
- If the SN attempts to resume an OSSAIN preprocessing call with no parties attached, the call is terminated by the switch.
- OSSAIN preprocessing is not supported on RONI trunk group types. Valid trunk group types are TOPS, IT (intertoll) and ATC (access tandem to carrier).

OSSAIN support for DA automation

OSSAIN supports the deployment of the database independent Directory Services (DA) automation services. Information pertinent to DA calls is provided to the SN for interpretation and call handling. Through OAP, this information is provided to the SN for initial call presentation.

For additional information, please refer to the *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following restrictions apply to OSSAIN support for DA automation:

- Existing OSSAIN restrictions on SentPd Coin/Hotel, Inward, CAMA, RONI, and INTC are still in effect.
- External rating is not supported.
- Only Nortel DMS/DAS protocol is supported.

The following section describes the interactions with OSSAIN support for DA automation:

Initial call routing

Standard OSSAIN routing and queuing is used to route the call to the SN. For SNs that are providing ADAS/ADAS Plus equivalent services, there are the following interactions:

- Sentpd coin and hotel calls are not supported
- Inward calls are not supported
- Billing is not verified
- ADAS/ADAS Plus related datafill/parms are ignored

When the passive operator is brought into the call. if the billing is not satisfied, the call would arrive at the operator's DA billing screen instead of the DA search screen.

Call arrival tone and caller join tone

No call arrival tone is provided when the passive operator is brought into the call. No caller join tone is provided when the caller's speech path is updated by the SN. If desired, the SN is responsible for providing these tones.

Call context block

The call context block is a generic block of data that contains additional information about the call. OSSAIN supports the passing of this context block between the SN, OPP-compatible positions, and the DAS. The context block can be used to coordinate call processing between the different systems. For example. the SN can use the context block to inform the DAS what top-line information should be displayed at the position.

Note 1: Standard Nortel DMS/DAS protocol 4 or higher is required to pass the context block information to the DAS.

Note 2: Additional DAS and/or position development may be required to populate the position with the information from the context block.

Text to operator text

The OAP Text to Operator Request allows a small block of text to be passed to the operator position once the operator becomes involved with the call. How this block of text is displayed at the operator screen is dependent on the position. Additional position development may be required to support the DA screens.

BILTYPES field in table ADACCOPT

When SOC OSAN0102 is ON for OSSAIN calls, an entry of NONE in the BILTYPES field in table ADACCOPT indicates that NO billing types are valid except auto-collect.

OSSAIN support for toll-free calls

TOPS and OSSAIN support calls that have already received toll-free processing. Either the U.S. variant (Enhanced 800) or the Canadian variant (800 Plus) of toll-free processing can be applied to a call before it arrives at an SN. This section describes how OSSAIN interacts with each variant.

Note: For details on DMS switch toll-free processing that can be applied to a call before it arrives at an SN, refer to *DMS-100 E800 SSP Toll-free Numbers Service Guide*, 297-5151-021 or *DMS-100 800Plus and End-Office Display Service Guide*, 297-5151-022.

Enhanced 800 features

Before a call arrives at a TOPS switch, the following Enhanced 800 features may have been applied. This section describes how OSSAIN interacts with a call that has already received Enhanced 800 processing.

Ten-digit routing

The SSP that initiated the toll-free query may strip off the NPA of the routing number returned by the SCP (service control point), depending on datafill in table NSCDEFS. The TOPS switch supports the ability to receive either seven- or ten-digit routing numbers. OSSAIN can complete calls to either seven- or ten-digit destinations. Also, an OSSAIN SN can *overwrite* this routing number with a new number if required by the service.

Comfort tones

Comfort tones do not apply to toll-free calls routed to OSSAIN, because the destination in this case is not an MF trunk. OSSAIN does not support comfort tones *after* the SN processes the call, regardless of whether the outgoing trunk is MF.

Call forwarding

Call forwarding may occur in a toll-free call before the call arrives at a TOPS switch. After the call has been presented to an SN for processing, toll-free call forwarding does not apply.

Four-digit CIC codes

A four-digit CIC (carrier identification code) may be returned by the SCP and used to route the call to a TOPS switch. In this case, the CIC is not sent to the SN. However, the SN can choose to set a three- or four-digit CIC code for completing the call if required by the service.

Automatic call gapping (ACG)

ACG processing occurs at the SSP that initiated the toll-free query. OSSAIN can process a call that has received ACG processing.

International 800

If the SCP returns an international number, it can be used to route the call to a TOPS switch. OSSAIN can complete the call to the international number after the call receives processing from an SN.

Northbound

The Northbound ISUP (ISDN user part) parameter conversion for toll-free calls terminating in Canada occurs at the SSP before the call is routed to a TOPS switch. OSSAIN can process a call that has received Northbound processing.

Termination notification

OSSAIN does not support the ability to notify the SCP when a toll-free call is ended.

800 Plus features

Before a call arrives at a TOPS switch, the following 800 Plus features may have been applied. This section describes how OSSAIN interacts with a call that has already received 800 Plus processing.

Call forwarding

Call forwarding may occur in a toll-free call before the call arrives at a TOPS switch. After the call has been presented to an SN for processing, toll-free call forwarding does not apply.

ACG

ACG processing occurs at the SSP that initiated the toll-free query. OSSAIN can process a call that has received ACG processing.

International 800

If the SCP returns an international number, it can be used to route the call to a TOPS switch. OSSAIN can complete the call to the international number after the call receives processing from an SN.

Southbound

The Southbound ISUP parameter conversion for toll-free calls terminating in the U.S. occurs at the SSP before the call is routed to a TOPS switch. OSSAIN can process a call that has received Southbound processing.

Overflow call routing (OCR)

OCR processing occurs at the SSP that initiated the toll-free query. OSSAIN can process a call that has received OCR processing.

Call prompter

OSSAIN does not support the call prompter feature. Toll-free calls that receive OSSAIN processing are not eligible for call prompter.

Courtesy response

OSSAIN does not support the courtesy response feature. Toll-free calls that receive OSSAIN processing are not eligible for courtesy response.

End office display

OSSAIN does not support end office display. Toll-free calls that receive OSSAIN processing are not eligible for end office display.

800 AMA

Without TOPS or OSSAIN interaction, an SCP response is received for an 800 query and the SCP data is saved for later inclusion in an 800 AMA record. This record contains the dialed number, the routing number returned from the SCP, and the calling number. If any 800 features are used by the caller, feature activation AMA records are generated for the call.

TOPS and OSSAIN do not support the generation of any 800-specific AMA records. Information used to generate toll-free AMA records (described in the previous paragraph) is discarded when the toll-free call arrives at the TOPS switch. However, standard TOPS AMA billing records are generated for the call. The AMA records generated for 800-to-TOPS calls contain both the routing number and calling number, among other data.

If the dialed number is required in the AMA record, SNs can produce custom AMA records at the TOPS switch for the call. Custom AMA records are essentially blank AMA modules whose format can vary depending on the service. The content of the custom AMA modules must be coordinated between the SN and the downstream AMA processing. TOPS software does not read or modify these modules in any way; they are simply appended to the AMA record for the call and sent to downstream processing.

Note 1: For details on custom AMA, refer to Chapter 9: “OSSAIN billing.”

Note 2: For more information on AMA records generated for TOPS calls, please refer to *Bellcore Format Automatic Message Accounting Reference Guide*, 297-1001-830.

Outgoing ISUP

An SN can change the DN in the Calling Party Number (CgPN) parameter in an outgoing ISUP initial address message (IAM). The SN can use this capability to coordinate network level and node level processing.

The following restrictions apply to the DN supplied by the SN to use in the CgPN parameter:

- The changed CgPN is used in the ISUP IAM only when the subsequent forward connection is made by the SN using OAP version 4 or above.
- The changed CgPN is no longer in effect if the SN transfers the call to an operator or TOPS automated system instead of making a forward connection.
- The changed CgPN is not recorded on the AMA record or used in any other processing of the call. The SN can use the OSSAIN custom AMA capability to record this number on the AMA record.

Release link trunking

This section discusses interactions and restrictions that apply to RLT.

OSSAIN call processing

An SN application that uses RLT saves trunk facilities for the operating company. For each OSSAIN SN application, the developer must decide whether, and for which calls, to activate RLT. An understanding of RLT is important to the OSSAIN SN application developer, because calls are released back to the previous switch with RLT. This functionality has an impact on post float processing.

SN developers should not make assumptions about a call’s data when using RLT and triggers. This is because the re-originated call from the previous switch *appears to be a new call* to the TOPS switch. For example, a call that has RLT performed and triggers back to an SN appears different (such as a changed call ID), compared to a call that triggers back, but which has not had RLT performed. The difference is in the data the TOPS switch sends to the SN.

If using RLT, the SN must request it *explicitly* during the call float process, The SN makes the request using an OAP RLT data block, which prompts the TOPS switch to perform the RLT eligibility checks.

Note 1: SNs are the only entity that must explicitly request RLT; in other environments (such as MCCS and TOPS operators) RLT is initiated implicitly.

Note 2: For complete information on OAP operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

OSSAIN trigger support

In RLT phase I, the TOPS switch does *not* perform trigger analysis. If the SN sends an RLT data block, TOPS performs RLT but any trigger functionality for the call is lost.

In RLT phase II, the TOPS switch analyzes the trigger profile if provided, and determines if the DMS-250 switch can support the profile. If the DMS-250 switch can support the profile, RLT is performed. If it cannot support the profile, the call remains at the TOPS switch.

Note: If a trigger profile is not provided, analysis will not occur. In this case, calls can RLT, but will not sequence.

The DMS-250 switch supports the following two trigger events (only one event per call):

- An octothorpe (#) trigger event is supported, if the release forward indicator is set in switch datafill. If it is not set, TOPS will not RLT the call. The octothorpe trigger is supported in all three call phases: connecting, talking and disconnect.
- A called disconnect trigger is supported in the talking phase.

When connecting triggers are present, the TOPS switch completes the call, and attempts RLT after answer is received.

Note: While RLT phase II supports a small set of triggers, the SN developer must use care in designing an application for these calls. When the DMS-250 returns the call to the TOPS switch, it is a re-originated call. A re-originated call contains the same data as the original call just prior RLT, except for data that is not in the UCP context block, such as the call ID.

Three-way calling

In the OSSAIN environment, RLT does not support three-way calling. There is no way for the DMS-250 switch to RLT and bring both parties back to the SN.

Routing and queuing

OSSAIN routing and queuing uses the TOPS QMS CAM. A QMS application is created for OSSAIN with its own set of call and agent queues. OSSAIN does not adversely affect QMS functionality.

The following restrictions apply to datafill for routing and queuing:

- The maximum number of functions that can be datafilled in table OAFUNDEF is 1023.
- The maximum number of session pools that can be datafilled in table OASESNPL is 4095, if 768 nodes are provisioned on the switch. However, a maximum of 256 session pools can be provisioned on a single node.

Simultaneous connections

The following restrictions apply to a simultaneous connection in OSSAIN:

- The operator cannot initiate a simultaneous connection or become an active agent.
- At most, two agents can be connected to a single call at the same time. So if the operator is involved in a simultaneous connection, the operator cannot bring in another agent by requesting assistance from SA, IC, or CSE operators. However, if assistance is needed, the operator can first release the SN from the call.
- The operator does not receive screen updates, except for calling and called hook status changes.
- Because the operator has fewer operations available during a simultaneous connection, the operator cannot split the calling and called voice path.
- A TOPS automated system cannot be connected to a call in a simultaneous interaction.
- AMA billing appends a module 184 for each SN attached in a simultaneous connection. Each module code reflects the elapsed time involved for that SN.

Special location routing number

The following restrictions apply to SLRN:

- Only an incoming ISUP call can signal an LRN, so only an incoming ISUP call can have an SLRN associated with it.
- The ANSONVLC (answer on voice link connection) subfield in table OAFUNDEF controls the sending of an ISUP answer message, so this subfield applies only to calls that arrive at the TOPS switch on an ISUP trunk.
- Datafill or ISUP connectivity must prevent an LNP query on the called number after the TOPS switch handles the SLRN call. A later LNP query would route the call back to the TOPS switch and prevent the call from reaching the terminating party.
- If an SLRN service subscriber has also undergone service provider portability, the LRN that routes to the subscriber must be stored local to the SN. Because the SN supplies the LRN, routing to the terminating end office must support ISUP signaling.

Note: For details on TOPS LNP and outgoing signaling, please refer to *TOPS LNP User's Guide*, 297-8403-902. For additional information on signaling types, please refer to the *Translations Guide*.

Transitions

This section discusses interactions and restrictions that apply to transition types in OSSAIN.

SN transfer to DA operator

For the following two conditions, it is recommended that the SN complete the billing for the call before it transfers the call to the DA operator:

- if the call has recalled to the SN by a trigger event
- if the called party is connected to the call

The SN can complete the billing for a call by requesting a GEN AMA. Otherwise, if the SN performs the transfer to a DA operator without completing the billing, the DMS takes the following actions:

- releases the called party
- generates an AMA record

An SN can transfer a call to the DAS without attaching an operator via the TOPSAUTO autosys, DAS, in table OAFUNDEF. This function can be used for ADAS+ service; however it is not dependent on it.

SN transfer to operator position

The following restriction applies to SN transfers to any supported operator position: If the SN updates the caller ID status to “blocked” prior to the transfer, the operator screen display does not indicate a status of “blocked.” Only if the operator blocks the call does the screen indicate “blocked.”

SN transfer to AABS

The following conditions must be met for the switch to accept an SN request to transfer a call to AABS:

- The incoming facility is configured to allow AABS.
- The calling number is present.
- The called number is present.
- The billed number passes validation checks.
- Equal access checks must pass.
- The room number is obtained for hotels that require room number.
- The billed number is not an overseas number.
- The call must not be marked for notify.

Note: The AABS protocol is no longer supported. An SN can still transfer a call to a function datafilled to go to AABS, and such a call must still meet the AABS criteria listed above. Then, if the call is eligible for MCCS/ACCS, it transfers to that automated system instead of to AABS. Otherwise it transfers to a TOPS operator. Please refer to *Translation Guide* for more information about TOPS MCCS/ACCS.

Operator transfer to control list

The following conditions must be met for an operator to transfer a call to an OSSAIN control list:

- The automatic call distribution type is QMS.
- The call origination (CO) type (in table QMSTOPS) is one of the following supported call types:
  - UNSPEC (unspecified)
  - OH (operator handled [0-])
  - OA (operator assisted [0+])
  - DD (direct dial [1+])
  - DELAY
  - 555
  - TS (toll station)
  - TSUB (toll subscriber)
  - 211, 311, 411, 511, 611, 711, 811, 911
  - 999
  - HOM555 (home 555)
  - FOR555 (foreign 555)
  - SPARE1, SPARE2, SPARE3, SPARE4, SPARE5
  - INTS (intercept service)
  - CDIR (country direct)
  - SLRN (Special Local Routing Number)
- The class charge is one of the following:
  - unspecified
  - person paid
  - person collect
  - person special calling
  - person special called
  - station paid
  - station collect
  - station special calling
  - station special called
  - auto collect
- The equal access route status indicates that operator services should be provided for this call.

- AMA verification is not in progress (no validation query outstanding).
- The call *cannot* be one of the following types:
  - — billed coin
  - — billed hotel
  - — notify
  - — time and charges
- If a class charge is entered, a billing number must also be entered. The operator must enter the billing number or remove the class charge before the transfer succeeds.
- If a 4-digit PIN is entered in the special number field, a called number must also be present. The operator must enter the called number or remove the special number before the transfer succeeds.

Operator transfer to SN

The SN or operator can specify the number of operator to SN transitions allowed on a per-call basis. This limits the number of times a call is redirected to an automated system. The remaining number of allowed transitions is sent to the SN by the switch at call arrival.

DAS transfer to SN

For DAS transfers to SNs, the VERSION field in table SERVICES must be set to version three or higher.

Note: For complete information on table SERVICES, please refer to the *Translations Guide*. For details on the DAS protocol, please refer to *Standard Nortel-DMS/DAS Protocol*, Q210-1.

Blocking of OPP requests

When software optionality control (SOC) is enabled (OSAN0002), OSSAIN-related OPP requests from OPP-compatible positions are allowed. When OSSAIN SOC is idle, the requested actions are denied.

Triggers

OSSAIN trigger processing interacts with transitions and routing and queuing during an OSSAIN call. Trigger processing determines the initial OSSAIN control list that applies to a call. Trigger processing also can change the control list or function name during subsequent trigger processing in the call floated PIC.

An OSSAIN transition takes place if the control list or function name is changed during trigger processing. Once the SN is determined, OSSAIN routing and queuing performs data call distribution functions required for that SN.

This section discusses interactions and restrictions that apply to trigger processing in OSSAIN.

Call float interactions with an SN

The SN can float the call under the following conditions:

- It can float the call prior to connecting to the forward party.
- It can float the call after connecting to the forward party, but prior to answer.
- It can float the call after the parties are connected and talking.
- It can float the call if the service type is toll and assist (TA).

Call float interactions with an operator

The operator has two options when attempting to float a call, as follows:

- The operator can connect the parties prior to the float operation. When the operator performs this option, the DMS switch starts trigger processing when the parties are in the talking state—after receiving answer from the terminating party and call float from the operator. No attempt is made to deny the float request from an operator prior to answer.
- The operator can connect the parties as part of the float operation. When the operator performs this option, the DMS switch immediately starts trigger processing as part of the float request.

Operator keyed notifications take precedence over the float trigger notification timer. Likewise, time and charge notifications at disconnect take precedence over the disconnect trigger events.

Triggering directly to an operator

A call can trigger directly to an operator (DA or TA), without triggering to an SN first. The following restrictions apply to calls that trigger directly to an operator:

- When a call triggers to a DA operator and the B-party is still attached, the B-party is released and an AMA record is generated.
- When a call triggers to a TA operator and the B-party is still attached, the B-party can remain with the call, depending on the trigger datafill.
- When directly triggering to an operator, an AMA record is not automatically generated for the portion of a call prior to the trigger. If an AMA record is required for this portion of the call, set the DTMFPRF field in table OADSCPRF to Y and datafill a DTMF profile index from table OADTFPRF. The GENAMA subfield under PROFILE in table OADTFPRF allows an AMA record to be generated during processing of a DTMF trigger event.

  Note: For details on datafill, refer to Chapter 7: “OSSAIN data schema.”
- If a call triggers to an operator during the connecting phase (before the B-party answers), the B-party connection is released and an AMA record is generated before the call is presented to the operator.

- When a call triggers to a TA operator during the talking or disconnecting phase, if the operator wants to start a new call, (such as a sequence TA call), the operator must ensure that the current forward party is released and an AMA record is generated before starting the new call.
- If a call that triggers to a DA operator is eligible for ADAS or ADAS+ service, the service is offered before the call is presented to the DA operator.
- The TOPS call origination type remains the same as when the call triggered, even when a floated toll call triggers to a DA operator. However, the base service of the call changes to DASERV.
- If a call is billed to a calling card or third number, the switch does *not* re-launch another LIDB query. However, the operator can re-class charge the call at the position to force another LIDB query.
- Datafilling a TOPSOPER function or control list against the CLGD (calling disconnect) trigger event in table OACNNPRF is not allowed.
- Triggering to an operator is not allowed when the call is in an OSSAIN conference.

Call float restrictions

The DMS switch performs call floated trigger processing for OSSAIN calls in the call floated state. The following restrictions apply to calls in the floated state:

- Call floated trigger datafill and processing *cannot* be used to route the call to a TOPS automated system (such as ACTS or MCCS/ACCS). An OAIN303 log is produced if a TOPS automated system is encountered during call floated trigger processing.
- If a call is floated prior to reaching the last function in a control list, the remaining functions in that control list are *not* processed. Subsequent OSSAIN processing can only be initiated by a call floated trigger event.
- If a trigger is hit while outpulsing to a forward connection (for example, the calling party enters a DTMF digit) the forward connection is released. This prevents calls from arriving at an SN in a partially connected state. The forward directory number (DN) is part of the call context, so the SN has the ability to establish the forward connection when it has control of the call.

Calls floated by the ACTS automated system *do not* receive OSSAIN trigger processing. This restriction applies because of possible coin recalls during the talking and disconnect points in call.

Voice link broadcasting

The SN is not prohibited from altering the speech path status of any party involved in a call that is using a broadcast voice link. However, the DMS network connection may limit the effectiveness of some requests. For example, a one-way network connection affects the ability of a party to speak or transmit DTMF tones over the voice link.

The following restrictions apply to OSSAIN broadcast voice links:

- In a JNET office, a network subgroup with a port assigned to a DRAM (digital recorded announcement machine) that has broadcasting announcements on it should *not* have any OSSAIN broadcasting voice links mapped to the network subgroup.

  Note: For more information on provisioning broadcast voice links in a JNET office, refer to Appendix C: "Provisioning broadcast voice links in a JNET office."

- Only the OSNM node type can have broadcasting assigned to its voice links. The BCST_SEL field in table OAVLMAP cannot be set to Y for OSAC node types.

- In table OAVLMAP, a logical voice channel (VOICENUM subfield) set up for broadcasting must map to a unique physical trunk member (CLLI and EXTRKNM). Table control denies any attempt to assign a trunk that is already in use by another logical channel. This restriction does not apply to *non*-broadcast voice links, which can have multiple logical channels mapping to the same physical trunk.

- The ability of an SN to cause a voice link to go remote make busy (RMB) *during a call* is lost when the voice link is used for broadcasting. An off hook received from an *idle* broadcast voice link causes it to be put into an RMB state.

  Note: There are no changes for non-broadcast voice links; they still can be RMB with or without a call.

- For OSSAIN calls, three possible entities can use a voice link: an active SN, a passive SN, and a passive operator. When a broadcast voice link is used on a call, it is the *only* voice link allowed. So when a call is using a broadcast voice link to an active or passive SN, it is not possible to add a standard or broadcast voice link to the other SN (passive or active). Also, a passive operator cannot be added.

  Likewise, when a call already has a standard voice link (active or passive) or includes a passive operator, a broadcast voice link cannot be added.

  Note: However, any combination of voice links are possible on a call as long as none of the SN voice connections is a broadcast. OSSAIN VLB does not change the allowable configurations for calls using only standard voice links.

- The loss of a broadcast voice link (such as on a DS-1 carrier failure) may result in lost or ignored messages from a peripheral module involved in a broadcast. Voice link failure causes an error message for *each call* that is currently connected to the voice link. So, voice links with many simultaneous connections, or DS-1 links with more than a single broadcast voice link assigned, will generate a significant amount of messaging to the compute module (CM) when each call detects the problem.

  To limit the possibility of lost messages, the following guidelines should be used when assigning broadcast voice links:

  — Assign only one broadcast voice link on a DS-1, especially if the link has a MAXCONNS value greater than 255.

  — If more than one broadcast voice link is assigned on a DS-1, keep the total number of all broadcast connections on the DS-1 to less than 255.

- The connection of a broadcast voice link disables a subscriber’s ability to transmit voice or DTMF tones (or both) over the voice link, regardless of the status of the subscriber’s speech path. A subscriber can still speak or listen to any other party in the call.
- The SN is responsible for collecting usage counts on voice link trunks used for broadcasting.
- OSAC software release 5 is required for STR digit supervision on broadcast voice links in the OSAC environment.
- For MF trunks, STR supervision listens in on only one port of the connection. The supervised party is the only party able to key the STR digit trigger.

Part 4: Planning and engineering

Part 4: Planning and engineering includes the following chapter:

Chapter 6: “OSSAIN engineering,” beginning on page 237.

Chapter 6: OSSAIN engineering

This chapter provides information on engineering OSSAIN. The following areas are considered:

- OSSAIN system limitations
- hardware resources
- table OFCENG (Office Engineering) parameters
- memory
- messaging and data network
  - engineering and provisioning OSSAIN using Ethernet interface unit (EIU)
  - engineering and provisioning OSSAIN using XA-Core Ethernet interface
  - determining OSSAIN messaging bandwidth requirements

OSSAIN system limitations

The following table lists the maximum number of OSSAIN system components that can be provisioned on a single DMS TOPS switch. Also listed is the data table in which the component is provisioned.

Table 14 OSSAIN system limitations

System component	Maximum number	Table
EIUs	8 (see Note 1)	LIUINV
XA-Core Ethernet Interface cards	8 (see Note 2)	CMIPADDR
SNs	256 (see Note 3)	OANODINV
Session pools	4095 (see Note 4)	OASESNPL
Sessions on a single session pool	1024	OASESNPL
Queuing agents	16383 (see Note 5)	QAPLNDEF
Call queues	255	QAPLNDEF
Call queue profiles	255	OQCQPROF

Table 14 OSSAIN system limitations

System component	Maximum number	Table
Functions	1023	OAFUNDEF
Control lists	4095	OACTLDEF
Trigger profiles	2047	CT4QNAMS
Function blocking profiles	1023	OAFUNBLK

Note 1: This maximum applies to EIUs on the switch. A maximum of only four EIUs can be provisioned on a single LPP; however, up to eight can be spanned across multiple LPPs. A maximum of eight EIUs can be provisioned on a single FLIS or spanned across multiple FLISs. Refer to for information about how traffic is distributed over multiple EIUs.

Note 2: Although up to eight XA-Core Ethernet interface cards can be provisioned, only two are active at any time.

Note 3: This maximum applies to each node type. For example, 256 OSNMs, 256 OSNs, and 256 OSACs can be datafilled on a single switch at the same time. This totals 768.

Note 4: However, the maximum number of session pools for a node is 256.

Note 5: This maximum applies only to queuing agents for session pools with the origination type of SUBSCRIBER (for subscriber originations).

Hardware resources

Engineering discussed in this section covers the following types of hardware resources:

- conference three ports (CF3P)
- specialized tone receivers (STR)
- dual-tone multifrequency (DTMF) receivers
- voice links

Conference three ports

OSSAIN calls require CF3Ps under the following SN connection conditions (with respect to the voice agents connected to the call):

- A party, B party
- SN voice channel, B party
- A party, B party, SN voice channel
- a floated OSSAIN three-way conference with three voice agents
- OSSAIN call with passive operator

In each combination except the last, two or more voice agents are connected to a call and one of the voice agents is the B party.

Calculation

The following formula can be used to calculate the number of CF3Ps to provision on a TOPS switch for OSSAIN calls:

1.5 * (#_OF_SESSNS * AVG_CF3P_%) / 100

where

- #_OF_SESSNS equals the number of OSSAIN call sessions datafilled on the switch (which is the sum of the value in the MAXSESN field for all tuples in table OASESNPL).
- AVG_CF3P_% equals the percentage of time (on average) that one of the connection conditions is in effect while each call is served by an SN (that is, an OSSAIN call session is in use).

To provide for a potential high load condition, a factor of 1.5 is included in the equation. This factor increases the number of conference three ports by 50%.

Sample scenario

In a given TOPS switch, a total of five tuples are in table OASENSPL, each with a value of 24 in the MAXSESN field. So the call sessions (5 x 24) total 120. OSSAIN services are provided by the 120 call sessions, which on average require a CF3P 30% of the time the call is handled by an SN.

Using the calculation, a total of 54 CF3Ps is the result, as follows:

1.5 * (120 * 30) / 100 = 54

In this scenario, the TOPS switch requires an additional 54 CF3Ps for OSSAIN.

OSSAIN conferencing considerations

Additional CF3P circuits are needed for OSSAIN conferencing, because an OSSAIN conference call does not have a session associated with it when it is floated. Traffic volume in CCS (hundred call seconds) also must be considered to determine the number of additional CF3Ps.

To calculate CCS, use High Day Busy Hour (HDBH) or Average Busy Season Busy Hour (ABSBH) for the number of calls per hour. The following formula shows this calculation:

CCS = (# of OSSAIN conference calls per hour) *

(average holding time of the conference call in seconds) / 100

Based on this CCS and the desired grade of service (known as blocking probability), look up the number of required CF3Ps in a standard Poisson traffic table. Add this number to the original CF3P calculation (see “Calculation” on page 239).

Note: *Provisioning Manual Active Detail*, PLN-8991-104, contains a Poisson traffic table.

Specialized tone receivers

Two STR cards must be installed on each digital trunk controller (DTC) configured to carry incoming OSSAIN traffic when the STAR (*) or OCTO (#) triggers (defined in table OATLKPRF) are in effect for calls. One STR card resides in each unit of the DTC-type peripheral, including DTCI and PDTC.

DTMF receivers

During periods when DTMF signaling is permitted or expected, additional DTMF tone receivers (known as RCVRMCCS) must be provisioned. The DTMF receivers support DTMF trigger detection and DTMF digit detection at an SN.

Calculation

The following formula should be added to the existing formula:

(# of simultaneous OSSAIN calls using DTMF trigger detection during the connect or disconnect trigger detection points following call float) +

(# of simultaneous OSSAIN calls that trigger back to an SN with a DTMF receiver connected)

Note: DTMF receivers are not required for calls in the talking state nor for calls that do not use DTMF trigger detection.

Voice links

OSSAIN voice links to are provisioned on Series 1 or Series 2 DMS peripherals. This section discusses engineering considerations for SN voice links (between the standalone or OSAC host switch and SN) and for OSAC voice links (between OSAC remote and OSAC host). In an OSAC call, the remote's voice connection to the SN is via the host; it uses an OSAC voice link in the remote and in the host, and it also uses an SN voice link in the host.

SN voice links

Sufficient voice channels must be provisioned to handle the maximum number of simultaneous calls that need a voice connection between the standalone/host switch and SN. The calculation depends on the use of voice links by the SN applications. Refer to service node engineering documentation.

OSAC voice links

Sufficient host-remote voice channels must be provisioned to handle the maximum number of simultaneous calls between an OSAC host switch and an OSAC remote switch. The following calculation can be used to determine this maximum. In table OASESNPL at the OSAC host switch, add the values of the MAXSESN field for each session pool datafilled as OSACTERM. Likewise, at the OSAC remote switch, add the values of the MAXSESN field for each session pool datafilled as OSACORIG.

Broadcast voice links

In ENET offices, broadcast voice links are assigned by the same rules used for standard, non-broadcast voice links. At most, 1023 simultaneous connections can be made to a broadcast voice link in an ENET office.

Note: In table OAVLMAP, if CUTTHRU is set to HKCHG, the limit for MAXCONN is 255 simultaneous connections.

In JNET offices, 255 simultaneous connections at most can be made to a broadcast voice link. Also, certain engineering guidelines must be followed to ensure that no more than 256 (for NT0X48 networks) or 512 (for NT5X13 or NT8X11 networks) simultaneous connections (broadcast and non-broadcast) occur on a network subgroup (NSG). Refer to Appendix C: “Provisioning broadcast voice links in a JNET office,” for details.

Table OFCENG parameters

OSSAIN processing requires two parameters in table OFCENG, as follows:

- OSSAIN_NUM_RU
- OSAC_NUM_RU

In addition, OSSAIN changes the provisioning and calculation for three parameters: CRS_SUBRU_POOL1_SIZE, CRS_SUBRU_POOL2_SIZE, and CRS_SUBRU_POOL4_SIZE. This section provides a description of each parameter.

OSSAIN_NUM_RU

This parameter determines how many OSSAIN recording units (extension blocks) are needed for processing OSSAIN call traffic. Each active OSSAIN call requires an OSSAIN recording unit (RU).

Table 15 Parameter OSSAIN_NUM_RU

Parameter name	Range of values/ units	Default value	Explanation
OSSAIN_NUM_RU	0 to 32767	100	This parameter determines how many OSSAIN recording units are needed for processing OSSAIN call traffic.

Method of change activation

Although activation is immediate for either an increase or a decrease to OSSAIN_NUM_RU, store is not automatically deallocated when a decrease occurs. The reason is that calls in progress may be using the affected store. A subsequent cold restart or reload restart causes the store to be deallocated.

Calculation

Each active OSSAIN call requires an OSSAIN recording unit. The following calculation determines how many active OSSAIN calls are possible in the switch (the resulting number can help to provision the number of OSSAIN RUs):

[(% of calls incoming on TOPS trunks that route to OSSAIN) * (number of TOPS trunk group members)] +

[(% of calls incoming on ATC trunks that route to OSSAIN) * (number of ATC trunk group members)] +

[(% of calls incoming on IT trunks that route to OSSAIN) * (number of IT trunk group members)] +

(the sum of MAXSESN in all OSSAIN session pools used for SN originations)

Memory requirements

Each OSSAIN recording unit requires 250 words (500 bytes) of memory. The memory requirement for OSSAIN RUs is 100 times the value of this parameter, as follows:

units per memory block = 1

words per memory block = 250

So, if the parameter value is 250, then 25,000 words are required.

Verification

To verify that OSSAIN recording units have been provisioned sufficiently, use the CI command OMSHOW EXT ACTIVE OSSAINRU. An entry similar to the following will be displayed.

Figure 103 MAP display example for OM group EXT

```
           EXTSEIZ    EXTOVFL    EXTHI   EXTSEIZ2
           EXTHI2
    139 OSSAINRU
              100
                0         0        0          0
                0
```

Any nonzero value in field EXTOVFL indicates under-provisioning. Refer to the *Operational Measurements Reference Manual* for a complete description of the EXT OM group.

Consequences

Over-provisioning the value of OSSAIN_NUM_RU causes memory to be allocated that may not be used. Under-provisioning the value causes OSSAIN calls not to be processed if they fail to obtain an OSSAIN RU.

An OAIN301 log is generated when the switch has trouble allocating the resource, resulting in the call being routed to the “no software resources” (NOSR) treatment.

Duration

When it is determined that a call requires OSSAIN processing, an OSSAIN RU is obtained. It is held for the remainder of the call (that is, until the call is taken down).

Operational measurements

This parameter is associated with the EXT OM group.

Extensions

The number of OSSAINRU extension blocks available on a switch is affected by this parameter.

OSAC_NUM_RU

This parameter determines how many OSAC recording units (extension blocks) are needed for processing OSAC call processing in the OSAC host switch.

A value for this parameter is needed only at the OSAC host switch. An OSAC host switch has the SWTCHTYP subfield in table OASESNPL set to SH (for host). If the switch is purely a standalone (S) or an OSAC remote switch (R), then the parameter can have the default value of zero.

Table 16 Parameter OSSAIN_NUM_RU

Parameter name	Range of values/ units	Default value	Explanation
OSAC_NUM_RU	0 to 32767	0	This parameter determines how many OSAC recording units are needed for processing OSAC calls in the OSAC host switch.

Method of change activation

Although activation is immediate for either an increase or a decrease to OSAC_NUM_RU, store is not automatically deallocated when a decrease occurs. The reason is that calls in progress may be using the affected store. A subsequent cold restart or reload restart causes the store to be deallocated.

Calculation

Each call at the OSAC host switch requires an OSAC recording unit. The calculation is determined by adding the values of the MAXSESN field in table OASESNPL *for each session pool datafilled as OSACTERM at the OSAC host*. The resulting number can help to provision the number of OSAC RUs.

Memory requirements

Each OSAC recording unit requires 50 words (100 bytes) of memory. The memory requirement for OSAC RUs is 100 times the value of this parameter, as follows:

units per memory block = 1

words per memory block = 50

So, if the parameter value is 100, then 5,000 words are required.

Verification

To verify that OSAC recording units have been provisioned sufficiently, use the CI command OMSHOW EXT ACTIVE OSACRU. An entry similar to the following will be displayed.

Figure 104 MAP display example for OM group EXT

```
              EXTSEIZ    EXTOVFL    EXTHI   EXTSEIZ2
              EXTHI2
     161 OSACRU
                 100
                 526         0        0          1
                   0
```

Any nonzero value in field EXTOVFL indicates under-provisioning. Refer to the *Operational Measurements Reference Manual* for a complete description of the EXT OM group.

Consequences

Over-provisioning the value of OSAC_NUM_RU causes memory to be allocated that may not be used. Under-provisioning the value causes OSAC calls not to be processed if they fail to obtain an OSAC RU.

An OAIN301 log is generated when the switch has trouble allocating the resource, resulting in the call being routed using table OAFNDISP (at the remote).

Duration

An OSAC RU is obtained at the OSAC host switch when the switch receives a session request from an OSAC remote switch. This applies to both subscriber originations and SN initiations. The OSAC RU is held until the session between the OSAC remote switch and the OSAC host switch ends.

Operational measurements

This parameter is associated with the EXT OM group.

Extensions

None

CRS_SUBRU_POOL*_SIZE parameters

The extension blocks that these parameters govern are used for OSSAIN calls.

Autoprovisioning

Parameters CRS_SUBRU_POOL1_SIZE, CRS_SUBRU_POOL2_SIZE, and CRS_SUBRU_POOL4_SIZE may be autoprovisioned in table OFCAUT (Office Autoprovisioning), instead of manually provisioned in table OFCENG (Office Engineering). When the operating company sets the corresponding ACTIVE field in table OFCAUT to Y, this datafill instructs the system to remove the selected parameter from table OFCENG for autoprovisioning in table OFCAUT. When the ACTIVE field in table OFCAUT is set to N, the system adds the selected office parameter back to table OFCENG for manual control. For details on table OFCAUT, please refer to the *Customer Data Schema Reference Manual*.

CRS_SUBRU_POOL1_SIZE

The following paragraphs describe the provisioning and calculation changes OSSAIN makes to CRS_SUBRU_POOL1_SIZE.

Provisioning rules

This extension block is used by AMA to store the Network Service ID of the OSSAIN call. Each call requires only one extension block.

Calculation

The following formula should be added to the existing formula:

<existing equation> + # of OSSAIN calls

CRS_SUBRU_POOL2_SIZE

The following paragraphs describe the provisioning and calculation changes OSSAIN makes to CRS_SUBRU_POOL2_SIZE.

Provisioning rules

This extension block is used by AMA to store SN information related to OSSAIN calls. An extension block is used for each SN involved in the call. The maximum number of SNs allowed in a call can be determined by the MAX_ALLOWED_TRANSITIONS parameter in table OAINPARM. Refer to “OAINPARM parameters” on page 357 for more information on parameter restrictions.

This extension block is also used to store SN formatted billing data (sent through OAP or OPP) that is less than or equal to 20 bytes. An extension block is used for each Append AMA Module request that contains AMA data less than or equal to 20 bytes.

This extension block is also used to store the authorization code for commercial credit cards which will generate module code 305 and the extended account code generated in module code 103. An extension block is used for each Authorization Code request that contains commercial credit card and for each Account Code which contains more than 4 digits.

Note: Refer to Chapter 9: “OSSAIN billing,” for more information on AMA.

Calculation

The following formula should be added to the existing formula:

<existing equation> +

1 * (# of OSSAIN calls involving 1 SN) +

2 * (# of OSSAIN calls involving 2 SNs) +

. . . .

[(MAX_ALLOWED_TRANSITIONS) * (# of OSSAIN calls involving MAX_ALLOWED_TRANSITIONS SNs)] +

[(MAX_ALLOWED_TRANSITIONS + 1) * (# of OSSAIN calls involving MAX_ALLOWED_TRANSITIONS + 1 SNs)] +

(# of TOPS OSSAIN Append AMA Module requests with AMA data less than or equal to 20 bytes) +

(# of TOPS OSSAIN Authorization Code requests with commercial credit cards type) +

(# of TOPS OSSAIN Account Code requests with more than 4 digits)

CRS_SUBRU_POOL4_SIZE

The following paragraphs describe the provisioning and calculation changes OSSAIN makes to CRS_SUBRU_POOL4_SIZE.

Provisioning rules

This extension block is used to store SN formatted billing data (sent through OAP or OPP) that is from 21 to 70 bytes. An extension block is used for each Append AMA Module request that contains AMA data that is from 21 to 70 bytes.

Calculation

The following formula should be added to the existing formula:

<existing equation> +

# of TOPS OSSAIN Append AMA Module requests with AMA data from 21 to 70 bytes

Memory

Memory considerations for engineering OSSAIN involve estimating the data store impact. OSSAIN increases the required data store for both initial program load (IPL) and datafill.

This section discusses the *estimated* IPL data store impact and datafill data store impact of OSSAIN, and lists reasons for the increases.

IPL data store impact

A total of 2600 kilobytes is estimated for OSSAIN.

Reasons for increases in IPL data store

OSSAIN functionalities increase data store allocation at IPL time for several reasons, including the following:

- allocating the first level descriptors for OSSAIN call types for queuing (CT4Q) in table CT4QNAMS
- allocating the first level descriptors for OSSAIN call profile indexes in table OATPRFIX
- allocating the internal tables for the OAP protocol

Maximum datafill data store impact

A total of 98,700 kilobytes is estimated for OSSAIN.

Reasons for increases in datafill data store

OSSAIN functionalities implement data tables that may require extensive store allocation if they are datafilled to their maximum capacity. Many tables (such as OANODINV and OASESNPL) are dynamically allocated to reduce store use.

OSSAIN maintenance also supports unsolicited messaging from OSNM nodes and OSAC nodes. This store is dynamically allocated to minimize store use.

Note 1: To avoid reaching the maximum store when provisioning nodes in table OANODINV, do not arbitrarily assign high node identifiers (NODEID field in table OANODNAM).

Note 2: To avoid reaching the maximum store when provisioning session pools in table OASESNPL, do not arbitrarily assign high session pool identifiers (SESNPLID field).

When adding the OSSAIN application tuple in table QAPLNDEF, additional data store is allocated for queue and agent related data. The following table lists the limits for items datafilled in table QAPLNDEF.

Table 17 Data store impact for table QAPLNDEF

Item in QAPLNDEF	Maximum # of items	Field name in QAPLNDEF
Call queues	255	CALLQS
Call queue elements	32767	CQELEMS
Agent queues	255	AGENTQS
Number of agents	16383	NUMAGNTS
Note: For complete information on table QAPLNDEF, please refer to the *Customer Data Schema Reference Manual*.		

Messaging and data network—introduction

An OSSAIN standalone or OSAC host switch uses an Ethernet local area network (LAN) for maintenance and call processing messaging with SNs. Each SN connects to the LAN using standard Ethernet technology. The switch may connect to the LAN through either EIUs or, if it has an XA-Core, through XA-Core Ethernet interface cards such as HIOP and HCMIC. The XA-Core Ethernet interface has much higher messaging capacity than EIUs have.

If OSSAIN centralization is used, OSAC switches use a wide area network (WAN) for messaging among themselves. The WAN is also used for messaging between OSAC remote switches and centralized SNs.

OSSAIN User's Guide does not include information about engineering the WAN. OSSAIN imposes no constraints on the WAN technology as long as it has sufficient bandwidth and reliability to handle the messaging.

It is strongly recommended that the data network be designed by a professional with appropriate qualifications and experience. Careful thought should be given to the IP address plan, if one has not already been developed, to achieve efficient routing and to allow for future expansion. It is difficult and time-consuming to change an established IP address plan.

The following two sections address engineering and provisioning topics specific to EIU and XA-Core Ethernet interface implementations, respectively. They include engineering restrictions and detailed descriptions of the datafill that is specific to EIU and to XA-Core Ethernet interface.

A section on determining OSSAIN messaging bandwidth requirements follows the sections that are specific to EIU and XA-Core Ethernet interface.

The reader is assumed to be familiar with basic concepts of IP addressing, including subnetting. Please refer to Appendix A: "OSSAIN data communications" for a more tutorial introduction. That appendix also includes other information, such as UDP port numbers, that may be useful to OSSAIN data network designers and engineers.

EIU engineering and provisioning

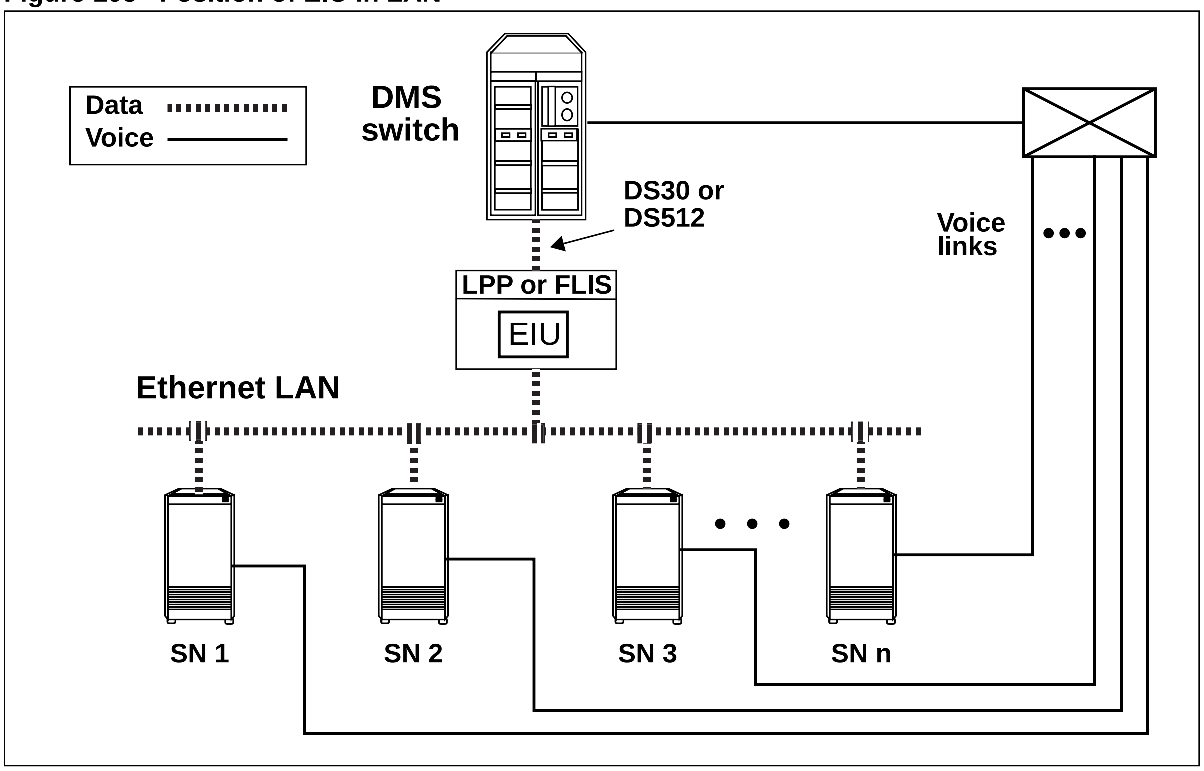
This section discusses the following topics related to use of EIUs in the OSSAIN network:

- IP address restrictions
- routing restrictions
- capacity restrictions
- example IP address plans using EIU
- datafilling EIUs
- table interdependency and autoconfiguration

Note: For more information about the EIU, please refer to 297-8991-910, *Ethernet Interface Unit User Guide*.

EIUs are provisioned on either a Link Peripheral Processor (LPP) or a Fiber Link Interface Shelf (FLIS). LPPs connect to the message switch using DS30s. FLISs connect using DS512s. The following figure shows a simplified view of the position of the EIU relative to the switch and the LAN.

Figure 105 Position of EIU in LAN



A switch can support a maximum of eight EIUs. Only four EIUs can be provisioned on a single LPP; however, up to eight can be spanned across multiple LPPs. A maximum of eight EIUs can be provisioned on a single FLIS or spanned across multiple FLISs.

EIUs act as routers between the DMS CM and the LAN subnets. They are not intended to be used as routers between different LANs.

Note: Although some other DMS applications support the use of EIUs as interfaces, OSSAIN does not support that. OSSAIN uses EIUs only as routers.

IP address restrictions

Use of EIUs for SuperNode LAN connectivity imposes certain restrictions on the IP address plan for the OSSAIN network.

- Classes A, B, and C address schemes are supported. Classes D and E are not supported.
- IP addresses for all DMS SuperNode hosts (such as CM, EIUs, and APUs) are assigned through DMS table control in tables IPNETWRK and IPROUTER.

  Note: Refer to “Datafilling EIUs” on page 256 for details on the DMS IP tables.
- IP addresses for all DMS SuperNode hosts are on a *single* subnet.
- Each EIU is assigned two IP addresses: one to address the SuperNode side subnet and one to address the Ethernet LAN side subnet.
- Each EIU LAN side IP address must be assigned the same network class and network number as the EIU SuperNode side IP address.
- The EIU LAN side IP address and the SuperNode side IP address must have different subnet IDs.
- Multiple Lan side subnets are supported.

Routing restrictions

Although some DMS applications allow EIUs to be used as interfaces, OSSAIN does not support that configuration. For OSSAIN, EIUs must be configured as routers. The following restrictions apply.

- A load balancing scheme among multiple EIU IP routers sets specific EIUs as primary routers for outgoing data. For incoming data, specific EIUs proxying for host nodes in the DMS SuperNode subnet perform load balancing when multiple SuperNode hosts are involved. The load balancing scheme is updated as host nodes are provisioned in table IPHOST. However, since the CM host is the only SuperNode host used by OSSAIN, the load balancing scheme does not evenly distribute OSSAIN messaging traffic over multiple EIUs.
- For each LAN side subnet, a single EIU is internally designated by the routing software as the primary EIU.
- All IP messages sent from the CM to hosts on a LAN side subnet are routed through the primary EIU for that subnet. The primary EIU also handles traffic to the CM from nodes on that subnet.

- Upon failure of the primary EIU for a LAN side subnet, traffic is switched to another EIU on the same subnet, if one is available (in service). This becomes the primary EIU for the subnet.
- The purpose of the EIU is to provide DMS SuperNode connectivity to a LAN. It is not intended to act as a router between two LANs.
- The industry standard Routing Information Protocol (RIP) is implemented on EIUs. RIP allows the EIU to exchange dynamic routing information. Dynamic routing information is required on the DMS SuperNode to route datagrams to hosts on LANs other than the LAN where the EIUs are connected).
- EIUs will proxy Address Resolution Protocol (ARP) for other DMS SuperNode IP hosts.
- A single EIU is designated in table IPNETWRK as the default router. IP messaging between the CM and nodes *not* on one of the LAN subnets uses the default EIU router, if that router can reach the far-end IP address. If not, the dynamic routing table is used to determine whether some other EIU router can reach the far-end address.
- An SN must not be on the same subnet as a core Ethernet interface unless it is intended that the core use that interface, and its associated core IP address, when sending messages to that SN. A local subnet check in the core routing software will select the core interface rather than the EIU router if a local subnet match is found. This applies both to XA-Core Ethernet interfaces and to EIUs that are configured as interfaces instead of routers, although whether a local subnet match is found for the XA-Core Ethernet interface depends on interactions with other features.

Capacity restrictions

Please refer to page 272 for information about determining OSSAIN messaging bandwidth requirements. Once the requirement has been determined, it is necessary to take into account the routing restrictions related to EIUs to determine whether EIUs can provide the capacity needed and, if so, how many LAN side subnets are needed.

- Throughput capacity of an EIU varies from about 10 to about 50 kilobytes per second in each direction.
- The main factor that accounts for variation in EIU capacity is how heavily the LPP or FLIS is used, other than for the EIU under consideration.
- Everything else being equal, EIU capacity is somewhat lower with LPP than with FLIS.
- Table IPTHRON should be datafilled to throttle the rate of messaging though EIUs.

Example IP address plans using EIUs

The following subsections illustrate two examples of IP address plans using EIUs for OSSAIN.

Network 1—one SN subnet

This example network consists of a single switch and four SNs. Class B addressing is used and the network ID is 128.43. An 8-bit subnet mask is used, allowing 255 subnets and 255 host IDs for each subnet.

Network 1 requires a total of two subnets: one for the DMS SuperNode internal LAN (subnet 1), and one for the four SNs (subnet 2). IP address assignments are listed in the following table.

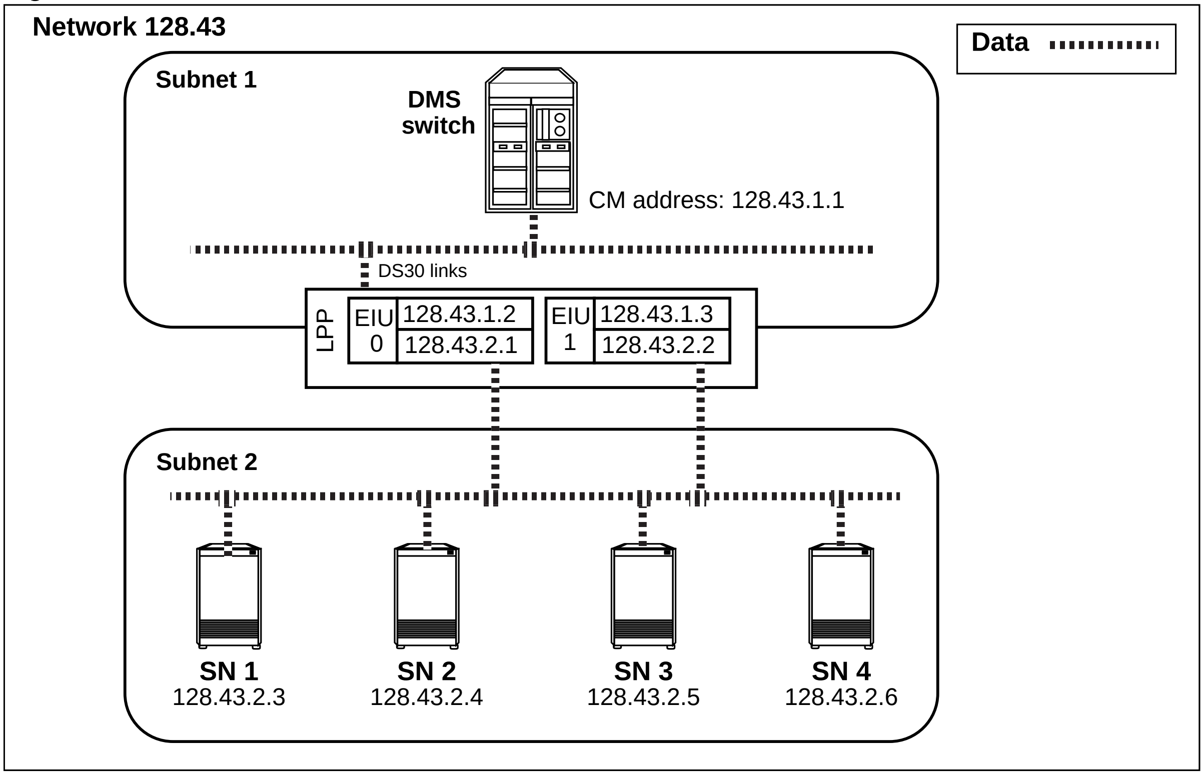
Table 18 IP address assignments for Network 1

Address	Description
128.43.1.1	DMS CM
128.43.1.2	EIU 0 SuperNode side
128.43.1.3	EIU 1 SuperNode side
128.43.2.1	EIU 0 LAN side
128.43.2.2	EIU 1 LAN side
128.43.2.3	SN 1
128.43.2.4	SN 2
128.43.2.5	SN 3
128.43.2.6	SN 4

The following figure illustrates Network 1, showing the IP address assignments in subnet 1 and subnet 2, and two EIUs, both of which connect to the SN subnet (subnet 2).

Note: Voice links are not shown in the figure.

Figure 106 Network 1



EIU routing software uses only one EIU at a time to carry IP traffic to its LAN subnet. This EIU is referred to as the *primary* EIU.

When a second EIU is connected to the LAN subnet, as is done in this example, it runs in a hot standby mode. In this mode, the second EIU does not carry IP traffic until the primary EIU is taken out of service. If the primary EIU is taken out of service, the hot standby EIU becomes the primary EIU.

Network 2—two SN subnets

When determining the number of subnets required in the OSSAIN network, a factor to consider is the traffic load on the EIU. When the traffic load generated by the SNs is capable of overloading the EIU, a second SN subnet (with its own primary and standby EIUs) can share the traffic load.

Note: When the number of SNs in the network is increased, adding a second SN subnet can serve to balance the IP traffic across the network.

Network 2 has a second SN subnet. This example network consists of a single switch and eight SNs. Once again, class B addressing is used and the network ID is 128.43. An 8-bit subnet mask is used, allowing 255 subnets and 255 host IDs for each subnet.

Network 2 requires a total of three subnets: one for the DMS SuperNode internal LAN (subnet 1), one for four SNs (subnet 2), and one for the other four SNs (subnet 3). IP address assignments are listed in the following table.

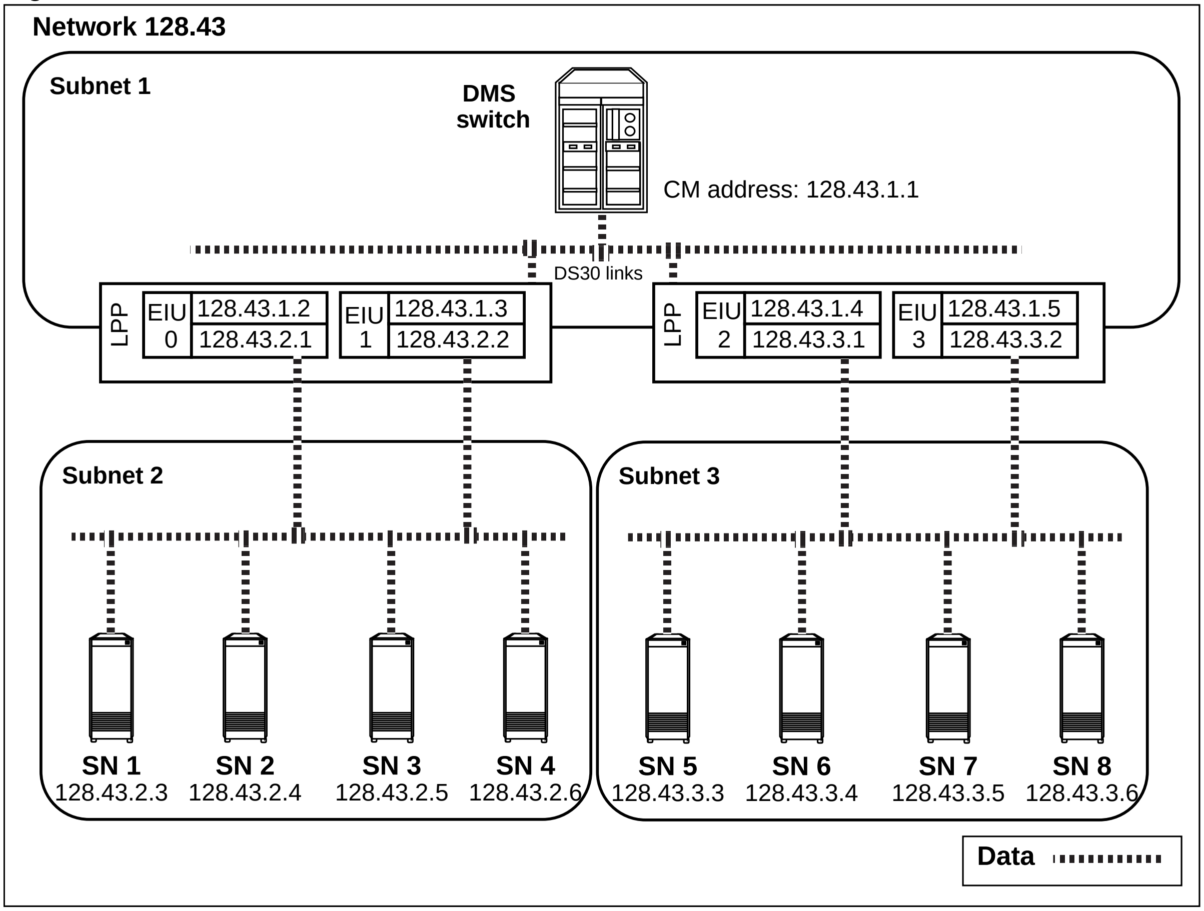
Table 19 IP address assignments for Network 2

Address	Description
128.43.1.1	DMS CM
128.43.1.2	EIU 0 SuperNode side
128.43.1.3	EIU 1 SuperNode side
128.43.1.4	EIU 2 SuperNode side
128.43.1.5	EIU 3 SuperNode side
128.43.2.1	EIU 0 LAN side
128.43.2.2	EIU 1 LAN side
128.43.3.1	EIU 2 LAN side
128.43.3.2	EIU 3 LAN side
128.43.2.3	SN 1
128.43.2.4	SN 2
128.43.2.5	SN 3
128.43.2.6	SN 4
128.43.3.3	SN 5
128.43.3.4	SN 6
128.43.3.5	SN 7
128.43.3.6	SN 8

The following figure illustrates Network 2, showing the IP address assignments in subnet 1, subnet 2, subnet 3; and four EIUs, two connecting to SN subnet 2 and two connecting to SN subnet 3.

Note: Voice links are not shown in the figure.

Figure 107 Network 2



In Network 2, two EIUs support subnet 2 and two support subnet 3. IP traffic between the DMS SuperNode and subnet 2 is exchanged on either EIU 0 or EIU 1; and IP traffic between the DMS SuperNode and subnet 3 is exchanged on either EIU 2 or EIU 3.

Again, one EIU is designated the primary EIU for message traffic to the LAN subnet it supports. The other EIU supporting the LAN subnet runs in a hot standby mode, coming into service if the primary EIU is taken out of service.

Datafilling EIUs

The translations tables for EIU Internet connectivity in OSSAIN are described in the following table. The tables are listed in the order in which they are to be datafilled.

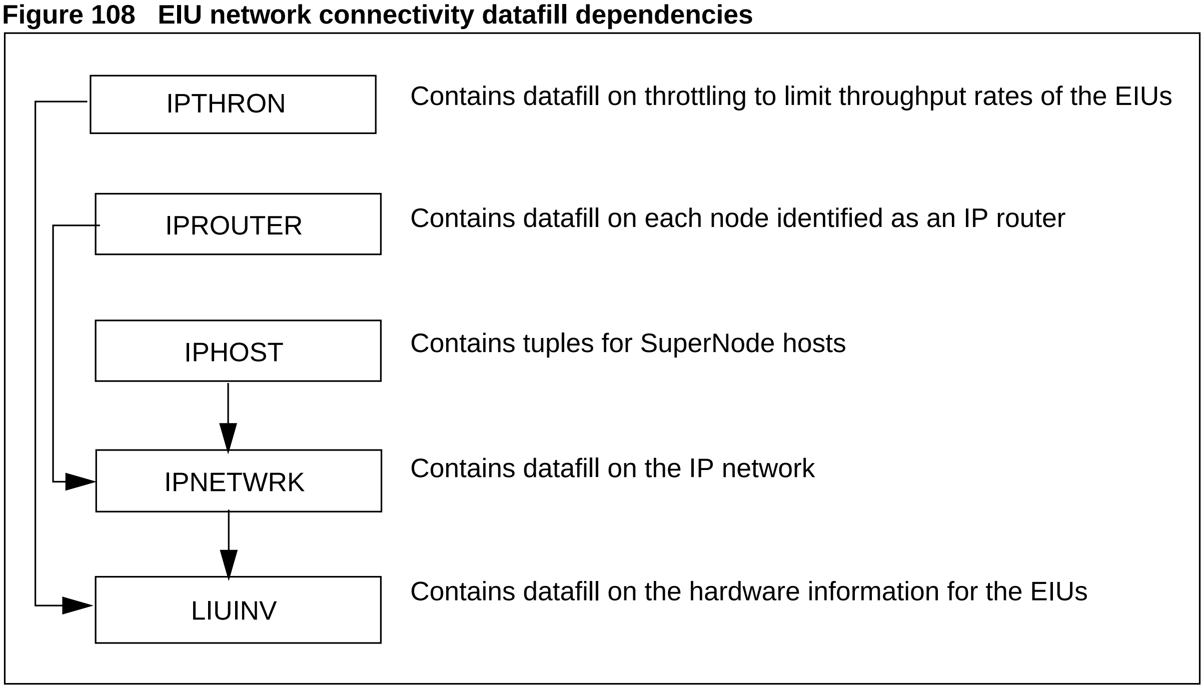
Table 20 Datafill sequence for EIU network connectivity

Table name	Explanation
LIUINV	The Link Interface Unit Inventory table contains information on the LIU processor types (such as the EIU) in the link peripheral processor (LPP).
IPNETWRK	The Internet Protocol Network table contains information on the IP network. IP screening also is enabled in this table.
IPHOST	The IP Host table must contain datafill for the CM in tuple 0.
IPROUTER	The Internet Protocol Router table assigns IP addresses to each node identified as an IP router.
IPTHRON	The Internet Protocol Throttling Numbers table contains the IP throttling numbers used by the DMS SuperNode host. Throttling limits the IP throughput rates of the EIUs to minimize the risk of overloading LPP resources.

Note: For more information about the EIU, please refer to 297-8991-910, *Ethernet Interface Unit User Guide*.

The following figure shows table dependencies for the five tables requiring datafill for EIU Internet connectivity. The arrows indicate a need relationship.

Figure 108 EIU network connectivity datafill dependencies



The following subsections provide information on how to datafill each table.

Note: In the MAP display example for each IP table that follows, the sample datafill is from OSSAIN Network 1. For an illustration of Network 1, refer to Figure 106, “Network 1,” on page 253.

LIUINV

Table LIUINV specifies the hardware configuration for the EIUs. Currently, a maximum of eight EIUs are allowed for each DMS SuperNode. Before datafilling table LIUINV, the unique Ethernet address must be determined.

Note: A maximum of only four EIUs can be provisioned on a single LPP; however, up to eight can be spanned across multiple LPPs. A maximum of eight EIUs can be provisioned on a single FLIS or spanned across multiple FLISs.

Table 21 Datafilling table LIUINV

Field	Subfield or refinement	Entry	Explanation and action
LIUNAME		See subfields	Link interface unit name. This field uniquely identifies the EIU present in the LPP and consists of subfields LIUTYPE and LIUNO.
	LIUTYPE	EIU	Link interface unit type. Enter EIU, which is the type of LIU.
	LIUNO	0 to 511	Link interface unit number. Enter the EIU number.
LOCATION		See subfields	Location. This field identifies the physical location on the host link interface module (LIM) and consists of the subfields CTRL, SHELFNUM, and LIUSLOT.
	CRTL	See subfields	Control. This subfield specifies the controlling host entity for the EIU and consists of two subfields, CONTROL and LIMNUM.
	CONTROL	LIM	Controlling host entity. Enter LIM and also datafill the LIMNUM refinement.
	LIMNUM	0 to 16	Link interface module number. Enter the LIM number where the LIU resides.
	SHELFNUM	0 to 3	Shelf number. Enter the shelf number at the host where the EIU resides.
	LIUSLOT	8 to 31	Link interface unit slot. Enter the LIU slot number at the host where the EIU resides.
LOAD		Alphanumeric up to 8 characters	Load. Enter the software load name for the EIU. ***Note:*** The value for the LOAD field is from table PMLOADS.

Table 21 Datafilling table LIUINV

Field	Subfield or refinement	Entry	Explanation and action
PROCINFO		See subfield	Processor information. This field consists of the subfield PROCPEC.
	PROCPEC	NBTEX22AA, NTEX22BA, NTEX22BB	Processor product engineering code. Enter the PEC for the processor circuit pack.
CARDINFO		See subfields	Card information. This field specifies the PEC of the EIU circuit packs and consists of the subfield APPLPEC and its refinements.
	APPLPEC	NT9X84AA	Application PEC. Enter NT9X84AA and also datafill the PBPEC, HEARTBEAT, and MAC_ADDRESS refinements.
	PBPEC	NT9X85AA, NT9X95BA	Paddleboard PEC. Enter the paddleboard PEC code.
	HEARTBEAT	YES, NO	Heartbeat. Enter YES or NO to indicate whether or not the Ethernet interface card should expect a heartbeat indication from the media access unit (MAU) on the LAN. ***Note:*** A value of YES is allowed only if the MAU supports heartbeat of signal quality error.
	MAC_ ADDRESS	12 hexadecimal digits	Media access control address. Enter the MAC address in the following form: 000075Fxxxxx. ***Note:*** The MAC address assignment is managed by Nortel installation. Arbitrary Ethernet addresses should not be assigned to the EIUs.

LIUINV example

The following figure shows example datafill.

Figure 109 MAP display example for table LIUINV

```
LIUNAME    LOCATION    LOAD            PROCINFO
                              CARDINFO
------------------------------------------------------------------
EIU 0      LIM 0 1 12  ETC05BC         NTEX22BB
                              NT9X84AA  NT9X85AA  YES 000075F07112
EIU 1      LIM 0 1 16  ETC05BC         NTEX22BA
                              NT9X84AA  NT9X85AA  YES 000075F07116
```

IPNETWRK

Table IPNETWRK contains information on the IP network and the DMS SuperNode subnetwork connected to the LAN or LANs. The first tuple in table IPNETWRK is reserved for the default EIU router. The table allows other tuples to be added for core Ethernet interfaces, but these other tuples are not applicable to OSSAIN.

Note: OSSAIN nodes should not be placed on the subnets of any interface tuples that are added to table IPNETWRK for other applications.

Table IPNETWRK also controls IP screening. With IP screening enabled, IP communication is allowed only between the DMS switch and the OSSAIN nodes datafilled in table OANODINV. Furthermore, enabling IP screening causes the EIU to screen messages to the CM if the sending node is OFFL or if it is SYSB and has been declared a babbler.

Note: When an SN has been declared a babbler, the SYSB reason is “failed message threshold exceeded.” Failed message detection is enabled by fields FDETECT and FTHRESH in table OANODINV. For more information on datafilling table OANODINV, refer to Chapter 7: “OSSAIN data schema.”

With IP screening disabled, IP communication is not restricted and the DMS switch is allowed to communicate with *any node* that has access to the LAN.

Table 22 Datafilling table IPNETWRK

Field	Subfield or refinement	Entry	Explanation and action
KEYREF		See subfield	Key reference. This field consists of the subfield TAB_KEY.
	TAB_KEY	0	Table key. Enter the index into table IPNETWRK. The default EIU router is datafilled with index 0.
CMIPADDR		0 to 255 for each of the 4 address parts	Computing module IP address. This field identifies the IP address of the DMS-Core (the CM). Values are expressed in the form x x x x separated by spaces. Enter the CM IP address.
SUBNET		1 to 23	Subnet size. Enter the bitsize of the of the DMS SuperNode subnet.
OPTION		See subfields	Option. This field specifies the default EIU (from table LIUINV) and consists of the subfields WORD_EIU and EIU_RNG.
	WORD_EIU	EIU	Word EIU. Enter EIU.
	EIU_RNG	0 to 750	EIU range. Enter the number of the EIU.
PARMAREA		See subfields	Parameter area. This field consists of subfield PARMAREA.

Table 22 Datafilling table IPNETWRK

Field	Subfield or refinement	Entry	Explanation and action
	PARM	NULLPARM, SCRNFLAG	Parameter. This field is used to indicate whether a screening flag is required. Enter NULLPARM if screening is not required. Enter SCRNFLAG if screening is required and also datafill the SCRNFLAG refinement.
	SCRNFLAG	N, Y	Screening flag. Enter Y to enable screening, or N to disable screening. (Screening is explained just above this table.)

IPNETWRK example

The following figure shows example datafill.

Figure 110 MAP display example for table IPNETWRK

```
KEYREF   CMIPADDR         SUBNET    OPTION         PARMAREA
----------------------------------------------------------------------
0        128 43 1 1       8        (EIU 0) $      (SCRNFLAG Y) $
```

IPHOST

Correct operation of the IP stack in the switch requires that a CM tuple be present in table IPHOST. For most SuperNode hosts, such as application processors (AP) and file processors (FP), table IPHOST assigns the host IP address. For the CM host, however, the IP address is assigned in table IPNETWRK (see above). The CM tuple in table IPHOST is used primarily for TCP (transmission control protocol) and FTP (file transfer protocol) session control. Since OSSAIN does not use FTP, and since it uses TCP only for OSSAIN QMS MIS, the session control parameters may be datafilled with 0 in most offices.

WARNING: Setting or changing these values to 0 will impact TCP endpoints and the applications that use them. For more informations, please refer to the *Ethernet Interface Unit Users Guide*, 297-8991-910, or the next level of support.

Table 23 Datafilling table IPHOST

Field	Subfield or refinement	Entry	Explanation and action
INDEX		0	The index field identifies the tuple. Enter 0 to datafill a tuple for the CM.

Table 23 Datafilling table IPHOST

Field	Subfield or refinement	Entry	Explanation and action
NODENAME		CM	Node name CM is a selector entry that controls which refinement fields are present in the NODEINFO field.
NODEINFO		See subfields	Node information.
	CMINDEX	0 to 1	Enter 0.
	TCPCONN	0 to 96	Transmission control protocol connections. Enter the maximum number of TCP connections.
	FTPCLCON	0 to 48	File transfer protocol connections. Enter the maximum number of FTP client sessions.
	FTPSVCON	0 to 48	File transfer protocol server connections. Enter the maximum number of FTP server sessions.

IPHOST example

The following figure shows example datafill.

Figure 111 MAP display example for table IPHOST

```
INDEX          NODENAME                NODEINFO
-------------------------------------------------------------------
0              CM   0                  0   0   0
```

IPROUTER

Table IPROUTER contains the list of EIUs and their corresponding parameters to configure the EIU as an Internet node. An EIU can be configured as a host, a router, or both. OSSAIN requires EIUs to be configured as routers.

Table 24 Datafilling table IPROUTER

Field	Subfield or refinement	Entry	Explanation and action
RKEY		0 to 63	Router key. Enter the router key.
ROUTER		See subfields	Router. This field specifies the EIU router (from table LIUINV) and consists of the subfields WORD_EIU and EIU_RNG.
	WORD_EIU	EIU	Word EIU. Enter EIU.
	EIU_RNG	0 to 750	EIU range. Enter the number of the EIU.

Table 24 Datafilling table IPROUTER

Field	Subfield or refinement	Entry	Explanation and action
SNIPADR		0 to 255 for each of the 4 address parts	SuperNode IP address. This field identifies the DMS SuperNode side IP address of the EIU router. Values are expressed in the form x x x x separated by spaces. Enter the SuperNode side IP address.
ETHIPADR		0 to 255 for each of the 4 address parts	Ethernet LAN IP address. This field identifies the Ethernet LAN side IP address of the EIU router. Values are expressed in the form x x x x separated by spaces. Enter the Ethernet LAN side IP address.
EHTARP		YES, NO	Ethernet address resolution protocol. Enter YES or NO to indicate whether or not the EIU participates in address resolution protocol (ARP) activity in the Ethernet subnet.
ETHPARP		YES, NO	Ethernet proxy address resolution protocol. Enter YES or NO to indicate whether or not the EIU participates in proxy ARP activity on behalf of the SuperNode in the Ethernet subnet.

IPROUTER example

The following figure shows example datafill.

Figure 112 MAP display example for table IPROUTER

```
ROUTER      SNIPADR      ETHIPADR           ETHARP   ETHPARP
------------------------------------------------------------------
1  EIU 0    128 43 1 2   128 43 2 1         YES      YES
2  EIU 1    128 43 1 3   128 43 2 2         YES      YES
```

IPTHRON

Table IPTHRON contains the IP throttling numbers used by DMS SuperNode hosts to control congestion in the shared communication resources between the local message switch and the message switch.

For EIUs, the throttling numbers represent the number of kilobytes per second of data that can be transmitted and received by the EIU. As long as the offered messaging traffic from the network does not exceed the datafilled level, the data will be immediately forwarded to the CM. The effect of throttling is to smooth out the rate at which the messages are delivered. Inbound data will be buffered by the EIU when necessary and eventually forwarded to the CM.

If the number of offered messages exceeds the limit over a sustained period, the EIU's buffers may become full and the overflow messages will be discarded. Outbound data will be buffered by the CM when necessary and eventually forwarded to the EIU.

IP throttling numbers in table IPTHRON default to zero, which is 100% throttling, for all EIUs datafilled in table LIUINV. A non-zero value must be datafilled in table IPTHRON to allow communication.

Values between approximately 10 and 50 will protect the stability of the LPP or FLIS. The amount of throttling that is needed depends on the traffic load on the other processors in the LPP or FLIS.

Table 25 Datafilling table IPTHRON

Field	Subfield or refinement	Entry	Explanation and action
LMSNODE		0 to 750	Local message switch node. Enter the node name and number of the EIU (from table LIUINV).
TXCAPCT		0 to 32767	Transmit capacity. Enter the number in kilobytes per second of total IP transmit capacity from the EIU to other SuperNode hosts across the LMS.
RXCAPCT		0 to 32767	Receive capacity. Enter the number in kilobytes per second of total IP receive capacity from other SuperNode hosts across the LMS to the EIU.
OPTION		$	Option. Enter $.

IPTHRON example

The following figure shows example datafill.

Figure 113 MAP display example for table IPTHRON

```
LMSNODE      TXCAPCT    RXCAPCT    OPTION
-----------------------------------------------------------
EIU 0        40         40           $
EIU 1        40         40           $
```

Table interdependency and autoconfiguration

Tables IPROUTER and IPHOST depend on table IPNETWRK. Changes made to the IP address datafill in table IPNETWRK will reconfigure the IP addresses of nodes listed in tables IPROUTER and IPHOST.

The autoconfiguration routine validates changes made to table IPNETWRK against the current entries in table IPROUTER and IPHOST. If the new parameters of table IPNETWRK do not conform with all current entries in the other tables, autoconfiguration is not performed and the reason is indicated.

The autoconfiguration routine also validates the host IDs of all nodes in tables IPROUTER and IPHOST with the new host ID size in table IPNETWRK.

Table change restrictions

The following restrictions apply to changes made to tables IPROUTER and IPNETWRK:

- The router should be in an offline state before changes are made to tables IPROUTER or IPNETWRK.
- All IP-capable nodes should be in an offline state before changes are made to table IPNETWRK.

XA-Core Ethernet interface engineering and provisioning

On XA-Core switches, operating companies have the option of using XA-Core Ethernet interface cards such as HIOP (NTLX04CA) and HCMIC (NTLX17AA), rather than EIUs, for CM access to the Ethernet LAN. These cards are provisioned in the core, and they provide a direct path between the core and the Ethernet LAN.

A primary reason to use the XA-Core Ethernet interface is that it has much higher messaging capacity than EIUs have. Additionally, the XA-Core Ethernet interface has advantages over EIUs in fault handling and recovery.

The XA-Core Ethernet interface is used primarily in Succession solutions; however, the interface is also supported on XA-Core TDM switches. OSSAIN does not restrict which XA-Core Ethernet interface cards can be used, but the product platform on which TOPS runs may have restrictions on which cards are supported. Contact your Nortel Networks representative for more information.

A maximum of eight XA-Core Ethernet interface cards can be provisioned in an XA-Core, and a minimum of two (if any) should be provisioned for redundancy.

XA-Core Ethernet interface does not use the CM IP address that is used with EIUs. Instead, two IP addresses are assigned to the XA-Core and datafilled in table CMIPADDR. These addresses are dynamically associated with in-service Ethernet interface cards. As long as at least two interface cards are in service, each of the two addresses will always be associated with a different card. If only one interface card is in service, it is associated with both addresses. If more than two Ethernet interface cards are provisioned, only two are active at any time. The others are in hot standby mode.

Upcoming sections discuss the following topics related to use of XA-Core Ethernet interface cards in the OSSAIN network:

- IP address restrictions
- routing restrictions
- capacity restrictions
- security restrictions
- datafilling XA-Core Ethernet interfaces

Note: For more information about engineering XA-Core Ethernet interfaces, especially in Succession networks, refer to the Engineering Guidelines / System Engineering Bulletins (SEB) section of the Succession Networks Solutions collection in Helmsman.

IP address restrictions

Use of XA-Core Ethernet interfaces for switch connectivity to the LAN imposes certain restrictions on the IP address plan for the OSSAIN network.

- Classes A, B, and C address schemes are supported.
- The CM is assigned two active host IP addresses through table control in DMS table CMIPADDR. These addresses are dynamically bound to in-service XA-Core Ethernet interfaces, and are used by nodes in the OSSAIN network when sending messages to the switch.
- Two additional IP addresses are required for each physical interface card. These addresses are not advertised to the network, but are used for maintenance from the core and for internal diagnostics.
- Only one subnet is supported for the core Ethernet interfaces. All active IP, maintenance IP, and card IP addresses must be unique, and they must all have the same netmask and the same subnet.

Note: Refer to “Datafilling XA-Core Ethernet interfaces” on page 267 for details on table CMIPADDR.

Routing restrictions

The HIOP and HCMIC cards are not routers. A gateway router (or a pair of routers, for redundancy) is required on the XA-Core LAN if OSSAIN messaging with nodes on other subnets is to occur. The IP address of the gateway router is datafilled in table CMIPADDR. VRRP and similar redundancy schemes allow a pair of routers to use a single IP address.

Capacity restrictions

Please refer to page 272 for information about determining OSSAIN messaging bandwidth requirements. Exact capacity numbers for the HIOP and HCMIC cards were not available at this time of this writing, but both cards have much higher messaging capacity than EIUs have. It is not anticipated that messaging capacity issues will arise on a TDM switch that uses XA-Core Ethernet interfaces for OSSAIN, even if the interfaces are also shared with the Core and Billing Manager (CBM) application.

- In TOPS CS2000 Succession switches in which OSSAIN is sharing the Ethernet interface cards with high-runner Succession calls, OSSAIN decreases the capacity of the CS2000 for non-TOPS calls. CS2000 capacity numbers published by Nortel Networks engineering groups do not apply in this case, unless the office model has specifically included OSSAIN calls.
- The XA-Core does not provide a way to reserve portions of its total Ethernet interface bandwidth for specific applications. All applications compete for the same bandwidth.
- OSSAIN requires that each OSSAIN node (SN or OSAC switch) use only one of the two active host CM IP addresses. That address is datafilled against the node in table OANODINV.
- The CM IP addresses should be datafilled against OSSAIN nodes in a way that balances the messaging load across the physical interfaces.

Security restrictions

Unlike EIUs, XA-Core Ethernet interface cards do not provide screening capabilities. (EIUs can be configured to protect the core by screening out incoming messages from unwanted IP addresses, including addresses that do not match datafilled nodes and addresses of nodes that are off-line or have been declared babblers.)

Since call processing can be adversely affected if the core is flooded with unwanted messages, it is especially important that the network be properly configured with firewalls and router filters when XA-Core Ethernet interfaces are used.

Refer to “Data port numbers” on page 592 for information about the ports used by OSSAIN.

Datafilling XA-Core Ethernet interfaces

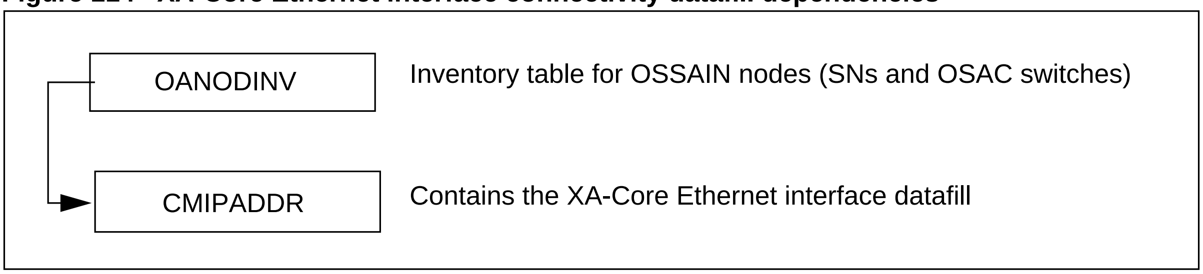
All information about XA-Core Ethernet interfaces is datafilled in a single table, as shown below.

Table 26 Datafill for XA-Core Ethernet interfaces

Table name	Explanation
CMIPADDR	The Computing Module Internet Protocol Addresses table contains the IP addresses and netmasks associated with the XA-Core, as well as the gateway router IP address and the edge device IP addresses and netmasks.

For OSSAIN nodes that exchange messages with the switch using its XA-Core Ethernet interfaces, one of the CM host active IP addresses from table CMIPADDR must be datafilled against the node in table OANODINV. The IP address must first be datafilled in table CMIPADDR before it can be datafilled in table OANODINV. The following table, in which the arrow indicates a needs relationship, illustrates this dependency.

Figure 114 XA-Core Ethernet interface connectivity datafill dependencies



Note: Refer to Chapter 7: “OSSAIN data schema” for details on table OANODINV.

The following subsection provides information on how to datafill table CMIPADDR.

CMIPADDR

Table CMIPADDR specifies the IP addresses and netmasks associated with the XA-Core, as well as the gateway router IP address and the edge device IP addresses and netmasks. The information in table CMIPADDR is independent of the datafill for IP connectivity via EIU.

Note: Table CMIPADDR is datafilled during commissioning of the Ethernet interfaces in an XA-Core switch. Changing the gateway or host address information after that may cause data mismatches between various nodes in the IP network.

Table 27 Datafilling table CMIPADDR

Field	Subfield or refinement	Entry	Explanation and action
KEY		See subfields	
	KEYCLASS	GATEWAY, CMHOST, AMDIHOST, ETHRLNK	Key class. Indicates the class of tuple being datafilled. Enter GATEWAY to datafill the IP address of a gateway router. There can be only one GATEWAY tuple. Enter CMHOST to datafill an active IP address for the XA-Core. There can be two CMHOST tuples (total two active host IP addresses). Enter ETHRLNK to datafill a physical Ethernet link. There can be up to 8 ETHRLNK tuples, with each tuple corresponding to a physical link. AMDIHOST was added to allow datafill of an IP address to be used with an AMDI link for CLIP (Classical IP) over ATM. This value cannot be datafilled in SN07 or ISN07.
	ENTRYNO	0 to 7	Entry number. This is used to create a unique key. The KEYCLASS limits the ENTRYNO values that can be datafilled: GATEWAY: 0 to 0 CMHOST: 0 to 1 MDIHOST: N/A ETHRLNK: 0 to 7
DATA		See subfields	This field consists of selector field SELCLASS and refinements.
	SELCLASS	GW, HOST, AMDI, ETHR	Selector class. The KEYCLASS in the key must match the SELCLASS entry as follows: GATEWAY: GW CMHOST: HOST AMDIHOST: AMDI ETHRLNK: ETHR

SELCLASS = GW

If the entry in subfield SELCLASS is GW, datafill refinements as shown in the following table.

Table 1 Datafilling table CMIPADDR GW refinement fields

Field	Subfield or refinement	Entry	Explanation and action
GTWYIP		four numbers, each 0 to 255	Gateway IP address. Enter an address for a gateway router. This address is downloaded to the Ethernet interface card or packlet. ***Note 1:*** VRRP and similar redundancy schemes allow a pair of routers to use a single IP address. ***Note 2:*** The GTWYIP entry may be the same as an address entered for an edge device in an ETHRLINK tuple, if that is appropriate for the network configuration. It may not be the same as any other IP address in table CMIPADDR.
NETID		0 to 1	Network (i.e., subnet) identifier. Enter 0. In SN07 there can be only one Ethernet-based subnet for the core.

SELCLASS = HOST

If the entry in subfield SELCLASS is HOST, datafill refinements as shown in the following table.

Table 2 Datafilling table CMIPADDR HOST refinement fields

Field	Subfield or refinement	Entry	Explanation and action
ACTIVEIP		four numbers, each 0 to 255	Active IP address for the XA-Core. This will be one of the IP addresses used by applications and advertised outside of the core. The active IP addresses are dynamically associated with physical interfaces. If an active IP address is bound to a physical interface that suffers a failure, it will be automatically rebound to a different interface in the same subnet.
NETMASK		1 to 30	Netmask. Enter the number of leading ‘1’s in the mask.
NETID		0 to 1	Network (i.e., subnet) identifier. Enter 0. In SN07 there can be only one Ethernet-based subnet for the core.

SELCLASS = ETHR

If the entry in subfield SELCLASS is ETHR, datafill refinements as shown in the following table.

Table 3 Datafilling table CMIPADDR ETHR refinement fields

Field	Subfield or refinement	Entry	Explanation and action
ACTIVEIP		four numbers, each 0 to 255	Active IP address for the XA-Core. This will be one of the IP addresses used by applications and advertised outside of the core. The active IP addresses are dynamically associated with physical interfaces. If an active IP address is bound to a physical interface that suffers a failure, it will be automatically rebound to a different interface in the same subnet.
SLOT		0 to 18	Slot. The physical slot number on the XA-Core shelf where the Ethernet interface card resides. (For EIOP, this is where the IOP resides.) ***Note:*** Refer to XA-Core engineering documentation for packfill restrictions.
SIDE		UNKNOWN, FRONT, REAR	Side. The location on the XA-Core shelf where the Ethernet interface card resides. Enter FRONT or REAR. ***Note:*** Refer to XA-Core engineering documentation for packfill restrictions.
PACKLET		NONE, LOWER, UPPER	Packlet. For EIOP, enter the location of the Ethernet packlet on the IOP card. For other Ethernet interfaces such as HIOP and HCMIC, enter NONE.
CARDIP		four numbers, each 0 to 255	Card IP address. This address is used only within the interface card or packlet, and is not advertised.
CARDNETM		1 to 30	Card netmask. Enter the number of leading ‘1’s in the mask.
MTCIP		four numbers, each 0 to 255	Maintenance IP address. This is used by the core for maintenance of the interface card or packlet from the core. This address is not advertised.
MTCNETM		1 to 30	Maintenance netmask. Enter the number of leading ‘1’s in the mask.

Table 3 Datafilling table CMIPADDR ETHR refinement fields

Field	Subfield or refinement	Entry	Explanation and action
EDGEIP		four numbers, each 0 to 255	Edge device IP address. This is the address of the device to which the Ethernet interface is directly attached. This address is used by the core in performing integrity checks, using ICMP echo requests, on the point-to-point link between the Ethernet interface card or packlet and the edge device to which it is attached. ***Note:*** In SN07 and higher, OAP Node Connectivity Test messaging replaces ICMP echo in an OSAC remote in the RTS sequence for an OSN, if the OSN is at OAP 9 or higher and the OSAC host is at SN07 or higher. Assuming the edge device is a Passport 8600, datafill the physical IP address of the CallP VLAN/subnet on the Passport 8600 that the HIOP or HCMIC subtends.
NETID		0 to 1	Network (i.e., subnet) identifier. Enter 0. In SN07 there can be only one Ethernet-based subnet for the core.

CMIPADDR example

The following figure shows example datafill.

Figure 115 MAP display example for table IPTHRON

```
KEY       DATA
----------------------------------------------------------
GATEWAY 0  GW (172 16 0 1) 0
CMHOST  0  HOST (172 16 14 108) 20 0
CMHOST  1  HOST (172 16 14 109) 20 0
ETHRLNK 1  ETHR 12 REAR NONE (172 16 14 104) 20 (172 16 14 110) 20
           (172 16 0 2) 0
ETHRLNK 2  ETHR 6 REAR NONE (172 16 14 106) 20 (172 16 14 112) 20
           (172 16 0 3) 0
```

Determining OSSAIN messaging bandwidth requirements

This section discusses messaging bandwidth requirements for OAP messaging and OSAC messaging.

OAP messaging

Logic in the OSSAIN SNs controls OSSAIN call flows. Therefore, SN providers need to supply information about the application-level messaging requirements for their services. Required information includes the number and sizes of the messages in each direction for the high-runner call flows.

On OSAC calls, all of the call-related OAP messaging is between the SN and the OSAC remote. OSAC hosts do not exchange call-related OAP messages with SNs on OSAC calls.

The OAP message size does not account for all of the bytes that are transmitted to or from the core when an OAP message is sent or received. Lower-layer headers, which encapsulate the OAP messages, must also be taken into account. The following table provides information about the header sizes. This information applies to both inbound and outbound OAP messages.

Table 4 Lower-layer header sizes

Core LAN connectivity	Total header size per message, encapsulating each OAP message	Explanation
EIU router	38 bytes	8 bytes UDP header 20 bytes IP header 4 bytes SNAP header 6 bytes FTS header
XA-Core Ethernet interface	28 bytes	8 bytes UDP header 20 bytes IP header

To compute the total bandwidth needed for OAP messaging, the header size shown in the table must be added to the size of each OAP message.

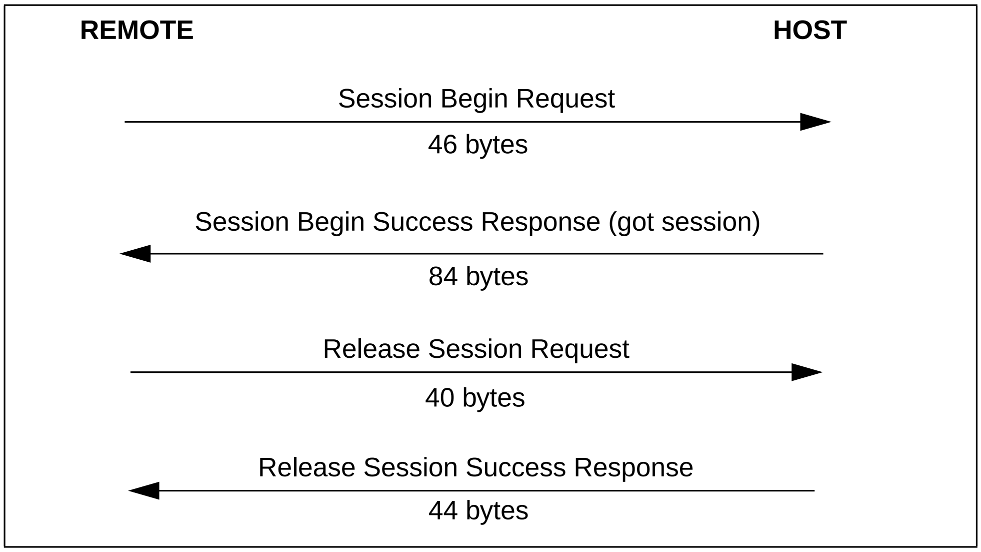
Once the per-call OAP messaging requirements are known, the operating company can derive total OAP messaging bandwidth requirements based on peak traffic levels.

OSAC messaging

When OSAC is used, OSAC messaging as well as OAP messaging needs to be taken into account. Recall that switches use the licensed OAP protocol to communicate with SNs, and they use the proprietary OSAC protocol to communicate with other switches in the OSAC network. SN vendors are not typically able to provide information about the OSAC messaging.

A centralized OSAC session that is immediately available (no queuing for the session) typically involves the OSAC messaging shown in the table below. The message sizes do not include the lower-layer headers that encapsulate each message.

Table 28 Per-call OSAC messaging



If the call has to queue in the host for a session, replace the 84-byte Session Begin Success Response with two messages, sizes 50 and 62 bytes.

There is an additional OSAC message exchange each time an SN requests a voice connection after the call has already been presented to the SN, or releases the voice connection while the call is still at the SN. These messages are normally not larger than 48 bytes in each direction. These messages are not used if datafill indicates that a voice connection should be automatically set up when the call is presented to the SN, and the voice connection is taken down only when the call leaves the SN.

Keep in mind that one switch call might require multiple OSSAIN sessions. For example, separate sessions might be used for front-end and back-end automation. The OSAC messaging shown in the table above is per session, not per call.

The OSAC message size does not account for all of the bytes that are transmitted to or from the core when an OSAC message is sent or received. Lower-layer headers, which encapsulate the OSAC messages, must also be taken into account. Lower-layer header sizes are the same for OSAC messages as for OAP messages. See Table 4, “Lower-layer header sizes,” on page 272.

OSAC and OAP audit messages do not need to be included in bandwidth calculations, as audits are sent only if no payload message has been recently received from the node or session pool.

Part 5: Provisioning

Part 5: Provisioning includes the following chapters:

Chapter 7: “OSSAIN data schema,” beginning on page 277.

Chapter 8: “OSSAIN software optionality control,” beginning on page 363.

Chapter 7: OSSAIN data schema

This chapter provides detailed information on datafilling the tables required by OSSAIN. It discusses each table, its fields and values, and gives example datafill. These tables include translations, node inventories, and parameters.

OSSAIN datafill descriptions

The datafill descriptions and examples in this chapter are organized around the following OSSAIN processing requirements:

- OSSAIN base datafill (page 279)
- centralized OSSAIN (OSAC) parallel datafill (page 315)
- OSSAIN voice connectivity datafill (page 316)
- direct transfers and new subscriber originations datafill (page 325)
- function blocking datafill (page 329)
- trigger datafill (page 332)
- OSSAIN preprocessing of TOPS calls datafill (page 350)
- OSSAIN alternate routing datafill (page 352)
- OSSAIN QMS MIS application datafill (page 355)
- OAINPARM (OSSAIN parameter) datafill (page 357)
- TOPSPARM (TOPS parameter) datafill (page 361)

Note 1: The TOPS QMS criteria tables that need OSSAIN datafill are discussed, but not fully described in this document. Please refer to the *Customer Data Schema Reference Manual* for complete information.

Note 2: Refer to Chapter 6: “OSSAIN engineering,” for information on datafilling the Internet Protocol (IP) tables and datafilling parameters in tables OFCENG (Office Engineering) or OFCAUT (Office Autoprovisioning).

Alphabetical reference for OSSAIN table descriptions

The following table lists each OSSAIN table in alphabetical order and the page in this chapter where its description begins.

Table 29 Alphabetical reference for OSSAIN table descriptions

Table name	Page number
OACAUPRF	page 336
OACNNPRF	page 338
OACTLDEF	page 326
OADSCPRF	page 341
OADTFPRF	page 334
OAFNDISP	page 310
OAFUNBLK	page 330
OAFUNDEF	page 301
OAINCTLA	page 329
OAINPARM	page 357
OAINPRE	page 351
OAINRTE	page 354
OANODINV	page 284
OANODNAM	page 282
OASESNPL	page 295
OATLKPRF	page 344
OATPRFIX	page 348
OAVLMAP	page 318
OQCQPROF	page 292
OSCVLGRP	page 322
QMSMIS	page 355
SNVLGRP	page 323

OSSAIN base datafill

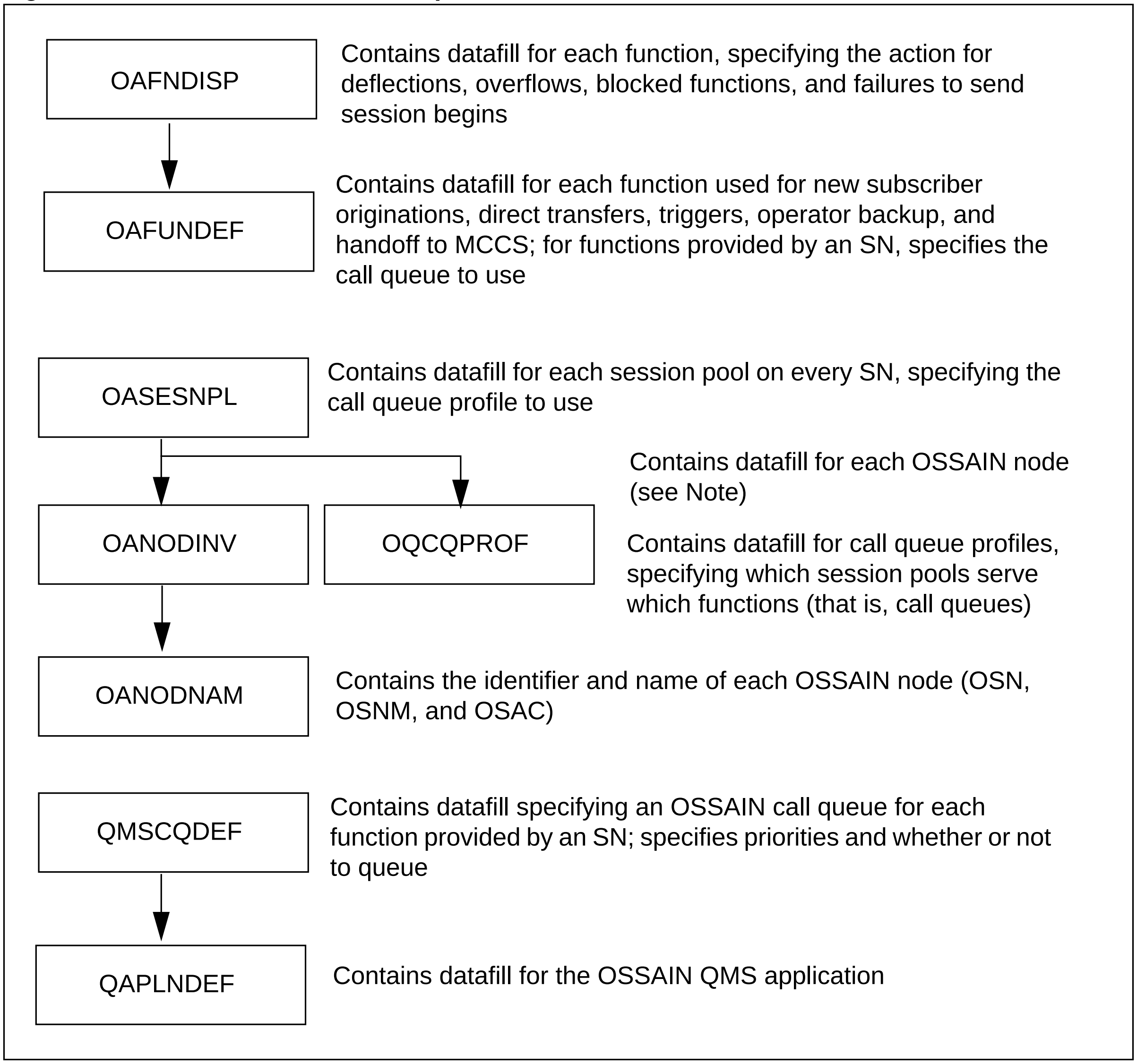
The translations tables for OSSAIN base processing are described in the following table. The tables are listed in the order in which they are to be datafilled.

Table 30 Datafill sequence for OSSAIN base processing

Table name	Explanation
QAPLNDEF	The QMS Application Definition table defines the OSSAIN application.
QMSCQDEF	The QMS Call Queue Definition table defines the OSSAIN call queues.
OANODNAM	The OSSAIN Node Name table contains the identifiers and names of OSSAIN nodes. (See Note)
OANODINV	The OSSAIN Node Inventory table provides OSSAIN node information. (See Note)
OQCQPROF	The OSSAIN QMS Call Queue Profile table defines OSSAIN call queues.
OASESNPL	The OSSAIN Session Pool table defines all OSSAIN session pools. (See Note)
OAFUNDEF	The OSSAIN Function Definition table defines all OSSAIN functions. (See Note)
OAFNDISP	The OSSAIN Function Disposition table identifies actions for deflections, overflows, blocking, and origination failures.
Note: In the OSAC configuration, these tables require parallel datafill. For these tables, examples in this chapter show switch datafill for standalone as well as for the OSAC host and OSAC remote.	

The following figure shows table dependencies for the eight tables that require datafill for minimum OSSAIN processing.

Figure 116 OSSAIN base datafill dependencies



Note: Table ENTYPES and table ENSITES must be datafilled before datafilling table OANODINV.

QAPLNDEF

The OSSAIN application is defined in table QAPLNDEF. For complete information on table QAPLNDEF, please refer to the *Customer Data Schema Reference Manual*.

QAPLNDEF example

The following figure shows example datafill. In the example, the OSSAIN application has its own tuple.

Figure 117 MAP display example for table QAPLNDEF

APLNAME	CALLQS	CQELEMS	AGENTQS	NUMAGNTS	CTSEARCH
TOPS	255	3000	127	8191	0
OSSAIN	220	5000	100	1500	0

QMSCQDEF

OSSAIN call queues are defined in table QMSCQDEF. This table has a two part key: application name and call queue number. OSSAIN call queues are independent of existing operator call queues. For complete information on table QMSCQDEF, please refer to the *Customer Data Schema Reference Manual*.

QMSCQDEF example

The following figure shows example datafill. In the example, call queues OSSAIN 0 and OSSAIN 1 are set up to disallow queuing of calls, by setting CQMAXSIZ = 0. Call queues OSSAIN 2 and OSSAIN 3 are set up to provide queuing for functions associated with these call queues.

Figure 118 MAP display example for table QMSCQDEF

APPLNCQ	CQPRIO	CQAGS	CQMAXSIZ	DEFLAREA	PRAQAREA
TOPS 0	20	18	300	Y 600 5	Y 4 100
TOPS 1	21	10	200	N	Y 4 0
TOPS 2	35	18	300	Y 600 5	Y 3 100
TOPS 3	36	10	240	Y 600 5	N
OSSAIN 0	0	18	0	N	N
OSSAIN 1	1	10	0	N	N
OSSAIN 2	0	18	300	Y 1200 5	N
OSSAIN 3	2	10	240	Y 1500 5	N

OANODNAM

Table OANODNAM contains the identifier (ID) and name of all nodes in the OSSAIN network (including standalone and OSAC). The value for node ID must be unique throughout the OSSAIN network.

In an OSAC network, table OANODNAM requires parallel datafill between the OSAC host switch and the OSAC remote switch.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 31 Datafilling table OANODNAM

Field	Subfield or refinement	Entry	Explanation and action
NODEID		0 to 767	Node identifier. Enter the OSSAIN identifier for the node. The node identifier must be unique throughout the OSSAIN network. ***Note:*** To conserve store, avoid arbitrarily using high numbers for node identifiers.
NODENAME		Alphanumeric up to 12 characters	Node name. Enter the name of the node.

OANODNAM example—standalone switch

The following figure shows example datafill. In the example, a standalone switch and three SNs are datafilled.

Note: While assigning a node ID to a standalone switch is not required, it is recommended. If the ID is not datafilled, a default value of FFFF is used in OAP messaging.

Figure 119 MAP display example for table OANODNAM (standalone)

```
NODEID       NODENAME
-------------------------------------------------------------------
11           SWITCH_1
12           BRAND_01
13           DEBIT_01
14           AABS_01
```

OANODNAM example—OSAC host switch

The following figure shows example datafill that is parallel. In the example, one OSAC host switch (SWITCH_A), one OSAC remote switch (SWITCH_B), and three SNs are datafilled.

Figure 120 MAP display example for table OANODNAM (host)

```
NODEID      NODENAME
---------------------------------------------------------------------
1           SWITCH_A
2           SWITCH_B
100         BRAND_01
101         DEBIT_01
102         AABS_01
```

OANODNAM example—OSAC remote switch

The following figure shows example datafill that is parallel. In the example, the same OSAC host, OSAC remote, and SNs are datafilled.

Figure 121 MAP display example for table OANODNAM (remote)

```
NODEID      NODENAME
---------------------------------------------------------------------
1           SWITCH_A
2           SWITCH_B
100         BRAND_01
101         DEBIT_01
102         AABS_01
```

OANODNAM error messages

A standard table control error message is displayed when the user tries to datafill a node ID or node name that is not unique.

OANODINV

Table OANODINV contains information on all nodes in the OSSAIN network. The node types OSNM, OSN, and OSAC are datafilled in table OANODINV, along with other information specific to the type of node.

In an OSAC network, table OANODINV requires parallel datafill between the OSAC host switch and the OSAC remote switch.

OSSAIN node type refinements

Refer to the following list for refinement datafill in table OANODINV that applies to each node type (NODEAREA field):

- For OSNM nodes, the following refinements apply:
  - ENNODENO
  - PROTOCOL
  - ADDRTYPE
  - IPV4ADDR
  - NMTCPORT
  - DATAPATH
  - CMIPADDR
  - FDETECT
  - FTHRESH
  - RTSTIME
  - TSTTIME
  - ONODTYPE
  - ONODLOC
  - ONODSITE
  - ONODINFO
  - AUDTRIES
  - AUDRTIMR
  - AUDFREQ
- For OSN nodes, the following refinements apply:
  - ENNODENO
  - PROTOCOL
  - ADDRTYPE
  - IPV4ADDR
  - DATAPATH
  - CMIPADDR

- HOSTNAME
- FDETECT
- FTHRESH
- ONODTYPE
- ONODLOC
- ONODSITE
- ONODINFO
- AUDTRIES
- AUDRTIMR
- AUDFREQ

• For OSAC nodes, the following refinements apply:

- RELATION
- If RELATION is set to OTHER, the following additional refinements apply:
  - ENNODENO
  - PROTOCOL
  - ADDRTYPE
  - IPV4ADDR
  - DATAPATH
  - CMIPADDR
  - ONODTYPE
  - ONODLOC
  - ONODSITE
  - ONODINFO
  - AUDTRIES
  - AUDRTIMR
  - AUDFREQ
- If RELATION is set to SELF, the following additional refinements apply:
  - AUDTRIES
  - AUDRTIMR
  - AUDFREQ

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 32 Datafilling table OANODINV

Field	Subfield or refinement	Entry	Explanation and action
NODENAME		Alphanumeric up to 12 characters	Node name. Enter the name of the node (from table OANODNAM).
NODEAREA		See subfields	Node information. This field consists of subfield ONPMTYPE and its refinements. Refer to the previous list of refinements that apply to the node type being datafilled.
	ONPMTYPE	OSN, OSNM, OSAC	OSSAIN node peripheral module (PM) type. Enter the PM type of the node. ***Note:*** MIS nodes must be datafilled with the OSNM type.
	RELATION	SELF, OTHER	Relation. Enter the relationship that the OSAC node being datafilled has to the switch. SELF specifies the OSAC node itself. OTHER specifies all other OSAC nodes. ***Note:*** Only one OSAC of type SELF can be datafilled at a given switch. An OSAC of SELF must be datafilled before an OSAC of OTHER.
	ENNODENO	0 to 255	External node number. Enter the number of the node.
	PROTOCOL	UDP	Protocol. Enter the protocol name for transport layer messaging. UDP is the only valid entry for this field. This field has the refinement ADDRTYP.
	ADDRTYP	IPV4	Internet protocol address type. Enter the type of IP address. IPV4 is the only valid entry for this field. This field has the refinement IPV4ADDR.
	IPV4ADDR	0 to 255 for each of the 4 address parts	IP version 4 address. This field identifies the 32-bit IP address of the node being datafilled. Values are expressed in the form x x x x separated by spaces. Enter the IP version 4 address. ***Note:*** For more information on IP addressing, refer to Chapter 6: “OSSAIN engineering.”

Table 32 Datafilling table OANODINV

Field	Subfield or refinement	Entry	Explanation and action
	DATAPATH	EIU, XAETHR	Data path. If the switch is to use an EIU router for messaging with the node, enter EIU. If the switch is to use XA-Core Ethernet interface cards for messaging with the node, enter XAETHR and also datafill the CMIPADDR refinement.
	CMIPADDR	0 to 255 for each of the 4 address parts	Computing Module IP Address. For nodes with the data path set to XAETHR, enter the CM IP address (from a CMHOST tuple in table CMIPADDR). ***Note:*** If the switch uses XA-Core Ethernet interface for messaging with multiple nodes, some nodes should be datafilled to use the IP address from one of the CMHOST tuples and other nodes should be datafilled to use the IP address from the other CMHOST tuple. The objective is to distribute the messaging evenly over the two IP addresses.
	NMTCPORT	1024 to 32767	Node maintenance port. For OSNM node types, enter the port number to be used when sending a node class message to the node being datafilled.
	HOSTNAME	Alphanumeric up to 12 characters	Host name. For OSN node types, enter the name of the host. The host name must have been datafilled in table OANODINV as an OSAC node type with a RELATION of OTHER.
	FDETECT	Y, N	Failed message detection. Enter Y or N to indicate whether or not failed message detection is enabled for OSN or OSNM node types. If set to Y, also datafill the FTHRESH refinement.

Table 32 Datafilling table OANODINV

Field	Subfield or refinement	Entry	Explanation and action
	FTHRESH	0 to 100	Failed message threshold. Enter the percent of messages originated from an external node that can be unsuccessfully processed by switch data communications software before the DMS switch declares the node a babbler and places it in a system busy state. For this purpose, an incoming message is considered unsuccessfully processed if it is obviously corrupted and cannot be read, if it contains incomplete or invalid routing information in the OAP message header, or if it causes an OAP protocol violation. ***Note:*** For OSNM nodes, the switch system busies the node. For OSN nodes, the switch (OSAC host) system busies the connection to the node.
	RTSTIME	20 to 600	Return to service time. For OSNM nodes, enter the number of seconds that the switch waits for a response to an RTS message to the node or one of its session pools. If the amount of time expires, the RTS attempt fails.
	TSTTIME	20 to 600	Test time. For OSNM nodes, enter the number of seconds that the switch waits for a response to a TST message to the node or one of its session pools. If the amount of time expires, the TST attempt fails.
	ONODTYPE	Alphanumeric up to 12 characters	OSSAIN node type. Enter the descriptive name for the type of node being datafilled (from table ENTYPES).
	ONODLOC	See subfields	OSSAIN node location. This refinement consists of FLOOR, ROW, and POSITION.
	FLOOR	0 to 99	Floor. Enter the floor number where the node is located.
	ROW	Alphanumeric up to 2 characters	Row. Enter the row where the node is located.
	POSITION	0 to 99	Position. Enter the position number where the node is located.
	ONODSITE	Alphanumeric up to 12 characters	OSSAIN node site. Enter the descriptive site for the type of node being datafilled (from table ENSITES).

Table 32 Datafilling table OANODINV

Field	Subfield or refinement	Entry	Explanation and action
	ONODINFO	Alphanumeric up to 20 characters	OSSAIN node information. Enter descriptive information for the type of node being datafilled. This information is displayed on the node maintenance MAP when the node is posted.
AUDTRIES		2 to 255	Audit tries. Enter the number of times the switch sends an audit without receiving a response (in other words, the number of consecutive audit failures). Once this limit is met, the switch takes the node or session pool out of service.
AUDRTIMR		1 to 600	Audit response timer. Enter the number of seconds the switch waits for a response to an audit from the node or one of its session pools.
AUDFREQ		1 to 600	Audit frequency. Enter the minimum number of seconds that elapse between node or session pool audits. ***Note:*** If the value entered is less than 60, it controls only the frequency of node audits. Successful session pool audits never occur more often than every 60 seconds. If the value entered is greater than 60, it controls the frequency of both node audits and session pool audits.

Note: The specific values of fields AUDTRIES, AUDTIMR, and AUDFREQ depend upon the applications in use. For more information, please refer to the *Billing and Access Services Administration Guide*, 203-3261-301, or the *Billing and Access Services Planning and Engineering Guide*, 203-3261-101

OANODINV example—standalone switch

The following figure shows example datafill. In the example, the standalone switch (SWITCH_1) is datafilled as an OSAC node type. This switch provides call processing and maintenance for two SNs having the OSNM node type.

Note: While assigning a node ID (in table OANODNAM) is not required for a standalone switch, it is recommended. When datafilling the standalone switch in table OANODINV, it has the OSAC node type with a relation of SELF.

Figure 122 MAP display example for table OANODINV (standalone)

```
NODENAME   NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_1  OSAC SELF    4        6         15
BRAND_01 OSNM 1 UDP IPV4 47  245 1 34 7000 EIU Y 2 240 60 SN 4 BB 3 CITYA BRANDING1
    4 6 20
AABS_01  OSNM 2 UDP IPV4 47  245 1 35 7002 EIU Y 2 240 60 SN 1 DD 33 CITYA AABS1
    4 6 20
```

OANODINV example—OSAC host switch

The following figure shows example datafill that is parallel. In the example, the OSAC host switch (SWITCH_A) maintains three SNs having the OSNM node type. These SNs are centralized at the OSAC host switch so that the OSAC remote switch (SWITCH_B) can use the functions provided by them.

Figure 123 MAP display example for table OANODINV (host)

```
NODENAME    NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_A   OSAC SELF    4        6        20
SWITCH_B   OSAC OTHER  0 UDP IPV4 47 245 2 1 XAETHR 47 245 1 54 SWITCH
    2 B 11 CITYB REMOTEOSAC 4 6 20
BRAND_01  OSNM 0 UDP IPV4 47 245 3 34 7000 XAETHR 47 245 1 54 Y 2 240 60 SN
    4 BB 3 CITYA BRANDING1 4 6 20
DEBIT_01  OSNM 1 UDP IPV4 47 245 3 35 7001 XAETHR 47 245 1 44 Y 2 240 30 SN
    1 A 01 CITYA DEBIT1 4 6 20
AABS_01   OSNM 2 UDP IPV4 47 245 3 36 7002 XAETHR 47 245 1 44 Y 2 240 60 SN
    1 DD 33 CITYA AABS1 4 6 20
```

OANODINV example—OSAC remote switch

The following figure shows example datafill that is parallel. In the example, the OSAC remote (SWITCH_B) uses the functions provided by three SNs having the OSN node type. These SNs are centralized at the OSAC host switch (SWITCH_A).

Figure 124 MAP display example for table OANODINV (remote)

```
NODENAME    NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_A   OSAC OTHER 0 UDP IPV4 47 245 1 54 EIU SWITCH 4 C 12 CITYA HOSTOSAC 4 6 20
SWITCH_B   OSAC SELF     4        6        20
BRAND_01   OSN 0 UDP IPV4 47 245 3 34 EIU SWITCH_A Y 2 SN 4 BB 3 CITYA
    BRANDING1 4 6 20
DEBIT_01   OSN 1 UDP IPV4 47 245 3 35 EIU SWITCH_A Y 2 SN 1 A 01 CITYA DEBIT1 4 6 20
AABS_01    OSN 2 UDP IPV4 47 245 3 36 EIU SWITCH_A Y 2 SN 1 A 33 CITYA AABS1 4 6 20
```

OANODINV error messages

The following table lists possible error messages.

Table 33 Error messages for table OANODINV

Error message	Explanation
`The specified ONPMTYPE and ENNODENO combination is currently in use by another node. The ONPMTYPE and ENNODENO combination must be unique in table OANODINV.`	The combination of ONPMTYPE and ENNODENO values are not unique.
`The external node PM type, ONPMTYPE, can not be modified. This node must be deleted and reentered if an ONPMTYPE change is desired.`	The user tries to change the ONPMTYPE value.
`OSSAIN node names must be unique.`	The NODENAME value is not unique.
`The specified IP address is currently in use by another node. IP addresses must be unique.`	The IPV4ADDR address is not unique.
`This node name is in use in table XXXXXXXX.` `You must remove all references to this node name before you can delete it.`	The user tries to delete a tuple in table OANODINV that has a node name used in one of the following tables: - OASESNPL - QMSMIS - OAVLMAP - OSCVLGRP - SNVLGRP XXXXXXXX is the name of the table using the node name.
`You have not requested that failed message detection be enabled for this node. It is recommended that FDETECT be set to Y to enable failed message detection.`	Failed message detection is not requested (FDETECT value is set to N).
`This node must be offline before its entry in table OANODINV can be modified. The change will not be performed.`	The user tries to change table OANODINV for a node that is not in an offline state.
`This node must be offline before it can be deleted from table OANODINV. The deletion will not be performed.`	The user tries to delete a node from table OANODINV that is not in an offline state.

Table 33 Error messages for table OANODINV

Error message	Explanation
`An OSAC node with a relation of OTHER cannot be datafilled until the OSAC node with a relation of SELF is datafilled.`	The user tries to datafill an OSAC node with a RELATION of OTHER before datafilling an OSAC node with a RELATION of SELF.
`Only one OSAC node with a relation of SELF may be datafilled.`	The user tries to datafill more than one OSAC node with SELF.
`The node name specified in field HOSTNAME has not been datafilled in table OANODINV.`	The user tries to datafill an OSN node with a HOSTNAME that has not been datafilled in table OANODINV as an OSAC node with a RELATION of OTHER.
`Node type in field ONODTYPE has not been datafilled in table ENTYPES.`	The user tries to datafill the ONODTYPE field with a node that has not been datafilled in table ENTYPES.
`Node site in field ONODSITE has not been datafilled in table ENSITES.`	The user tries to datafill the ONODSITE field with a site that has not been datafilled in table ENSITES.
`This OSAC node is referenced by an OSN node. All OSN references must be removed before this tuple can be deleted.`	The user tries to delete an OSAC node in table OANODINV that is used as the host for an OSN node.
`The OSAC SELF tuple may not be deleted until all OSAC OTHER tuples are deleted.`	The user tries to delete an OSAC node with a RELATION of SELF before first deleting any OSAC nodes with a RELATION of OTHER.

OQCQPROF

Table OQCQPROF associates lists of OSSAIN QMS call queues with a call queue profile number. Session pools can then be associated with one of the profile numbers in table OASESNPL.

Call queue profiles may be specified in one of two ways:

- using office-wide priority and office-wide Assignable Grade of Service (AGS) values
- using profile-specific priority and profile-specific AGS values

Call queue profiles datafilled using office-wide priorities contain a list of up to 255 call queues. Each call queue is specified as a value in the range CQ0 to CQ254.

The priority and AGS associated with each call queue specified are the office-wide values defined for the queue in table QMSCQDEF in the QMS Call and Agent Manager (QMS CAM). Office-wide priority call queue profiles are the simplest profiles to datafill, and result in consistent office-wide priorities and AGS values being applied to all session pools.

If, for a particular call queue profile, the operating company wishes to override the system priority and AGS associated with each call queue, the profile may be datafilled using priority and AGS values specific to the profile. For each of up to four priority levels (labeled PRIO3, PRIO2, PRIO1, and PRIO0, with PRIO3 being the highest priority level), a list of up to 32 call queues along with the desired AGS for each may be specified.

Note: Table QAPLNDEF must be datafilled before table OQCQPROF. The AGENTQ and CALLQ values in the tuples in table OQCQPROF are effectively limited by the datafill for the OSSAIN application in table QAPLNDEF.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 34 Datafilling table OQCQPROF

Field	Subfield or refinement	Entry	Explanation and action
CQPROFNM		0 to 254	Call queue profile number. Enter the OSSAIN QMS call queue profile number.
AGENTQ		AQ0 to AQ254	Agent queue. Enter the agent queue used for idle agents with this call queue profile number. ***Note:*** The range of this field is effectively limited by the value datafilled for the OSSAIN application in table QAPLNDEF (AGENTQS field) in the QMS CAM. For example, if AGENTQS is set to 5 for OSSAIN, only agent queues in the range AQ0 to AQ4 should be datafilled in the AGENTQ field.
PRIOAREA		See subfields	Priority area. This field consists of subfields PRIOTYPE and its refinements: CQLIST, PRIO3, PRIO2, PRIO1, and PRIO0.
	PRIOTYPE	OFC, PROF	Priority type. Enter the priority type for session pools with this profile. OFC indicates office-wide call queue priorities. PROF indicates profile-specific priorities.
	CQLIST	CQ0 to CQ254	Call queue list. If PRIOTYPE is set to OFC, datafill this refinement. Enter a value for each call queue included in this profile. End the list with $. No further refinements require datafill.

Table 34 Datafilling table OQCQPROF

Field	Subfield or refinement	Entry	Explanation and action
	PRIO3	See subfields	Priority level 3. If PRIOTYPE is set to PROF, datafill this refinement. This field is the highest priority level and consists of refinements CALLQ and AGS.
	PRIO2	See subfields	Priority level 2. If PRIOTYPE is set to PROF, datafill this refinement. This field is the second highest priority level and consists of refinements CALLQ and AGS.
	PRIO1	See subfields	Priority level 1. If PRIOTYPE is set to PROF, datafill this refinement. This field is the third highest priority level and consists of refinements CALLQ and AGS.
	PRIO0	See subfields	Priority level 0. If PRIOTYPE is set to PROF, datafill this refinement. This field is the lowest priority level and consists of refinements CALLQ and AGS.
	CALLQ	CQ0 to CQ254	Call queue. Enter up to 32 call queues included in the profile. If less than 32 queues are required, end the list with a $. ***Note:*** The range of call queues is effectively limited by the value datafilled for the OSSAIN application in table QAPLNDEF (CALLQS field) in the QMS CAM. For example, if CALLQS is set to 10 for OSSAIN in table QAPLNDEF, only call queues in the range CQ0 to CQ9 should be datafilled in the CALLQ field.
	AGS	10 to 80	Assignable grade of service. Enter the aging factors from 1.0X to 8.0X for the grade of service to be applied to the call queue for this profile.

OQCQPROF example

The following figure shows example datafill.

Figure 125 MAP display example for table OQCQPROF

```
CQPROFNM        AGENTQ       PRIOAREA
---------------------------------------------------------------------
0               AQ0          OFC (CQ0) $
1               AQ1          OFC (CQ1) $
2               AQ2          OFC (CQ0) (CQ1) $
3               AQ3          PROF (CQ0 10) (CQ1 80) (CQ2 10) $
```

OQCQPROF error messages

The following table lists possible error messages.

Table 35 Error messages for table OQCQPROF

Error message	Explanation
`At least one call queue must be specified.`	No call queue is specified when adding a tuple to table OQCQPROF.
`A call queue may not appear in more than one priority level.`	The user tries to datafill the same call queue in more than one priority level.
`A call queue has been specified in the same priority level with different AGS values.`	The user tries to datafill the same call queue in the same priority level with different AGS values.
`This call queue profile is in use in table OASESNPL.`	The user tries to delete a call queue profile from table OQCQPROF that is in use in table OASESNPL.
`UNABLE TO CHANGE CALL QUEUE PROFILE. CHECK QMS LOGS AND SWERRS.`	The user tries to change a tuple in OQCQPROF and an error is encountered when trying to update the call queue profile in the QMS CAM.

OASESNPL

Table OASESNPL defines the attributes of session pools for OSSAIN calls. A session pool is a group of sessions on a particular SN (standalone or centralized) that serve the same function or functions. This table also defines host-remote session pools between OSAC nodes.

In an OSAC network, table OASESNPL requires parallel datafill between the OSAC host switch and the OSAC remote switch.

Session pool origination type refinements

Refer to the following list for refinement datafill in table OASESNPL that applies to each session pool origination type (ORIGAREA field):

- For subscriber originations, the following refinements apply:
  - SWTCHTYP
  - ONHKTMR
  - CALLTMR
  - TRIGEVNT
  - OCQPROF
  - PROTOCOL
  - CPPORT
  - MTCPORT

- For SN originations, the following refinements apply:
  - ONHKTMR
  - CALLTMR
  - PROTOCOL
  - CPPORT
  - MTCPORT
- For OSACORIG and OSACTERM types, no refinements apply.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 36 Datafilling table OASESNPL

Field	Subfield or refinement	Entry	Explanation and action
SESNPLID		0 to 4094	Session pool identifier. Enter the session pool identifier for the session pool. ***Note:*** To conserve store, avoid arbitrarily using high numbers for session pool identifiers.
SESNPLNM		Alphanumeric up to 16 characters	Session pool name. Enter the name of the session pool.
MAXSESN		0 to 1023	Maximum sessions. Enter the maximum number of simultaneous calls that can be handled by the node for this session pool.
NODENAME		Alphanumeric up to 12 characters	Node name. Enter the node name from table OANODINV for this session pool.
ORIGAREA		See subfields	Origination area. This field consists of subfield ORIGTYPE and its refinements.

Table 36 Datafilling table OASESNPL

Field	Subfield or refinement	Entry	Explanation and action
	ORIGTYPE	SUSCRIBER, SN, OSACORIG, OSACTERM	Origination type. This field specifies how the sessions in this session pool are selected. Enter SUBSCRIBER for a session pool that has sessions selected by the switch for communication with an SN. Enter SN for a session pool that has sessions selected by the SN associated with this session pool. This value allows the SN to originate calls. Enter OSACORIG for a session pool that is used to originate a host-remote session with another OSAC node. Enter OSACTERM for a session pool that is used to terminate a host-remote session with another OSAC node.
	SWTCHTYP	S, SH, R	Switch type. If ORIGTYPE is set to SUBSCRIBER, datafill this refinement. It identifies the switch type for this session pool. Enter S for standalone. Enter SH for standalone/host. Enter R for remote.
	ONHKTMR	USEDEFLT, OVERRIDE	On-hook timer. Enter the on-hook sanity timer value for this session pool. It is initiated when the switch detects that a subscriber has gone on-hook. Enter USEDEFLT to use the default on-hook sanity timer value from the OSSAIN parameter DEFAULT_ON_ HOOK_TIMER_DURATION. Enter OVERRIDE to override the default value, and also datafill the TIMERVAL refinement.
	TIMERVAL	1 to 3600	Timer value. Enter the value of the timer in the range of 1 to 3600 seconds.

Table 36 Datafilling table OASESNPL

Field	Subfield or refinement	Entry	Explanation and action
	CALLTMR	USEDEFLT, OVERRIDE	Call timer. Enter the call sanity timer value for this session pool. It is initiated when no subscribers are connected to the call. Enter USEDEFLT to use the default call sanity timer value from the OSSAIN parameter CALL_SANITY_ TIMER_DURATION. Enter OVERRIDE to override the default value, and also datafill the TIMERVAL refinement.
	TIMERVAL	1 to 3600	Timer value. Enter the value of the timer in seconds.
	TRIGEVNT	Y, N	Trigger event inform session pool. If ORIGTYPE is set to SUBSCRIBER, datafill this refinement. It specifies whether or not this session pool is used only for trigger event inform messages. If set to Y, the session pool is used only for trigger event inform messages. No agent data is allocated and no call queue profile is required. Sessions for this session pool are not provided by the QMS CAM. The first available session is selected. The name of this session pool is specified in table OAFUNDEF (TRIGPOOL field). If set to N and SWTCHTYP is set to S or SH, the session pool is used for subscriber originations and it can also be used for trigger event inform messages. Also datafill the OCQPROF refinement.
	OCQPROF	0 to 254	OSSAIN call queue profile. If ORIGTYPE is set to SUBSCRIBER with a SWTCHTYP of S or SH, and TRIGEVNT is set to N, datafill this refinement. Enter the OSSAIN call queue profile index from table OQCQPROF.
	PROTOCOL	UDP	Protocol. Enter the transport protocol used to communicate between the switch and SN or between OSAC switches. UDP is the only valid entry for this field. Also datafill the CPPORT and MTCPORT refinements.
	CPPORT	1024 to 32767	Call processing port. Enter the UDP port number the session pool uses to receive call processing messages.

Table 36 Datafilling table OASESNPL

Field	Subfield or refinement	Entry	Explanation and action
	MTCPORT	1024 to 32767	Maintenance port. Enter the UDP port number the session pool uses to receive maintenance messages. ***Note:*** This field appears only when SWTCHTYP is S or SH.

OASESNPL example—standalone switch

The following figure shows example datafill. In the example, three session pools are datafilled. Two are for subscriber originations and one is for SN originations.

Figure 126 MAP display example for table OASESNPL (standalone)

```
SESNPLID  SESNPLNM  MAXSESN  NODENAME  ORIGAREA
------------------------------------------------------------------------------
4      AABS      15    AABS_01   SUBSCRIBER S USEDEFLT USEDEFLT N 3 UDP 7001 7002
5      DEBIT     10    DEBIT_01  SUBSCRIBER S USEDEFLT USEDFLT N 5 UDP 7001 7002
6       BRANDING    10     BRAND_01  SN USEDEFLT USEDEFLT UDP 7001 7002
```

OASESNPL example—OSAC host switch

The following figure shows example datafill that is parallel. In the example, the host-remote session pool (HOST_2_REM) is datafilled as OSACTERM, because all OSAC requests originate from the remote. Session pool ID 8 (DEBIT) is used only for trigger event informs.

Figure 127 MAP display example for table OASESNPL (host)

```
SESNPLID  SESNPLNM   MAXSESN  NODENAME  ORIGAREA
------------------------------------------------------------------------------
1     HOST_2_REM    20    SWITCH_B  OSACTERM
2    AABS_SN      15   AABS_01   SUBSCRIBER SH USEDEFLT USEDEFLT N 5 UDP 7001 7002
3     BRANDING      10    BRAND_01  SN USEDEFLT USEDEFLT UDP 7001 7002
8    DEBIT        10   DEBIT_01  SUBSCRIBER SH USEDEFLT USEDEFLT Y UDP 7003 7002
```

OASESNPL example—OSAC remote switch

The following figure shows example datafill that is parallel. In the example, the host-remote session pool (REM_2_HOST) is datafilled as OSACORIG.

Figure 128 MAP display example for table OASESNPL (remote)

```
SESNPLID  SESNPLNM   MAXSESN  NODENAME  ORIGAREA
------------------------------------------------------------------------------
1      REM_2_HOST    20     SWITCH_A    OSACORIG
2      AABS_SN      15     AABS_01     SUBSCRIBER R USEDEFLT USEDEFLT N UDP 7001
3      BRANDING     10     BRAND_01    SUBSCRIBER R USEDEFLT USEDFELT N UDP 7001
8      DEBIT        10     DEBIT_01    SUBSCRIBER R USEDEFLT USEDEFLT Y UDP 7003
```

OASESNPL error messages

The following table lists possible error messages.

Table 37 Error messages for table OASESNPL

Error message	Explanation
OSSAIN session pool names must be unique.	The user tries to add two tuples with the same session pool name.
This session pool name is in use in Table OAFUNDEF. You must remove all references to this session pool before you can delete it.	The user tries to delete a tuple from OASESNPL that is used in table OAFUNDEF.
Field ORIGTYPE must be set to OSAC when the node in field NODENAME is of PM type OSAC.	One of the following conditions applies: - The node type in NODENAME is of the OSAC node type and ORIGTYPE is set to either SUBSCRIBER or SN - The node type in NODENAME is of the OSNM node type and ORIGTYPE is set to OSAC
Field SWTCHTYP must be datafilled as R for OSN nodes.	The node type in NODENAME is of the OSN node type and SWTCHTYP is not set to R.
Field SWTCHTYP cannot be datafilled as R for OSNM nodes.	The node type in NODENAME is of the OSNM node type and SWTCHTYP is set to R.
The service node associated with this tuple must first be OFFLINE before this tuple can be changed.	The user tries to change a tuple while the SN associated with it is not in a offline state.
Field ORIGTYPE cannot be changed. You must delete this tuple first and then re-add with the new value.	The user tries to change the ORIGTYPE field.
Field SWTCHTYP cannot be changed. You must delete this tuple first and then re-add with the new value.	The user tries to change the SWTCHTYP field.
Field NODENAME cannot be changed. You must delete this tuple first and then re-add with the new value.	The user tries to change the NODENAME field.

Table 37 Error messages for table OASESNPL

Error message	Explanation
`You must datafill the call queue profile in table OQCQPROF before the index can be datafilled in this table.`	The user tries to datafill the OCQPROF field with a call queue profile that has not been defined in table OQCQPROF.
`Field TRIGEVNT cannot be changed. Delete this tuple first and re-add with the new value.`	The user tries to change the TRIGEVNT field.
`OSAN0103: OAIN MAXSESNS exceeded.`	The user tries to increase the value for MAXSESN beyond the SOC limit.
`Decrease in sessions allowed.`	The user decreases (or does not increase) the value for MAXSESN.
`Unexpected error during tuple delete or SOC usage may still be over limit.`	The user deletes a tuple, but the value for MAXSESN (without the tuple) still exceeds the SOC limit.
`The node name specified is already in use in Table QMSMIS.`	The user tries to datafill a node name that is used in table QMSMIS (for MIS nodes).

OAFUNDEF

Table OAFUNDEF defines the functions used in processing OSSAIN calls. A function is a service or portion of a service that can be provided by either a service node, a live operator, or a TOPS automated system. Tables OANODINV (page 284) and CT4QNAMS (page 327) must be datafilled first.

In an OSAC network, table OAFUNDEF requires parallel datafill between the OSAC host switch and the OSAC remote switch.

Function type refinements

Refer to the following list for refinement datafill in table OAFUNDEF that applies to each function type (FUNCAREA field):

- For SNs, the following refinements apply:
  - ORIGSERV
  - TRIGEVNT
  - TRIGPOOL
  - SBTIMEOUT
  - SBTIMER
  - ISAUTOFN
  - NEWFUNC
  - ANSONVLC

- — CAMHERE
- — CALLQ
- — CONVOICE
- — DISPROUT
- — HRSESNPL
- — USESERV
- — DARECALL

- For TOPS operators, the following refinements apply:
  - — OPRCT4Q
  - — QREFINMT
  - — CQORDER
  - — DARECALL
- For TOPS automated systems, the following refinements apply:
  - — AUTOSYS
  - — OPRONFL or OPRCT4Q/QREFINMT/CQORDER

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
FUNCID		0 to 1022	Function identifier. Enter the index associated with the function.
FUNCNAME		Alphanumeric up to 16 characters	Function name. Enter the name of the function.
FUNCAREA		See subfields	Function area. This field consists of subfield FUNCTYPE and its refinements.

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
	FUNCTYPE	SN, TOPSOPER, TOPSAUTO	Function type. This subfield specifies the function provider type. Enter SN for functions provided by an SN and also datafill refinements ORIGSERV, TRIGEVNT, ISAUTOFN, and CAMHERE. Enter TOPSOPER for functions provided by a live operator and also datafill refinements OPRCT4Q and QREFINMT. Enter TOPSAUTO for functions provided by a TOPS automated system and also datafill refinements AUTOSYS and OPRONFL or OPRCT4Q/QREFINMT/CQORDER for DAS autosys.
	ORIGSERV	TASERV, DASERV, INTCSERV	Originating service. If FUNCTYPE is set to SN, datafill this refinement for the base service for calls originating at the DMS that route to an SN. Enter TASERV for Toll and Assist. Enter DASERV for directory assistance. Enter INTCSERV for Intercept. ***Note 1:*** Calls receiving DA or Intercept service are in a restricted mode while at the SN. DA and Intercept calls should be routed to an SN only to determine billing for the call prior to transferring the call to an operator. ***Note 2:*** The base service is *not* applied to calls originated by the SN or to calls transferred by the SN or operator.
	TRIGEVNT	Y, N	Trigger event. If FUNCTYPE is set to SN, datafill this refinement. Enter Y or N to specify whether or not this function is used for a trigger event. If set to Y, also datafill the TRIGPOOL refinement.
	TRIGPOOL	Alphanumeric up to 16 characters	Trigger session pool. Enter the name of the trigger event session pool to use when the call triggers back to this function. The session pool must be datafilled in table OASESNPL with TRIGEVNT set to Y.

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
	SBTIMOUT	Y, N	Session Begin Timeout. If FUNCTYPE is set to SN, datafill this refinement. Enter Y or N to specify whether or not the DMS switch starts a timer after sending a Session Begin message to the SN. If set to Y, also datafill the SBTIMER refinement.
	SBTIMER	1 to 7	Session Begin Timer. Enter the value in seconds for the timer. If the timer expires, the call is routed using the origination failure fields in table OAFNDISP.
	ISAUTOFN	Y, N	Automated function. If FUNCTYPE is set to SN, datafill this refinement. Enter Y or N to specify whether or not this SN function is associated with an automated service. If set to Y, also datafill the NEWFUNC refinement. ***Note:*** For “no automation” calls to this SN function, the ISAUTOFN field should be set to Y, with a TOPSOPER function datafilled in the NEWFUNC refinement. This new function is used to route the call to an operator function.
	NEWFUNC	Alphanumeric up to 16 characters	New function. Enter the name of the TOPSOPER function used to route the call.This new function must already be datafilled in table OAFUNDEF with a FUNCTYPE of TOPSOPER.
	ANSONVLC	Y, N	Answer on voice link connection. Enter Y or N to determine whether or not to return answer on connection of a voice link ***Note***: The ANSONVLC subfield is only applicable to incoming calls on ISUP trunks.
	CAMHERE	Y, N	CAM location. If FUNCTYPE is set to SN, datafill this refinement. Enter Y when routing and queuing is not centralized at an OSAC node and also datafill the CALLQ and CONVOICE refinements. Enter N when routing and queuing is centralized at an OSAC node and also datafill refinement HRSESNPL.

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
	CALLQ	CQ0 to CQ254	Call queue. If CAMHERE is set to Y, datafill this refinement. Enter the OSSAIN call queue from table QMSCQDEF.
	CONVOICE	Y, N	Connect voice. If CAMHERE is set to Y, datafill this refinement. Enter Y or N to specify whether or not the switch should establish a voice link connection before beginning a session with the SN. If set to Y, also datafill the DISPROUT refinement. ***Note:*** When set to Y, the switch uses datafill in table SNVLGRP to select the voice link for the given SN and function pair.
	DISPROUT	Y, N	Disposition routing. Enter Y or N to specify whether or not the switch should perform disposition routing if the automatic voice link connection attempt fails. If set to Y, the switch uses datafill for the origination failure action in table OAFNDISP. The maximum number of transfers that can occur while attempting to connect a call to an SN for a given function is specified in table OAINPARM (page 357). If set to N, the switch routes the call to the SN without a voice link connection.
	HRSESNPL	Alphanumeric up to 16 characters	Host-remote session pool. If CAMHERE is set to N, datafill this refinement. Enter the host-remote session pool used for messaging between the OSAC host switch and the OSAC remote switch. The session pool must be datafilled in table OASESNPL as an OSACORIG type.
	AUTOSYS	MCCS, DAS	Automated system. If FUNCTYPE is set to TOPSAUTO, datafill this refinement. Enter the TOPS automated system providing the function. ***Note 1:*** MCCS will route the call to MCCS. If MCCS is not available for the call then it is routed to operator. ***Note 2:*** Over an ONP, any remaining AABS will be replaced with MCCS and the call will route to MCCS.

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
	OPRONFL	Alphanumeric up to 32 characters	Operator backup on failure. If FUNCTYPE is set to TOPSAUTO, datafill this refinement. It indicates the CT4Q used as input to the QMS POSTAUTO refinement ordering if operator backup is required for the system specified in AUTOSYS. Enter the value of the CT4Q name from table CT4QNAMS.
	OPRCT4Q	Alphanumeric up to 32 characters	Operator call type for queuing. If FUNCTYPE is set to TOPSOPER or TOPSAUTO:DAS, datafill this refinement. It indicates the CT4Q used for final call queue assignment to provide this function. For TOPSAUTO:DAS it also determines the service and thus the DA vendor. Enter the value of the CT4Q name from table CT4QNAMS.
	QREFINMT	Y, N	Queue refinement. If FUNCTYPE is set to TOPSOPER or TOPSAUTO:DAS, datafill this refinement. Enter Y or N to specify whether the switch should apply QMS refinements to the OPRCT4Q before routing the call to the operator or DAS. If set to Y, also datafill the CQORDER refinement.
	CQORDER	PREOPR, POSTAUTO, RECALL, ASST	Call queue order. Enter the CT4Q refinement order to use for the operator function. These values map to existing datafill in tables TQORDERA/TQORDERB (QMS tables).

Table 38 Datafilling table OAFUNDEF

Field	Subfield or refinement	Entry	Explanation and action
	USESERV	Y, N	Use ORIGSERV. This field is not needed to set the service with ORIGSERV for initial call processing. It is used for transitions only. N means to allow the switch to change the DA service to TA when an operator transfers the call to an SN. All other transition scenarios will retain the service of the call. Y means to ensure the service datafilled in field ORIGSERV of OAFUNDEF is used when transferring or triggering to the function. If a service switch occurs, then other fields may be effected based on service switch rules defined in the OAP Specification document. ***Note:*** Whenever the OAFUNDEF option USESERV switches services, the same side effects occur as when an OAP Service Change Request is processed. Consult the OAP Specifications Document for details.
	DARECALL	Y, N	DA Recall. This field is not used for initial call processing. N is not a DA recall Y means the call going to the function is a DA recall and the switch will increment the DA recall counter and set operator indicators for recall if going to an operator.

OAFUNDEF example—standalone switch

The following figure shows example datafill. In the example, the branding function is provided by an SN using the CQ0 call queue. Automatic voice connection and disposition routing are enabled for the branding function.

Figure 129 MAP display example for table OAFUNDEF (standalone)

```
FUNCID      FUNCNAME             FUNCAREA
-------------------------------------------------------------------------------
1           BRANDING             SN  TASERV N N N N Y Y CQ0 N
2           0_MINUS_OPER         TOPSOPER  0_MINUS Y POSTAUTO
3           TA_AUTO              TOPSAUTO  MCCS 0_PLUS
4           DA_OPER              TOPSOPER  OPER_DA N
```

OAFUNDEF example—OSAC host switch

The following figure shows example datafill that is parallel. In the example, the QMS CAM is set to Y and the call queue is datafilled.

Figure 130 MAP display example for table OAFUNDEF (host)

```
FUNCID      FUNCNAME             FUNCAREA
-------------------------------------------------------------------------------
4           BRANDING             SN  TASERV N N Y 7 N Y Y CQ3 Y Y
```

OAFUNDEF example—OSAC remote switch

The following figure shows example datafill that is parallel. In the example, the QMS CAM is set to N and the host-remote session pool (REM_2_HOST) is datafilled.

Figure 131 MAP display example for table OAFUNDEF (remote)

```
FUNCID      FUNCNAME             FUNCAREA
-------------------------------------------------------------------------------
4           BRANDING             SN  TASERV N N Y 7 N Y N REM_2_HOST
```

OAFUNDEF error messages

A standard table control error message is displayed when the user tries to datafill a NEWFUNC function name before it is defined in table OAFUNDEF. The following table lists other possible error messages.

Table 39 Error messages for table OAFUNDEF

Error message	Explanation
`Only session pools with TRIGEVNT = Y may be datafilled in field TRIGPOOL`	The user tries to datafill TRIGPOOL with a session pool from table OASESNPL that has TRIGEVNT set to N.
`OSSAIN function names must be unique.`	The user tries to add two tuples with the same function name.
`AABS and DAS are the only TOPS automated systems that are currently supported for OSSAIN.`	The user tries to datafill the AUTOSYS refinement with an automated system other than AABS or DAS. See Note regarding AABS.
`Only session pools of ORIGTYPE OSACORIG may be datafilled in field HRSESNPL.`	The CAMHERE field is set to N and the user tries to datafill a session pool that is not set to OSACORIG.
`Field <fieldname> must be datafilled with a CT4Q from table CT4QNAMS that has a SYSTEM value of TOPSOPR.`	The CT4Q in fields OPRCT4Q or OPRONFL are not datafilled with a CT4Q from table CT4QNAMS that has SYSTEM set to TOPSOPER.

Table 39 Error messages for table OAFUNDEF

Error message	Explanation

`This function is datafilled in table XXXXXXXX.`
 	
`Since field FUNCTYPE is being changed to TOPSAUTO, you must remove any references to this function from the OSSAIN trigger tables.`
	The user tries to change the FUNCTYPE subfield from SN to TOPSAUTO and the function is used in an OSSAIN trigger table.
	
	XXXXXXXX is the name of the table using the function name.

`This function name is in use in table XXXXXXXX.`
 	
`You must remove all references to this function name before you can delete it.`
	The user tries to delete a tuple in table OAFUNDEF that has a function name used in one of the following tables:
	
	- OAFNDISP
	- OAFUNBLK
	- OACNNPRF
	- OATLKPRF
	- OACAUPRF
	- OADSCPRF
	- OACTLDEF
	- OAINPRE
	- SNVLGRP
	
	XXXXXXXX is the name of the table using the function name.

`Field NEWFUNC must be datafilled with a function from table OAFUNDEF that has a functype value of TOPSOPER.`	The user tries to datafill the NEWFUNC refinement with a function that has a FUNCTYPE other than TOPSOPER.

`This function name is in use in NEWFUNC field of table OAFUNDEF.`
 	
`You must remove all references to this function name before you can delete it.`
	The user tries to delete a function name that is referenced in the NEWFUNC refinement.

`This function is datafilled in table OAFUNDEF.`
 	
`Since field FUNCTYPE is being changed from TOPSOPER, you must remove any references to this function from the NEWFUNC field in table OAFUNDEF.`
	The user tries to change the FUNCTYPE from TOPSOPER to TOPSAUTO or SN when the TOPSOPER function is used in the NEWFUNC refinement.

OAFNDISP

Table OAFNDISP defines the disposition taken when an OSSAIN call is deflected or overflowed by the QMS CAM. It also defines the disposition for blocked functions and failures to send an OAP session begin message to an SN (origination failures).

Outpulsing disposition

OSSAIN has the option to outpulse to a designated DN when a session with an SN cannot be obtained. This disposition is available only to calls that have not received any processing from an SN and which are in the *initial call setup* state. It is not available for calls that transition or trigger to an SN.

A call in the initial call setup state is defined as one that arrives at the TOPS switch and, through QMS refinements, is routed to OSSAIN processing. The final CT4Q obtained during QMS refinements is mapped to a control list, which contains the name of a function provided by an OSSAIN SN. Datafill in table OAFNDISP specifies the disposition used for the call if it fails to establish a session with an SN.

OAFNDISP refinement values

In table OAFNDISP, the FUNCNAME field refers to the name of the requested function (defined in table OAFUNDEF). The other eight fields in table OAFNDISP define two sets of four types of actions. One set applies to initial call setup calls (fields beginning with I); the other set applies to transition/trigger calls (fields beginning with T).

The eight action fields are as follows:

- IDFLACTN (initial deflection)
- IOVFACTN (initial overflow)
- IBLKACTN (initial blocking)
- IOFLACTN (initial origination failure)
- TDFLACTN (transition deflection)
- TOVFACTN (transition overflow)
- TBLKACTN (transition blocking)
- TOFLACTN (transition origination failure)

Four possible disposition values can be specified, each with their own refinements:

- TREAT (treatment)—This value indicates that the call should be routed to treatment. TREAT has one refinement: a treatment name from table TMTCNTL must be specified.
- GOTOFN (go to function)—This value indicates that the call should be transferred to another function. GOTOFN has one refinement: a function name from table OAFUNDEF must be specified.
- GOTOCTL (go to control list)—This value indicates that the call should be transferred to another control list. GOTOCTL has one refinement: a control list name from table OACTLDEF must be specified.
- CLDOUTP (called outpulsing)—This value indicates that the call should be outpulsed and floated. CLDOUTP has three refinements:
  - USE0PLUS. When set to Y, the B-party number is used if it is available from call details. When set to N, the DN in the DIRECTDN refinement is used.
  - DIRECTDN. When set to Y, a 1- to 18-digit DN must be datafilled. When set to N, the B-party number is used. Otherwise, the call routes to treatment.
  - ALLOWTRG. When set to Y, the current trigger profile index is applied to the call when floating. When set to N, the trigger profile index is not applied.

  Note 1: The CLDOUTP disposition value applies only to the four *initial call setup* action fields and not to the transition action fields.

  Note 2: When USE0PLUS is set to N, DIRECTDN must be set to Y with a DN specified.

  Note 3: All calls that complete using the DIRECTDN refinement are billed as toll-free, regardless of whether the called number was signaled to the TOPS switch or obtained through datafill in table OAFNDISP. Downstream AMA processing can identify these records by the network service ID contained in the billing record. This ID is datafilled for the control list in table OACTLDEF.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 40 Datafilling table OAFNDISP

Field	Subfield or refinement	Entry	Explanation and action
FUNCNAME		Alphanumeric up to 16 characters	Function name. Enter the function name from table OAFUNDEF.
IDFLACTN		TREAT, GOTOFN, GOTOCTL, CLDOUTP	Initial deflection action. Enter the action for calls that are deflected from the QMS CAM, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list. - Enter CLDOUTP and its refinements to outpulse to a called DN.
IOVFACTN		TREAT, GOTOFN, GOTOCTL, CLDOUTP	Initial overflow action. Enter the action for calls that are overflowed from the QMS CAM, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list. - Enter CLDOUTP and its refinements to outpulse to a called DN.
IBLKACTN		TREAT, GOTOFN, GOTOCTL, CLDOUTP	Initial blocking action. Enter the action for calls that are blocked from the requested function, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list. - Enter CLDOUTP and its refinements to outpulse to a called DN. (See Note.)

Table 40 Datafilling table OAFNDISP

Field	Subfield or refinement	Entry	Explanation and action
IOFLACTN		TREAT, GOTOFN, GOTOCTL, CLDOUTP	Initial origination failure action. Enter the action for calls that have origination failures, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list. - Enter CLDOUTP and its refinements to outpulse to a called DN.
TDFLACTN		TREAT, GOTOFN, GOTOCTL	Transition deflection action. Enter the action for calls that are deflected from the QMS CAM, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list.
TOVFACTN		TREAT, GOTOFN, GOTOCTL	Transition overflow action. Enter the action for calls that are overflowed from the QMS CAM, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list.
TBLKACTN		TREAT, GOTOFN, GOTOCTL	Transition blocking action. Enter the action for calls that are blocked from the requested function, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list.

Table 40 Datafilling table OAFNDISP

Field	Subfield or refinement	Entry	Explanation and action
TOFLACTN		TREAT, GOTOFN, GOTOCTL	Transition origination failure action. Enter the action for calls that have origination failures, as follows: - Enter TREAT and a treatment name to route to treatment. - Enter GOTOFN and a function name to route to another function. - Enter GOTOCTL and a control list name to route to a control list.
Note: At initial call arrival, if the OSSAIN function used to route the call to an SN is blocked for the calling subscriber, the call receives the default treatment datafilled in table OAINPARM instead of receiving the CLDOUTP disposition action datafilled in the IBLKACTN field.			

OAFNDISP example

The following figure shows example datafill.

Figure 132 MAP display example for table OAFNDISP

```
FUNCNAME          IDFLACTN                  IOVFACTN
          IBLKACTN              IOFLACTN
                  TDFLACTN                  TOVFACTN
          TBLKACTN              TOFLACTN
-----------------------------------------------------------------------
BRANDING         CLDOUTP Y Y 9191234567 Y   CLDOUTP Y N N
          GOTOFN TOPS_BRANDING   GOTOFN LIVE_OPER
                  GOTOFN TOPS_BRANDING      GOTOCTL CTLLIST3
          TREAT VACT             GOTOFN ALT_NODE
```

OAFNDISP error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a function name before it is defined in table OAFUNDEF
- when the user tries to datafill a control list name before it is defined in table OACTLDEF

Note: Refer to “Direct transfers and new subscriber originations datafill” for information on table OACTLDEF.

OSAC parallel datafill

OSAC processing requires datafill in all the base call processing and voice connectivity OSSAIN tables. However, in five of these tables, the datafill must be *parallel* between the OSAC host switch and the OSAC remote switch. Parallel datafill ensures that each OSAC switch is aware of the other switches and SNs in the network. This allows the OSAC host switch to distribute centralized SN sessions and make voice connections correctly to OSAC remote switches.

Tables requiring parallel datafill

The tables in the following list are described in this chapter along with examples of parallel datafill at both the OSAC host switch and the OSAC remote switch. Refer to the following page number for information on each table:

- OANODNAM, page 282
- OANODINV, page 284
- OASESNPL, page 295
- OAFUNDEF, page 301
- OAVLMAP, page 318

Verification of parallel datafill

The OSAC switches verify parallel datafill in these tables, as follows:

- Tables OANODNAM and OANODINV—during the node RTS (between OSAC nodes) and during the connection RTS, TST, and audit (between the OSAC remote and the OSN).
- Table OASESNPL—during RTS and during call processing.
- Tables OAFUNDEF and OAVLMAP—during call processing.

Note: The switch generates a log report when the parallel datafill check fails. Refer to Chapter 11: “OSSAIN logs,” for more information on OSAC log reports.

OSSAIN voice connectivity datafill

The translations tables for OSSAIN voice connectivity are described in the following table. The tables are listed in the order in which they are to be datafilled.

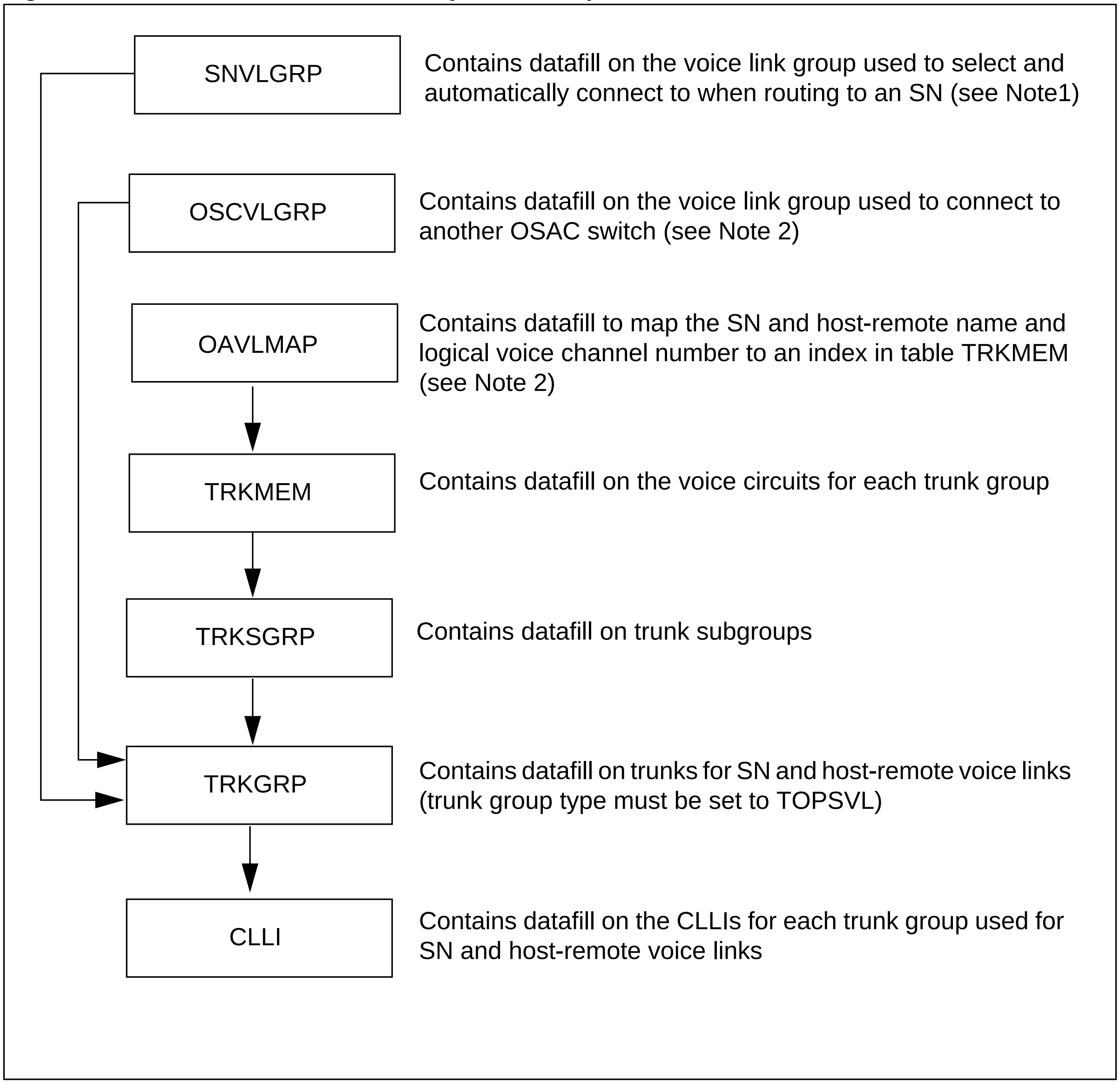
Note: This section describes the datafill only for tables OAVLMAP, OSCVLGRP, and SNVLGRP. Please refer to the *Customer Data Schema Reference Manual* for datafill information on tables CLLI, TRKGRP, TRKSGRP, and TRKMEM**.**

Table 41 Datafill sequence for SN voice connectivity

Table name	Explanation
CLLI	The Common Language Location Identifier table contains the CLLIs for each trunk group.
TRKGRP	The Trunk Group table contains information on trunk groups, such as the trunk group type (for OSSAIN, the type is TOPSVL). ***Note:*** The host-remote voice trunk is datafilled with the direction of 2W (two-way) at the OSAC host switch and IC (incoming) at the OSAC remote switch.
TRKSGRP	The Trunk Subgroup table contains information on subgroup signaling for the TOPSVL trunk group type.
TRKMEM	The Trunk Member contains information on the hardware locations of trunk members.
OAVLMAP	The OSSAIN Voice Link Mapping table contains information to map the SN and host-remote name and logical voice channel to an index in table TRKMEM. This table is used when the SN selects the voice link. ***Note:*** In the OSAC configuration, this table requires parallel datafill. Examples in this chapter show datafill for a standalone switch, and both an OSAC host switch and OSAC remote switch.
OSCVLGRP	The OSAC Voice Link Group table defines the voice link group to use for a voice connection between an OSAC Host and OSAC Remote. This table is used in a switch if it acts as an OSAC Host for an SN. ***Note:*** This table is used only in the OSAC network and only at the OSAC host switch.
SNVLGRP	The Service Node Voice Link Group table defines the voice link group that is selected by the DMS switch to connect to automatically when routing a call to an SN. The voice link corresponds to a particular SN and function pair. This table is used when the switch selects the voice link.

The following figure shows table dependencies for the seven tables that require datafill for SN voice connectivity.

Figure 133 OSSAIN voice connectivity datafill dependencies



Note 1: Tables OANODINV and OAFUNDEF must be datafilled before datafilling table SNVLGRP.

Note 2: Table OANODINV must be datafilled before datafilling tables OSCVLGRP and OAVLMAP.

Note 3: For information on OSSAIN voice link attributes, refer to “Trunk group type TOPSVL” on page 75.

OAVLMAP

Table OAVLMAP maps logical voice channels to an index in table TRKMEM. All voice circuits first must be datafilled in table TRKMEM. The SN must supply the logical voice channel number for the voice connection in the OAP message.

For broadcast voice links, the SN must supply the node ID as well as the logical voice channel number in the OAP message. Additional datafill in OAVLMAP specifies the method the switch uses to establish the voice path. Datafill also specifies the maximum number of simultaneous connections to a particular broadcast voice link.

Note 1: A voice link can function as only one type: either non-broadcast or broadcast.

Note 2: Only SN voice links, not host-remote voice links, can be set up for broadcasting.

Note 3: Tuples cannot be deleted from table TRKMEM until all references to that circuit are removed from table OAVLMAP. Tuples cannot be deleted from table OANODINV until all references to that node name are removed from table OAVLMAP.

In an OSAC network, table OAVLMAP requires parallel datafill between the OSAC host switch and the OSAC remote switch. For details on OSAC, refer to Chapter 3: “OSAC call processing.”

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 42 Datafilling table OAVLMAP

Field	Subfield or refinement	Entry	Explanation and action
NDANDCH		See subfields	Node and channel. This field consists of subfields NODENAME and VOICENUM.
	NODENAME	Alphanumeric up to 12 characters	Node name. Enter the node name from table OANODINV for this voice channel.
	VOICENUM	0 to 8191	Voice channel number. Enter the number supplied by the SN in the OAP message (this is the message requesting a voice connection between the switch and the SN).
CLLI		Alphanumeric up to 16 characters	Common language location identifier. Enter the CLLI name of the trunk used to make the voice connection between the switch and the SN. This CLLI maps to table TRKMEM. The trunk group type should first be set to TOPSVL in table TRKGRP.

Table 42 Datafilling table OAVLMAP

Field	Subfield or refinement	Entry	Explanation and action
EXTRKNUM		0 to 9999	External trunk number. Enter the external trunk number that maps to table TRKMEM.
BCSTAREA		See subfields	Broadcast area. This field consists of subfield BCST_SEL and its refinements.
	BCST_SEL	Y, N	Broadcast selector. Enter Y if the voice link is used for broadcasting, and also datafill the refinements CUTTHRU and MAXCONNS. Enter N if the voice link is not used for broadcasting.
	CUTTHRU	IMMEDIATE, HKCHG	Cut-through. Enter IMMEDIATE to enable the voice path as soon as the switch makes the network connection. Enter HKCHG to disable the voice path until the switch detects an off-hook on the voice link of at least 10 milliseconds.
	MAXCONNS	2 to 1023	Maximum connections. Enter the maximum number of simultaneous connections allowed on the broadcast voice link. ***Note:*** The maximum value 1023 applies only to ENET offices with CUTTHRU set to IMMEDIATE. Otherwise, the maximum value is 255.
	STRSEL	Y, N	Specialized tone receiver selector. Enter Y if STR supervision is performed on the calling party while the call is connected to the Broadcast voice link. If set to Y, also datafill the STRDIGIT subfield. Enter N if STR supervision is not performed. ***Note:*** For calls connected to a broadcast voice link in the OSAC environment, the OSAC Protocol Release 5 is required.
	STRDIGIT	STAR, OCTO	STR digit. Enter STAR (*) or OCTO (#) for the STR digit.

OAVLMAP example—standalone switch

The following figure shows example datafill. In the example, OSSAINVL2 3 is used for voice link broadcasting.

Figure 134 MAP display example for table OAVLMAP (standalone)

```
NDANCH                 CLLI          EXTRKNUM   BCSTAREA
----------------------------------------------------------------------
NODE_1 10              OSSAINVL1     2          N
NODE_2 11              OSSAINVL2     3          Y IMMEDIATE 800  Y OCTO
```

OAVLMAP example—OSAC host switch

The following figure shows example datafill. In the example, the OSAC host switch makes voice connections for the OSAC remote switch (SWITCH_B). The OSAC host selects trunk group OSACVL_B. It then selects the most idle voice link from that trunk group (for example, member 2). The OSAC host maps the actual group and trunk number (OSACVL_B 2) to the logical voice channel number for the OSAC remote switch (channel 2). The OSAC host sends this channel number to the OSAC remote.

Figure 135 MAP display example for table OAVLMAP (host)

```
NDANCH               CLLI           EXTRKNUM    BCSTAREA
----------------------------------------------------------------------
SWITCH_B 1           OSACVL_B       1           N
SWITCH_B 2           OSACVL_B       2           N
SWITCH_B 3           OSACVL_B       3           N
SWITCH_B 4           OSACVL_B       4           N
```

OAVLMAP example—OSAC remote switch

The following figure shows example datafill. In the example, the OSAC remote switch receives the logical voice channel 2 from the OSAC host switch (SWITCH_A). The OSAC remote then maps this channel to the actual group and trunk number (OSACVL_A 2).

Figure 136 MAP display example for table OAVLMAP (remote)

```
NDANCH               CLLI           EXTRKNUM    BCSTAREA
----------------------------------------------------------------------
SWITCH_A 1           OSACVL_A       1           N
SWITCH_A 2           OSACVL_A       2           N
SWITCH_A 3           OSACVL_A       3           N
SWITCH_A 4           OSACVL_A       4           N
```

OAVLMAP error messages

A standard table control error message is displayed when the user tries to datafill a node name before it is defined in table OANODNAM. The following table lists other possible error messages.

Table 43 Error messages for table OAVLMAP

Error message	Explanation
`This CLLI and external trunk name must be datafilled in table TRKMEM before datafilling this table.`	The user tries to datafill a CLLI and EXTRKNM before that voice circuit is defined in table TRKMEM.
`CLLI and external trunk name is already datafilled against this node.`	The user tries to datafill a duplicate CLLI and external trunk name combination against the same node.
`Trunk group type for CLLI must be TOPSVL.`	The user tries to datafill a CLLI that is not of trunk group type TOPSVL.
`Only service nodes of PM type OSNM or OSAC may have voice connections with the switch.`	The user tries to datafill a NODENAME that has the OSN node type (in table OANODINV).
`OSAC nodes must have a relation of OTHER in table OANODINV.`	The user tries to datafill an OSAC node with a RELATION of SELF (in table OANODINV).
`TOPSVL must be datafilled as outgoing for OSNM nodes.`	The user tries to datafill an OSNM node with a voice link that is not datafilled as outgoing (in table TRKSGRP).
`The node for the logical voice channel must be first datafilled in table OANODINV.`	The user tries to datafill the voice link of a node before the node is defined in table OANODINV.
`Only nodes of PM type OSNM can have the BCST field set to Y.`	The user tries to set the BCST_SEL field to Y for a NODENAME that has the OSAC node type (in table OANODINV).
`The MAXCONNS value cannot exceed 255 in a JNET office.`	The user tries to set the MAXCONNS field greater than 255 on a broadcast voice link in a JNET office.
`The MAXCONNS value cannot exceed 255 when the CUTTHRU field is set to HKCHG.`	The user tries to set the MAXCONNS field greater than 255 on a broadcast voice link whose CUTTHRU field is set to HKCHG in an ENET office.
`The CLLI and EXTRKNM is already in use by another voice link. This is not allowed for voice links used for broadcasting.`	The user tries to datafill the same CLLI and external trunk name combination against a voice channel number used for a broadcast voice link. ***Note:*** This restriction does not apply to non-broadcast voice links, which can have multiple logical channels to the same physical trunk.

Table 43 Error messages for table OAVLMAP

Error message	Explanation
`Trunk must be INB at the MAP before adding or deleting a tuple with the BCST_SEL set to Y, or when changing the BCST_SEL or MAXCONNS field of an existing tuple.`	The user tries to change the datafill for a broadcast voice link that is not in the INB (installation busy) state at the MAP, as follows: - adding a new tuple with the BCST_SEL field set to Y - deleting a tuple whose BCST_SEL field is set to Y - changing the BCST_SEL field of an existing tuple - changing the MAXCONNS field of an existing tuple
`Voice link audit in progress - try again later.`	The user tries to change the MAXCONNS field at the same time that a voice link audit is running.
`The CLLI name is used in table XXXXXXXX; it cannot be reused here.`	The user tries to datafill a CLLI that is already used in table SNVLGRP. XXXXXXXX is the name of the table containing the CLLI.

OSCVLGRP

Table OSCVLGRP defines the voice link group for the voice connection between an OSAC host switch and an OSAC remote switch. This table requires datafill if a switch acts as the OSAC host switch for any SN. In an OSAC network, table OSCVLGRP is datafilled *only* at the OSAC host switch.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 44 Datafilling table OSCVLGRP

Field	Subfield or refinement	Entry	Explanation and action
NODENAME		Alphanumeric up to 12 characters	Node name. Enter the name of the OSAC remote switch from table OANODINV.
CLLI		Alphanumeric up to 16 characters	Common language location identifier. Enter the CLLI name (from table CLLI) for the OSAC voice link group to use for voice connections between the OSAC remote switch and OSAC host switch.

OSCVLGRP example

The following figure shows example datafill. In the example, the OSAC remote switch and voice link group are identified.

Figure 137 MAP display example for table OSCVLGRP

```
NODENAME          CLLI
----------------------------------------------------------------------
SWITCH_B          OSACVL_B
```

OSCVLGRP error messages

The following table lists possible error messages.

Table 45 Error messages for table OSCVLGRP

Error message	Explanation
`Only node IDs of PM type OSAC can be datafilled in this table.`	The NODENAME is not datafilled as an OSAC node type (defined in table OANODINV).
`This CLLI must be datafilled in Table TRKGRP before datafilling this table.`	The CLLI is not datafilled first in table TRKGRP.
`Trunk group type for CLLI must be TOPSVL.`	The trunk group for the CLLI is not datafilled as TOPSVL type (defined in table TRKGRP).
`The CLLI name is used in table XXXXXXXX; it cannot be reused here.`	The user tries to datafill a CLLI that is already used in table SNVLGRP. XXXXXXXX is the name of the table containing the CLLI.

SNVLGRP

Table SNVLGRP specifies a voice link for a particular SN and function pair. The switch uses datafill in this table if the voice link selection capability (SOC option OSAN0102) is enabled. The switch selects the voice link group from table SNVLGRP when automatic SN voice connections are required (CONVOICE field set to Y in table OAFUNDEF) or when the SN makes an OAP request to the switch for a voice connection.

With OSAN0102 enabled, the SN *does not* have to specify the logical voice channel number in the OAP request. This request also allows the switch to automatically *release* any previous voice link connection before processing the request.

Note: Existing OSAC host-remote voice links datafilled in table OSCVLGRP or SN voice links datafilled in table OAVLMAP *cannot* be reused in table SNVLGRP.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 46 Datafilling table SNVLGRP

Field	Subfield or refinement	Entry	Explanation and action
NODEFUNC		See subfields	Node and function. This field consists of subfields NODENAME and FUNCNAME.
	NODENAME	Alphanumeric up to 12 characters	Node name. Enter the node name from table OANODINV. This node must have an ONPMTYPE of OSNM.
	FUNCNAME	Alphanumeric up to 16 characters	Function name. Enter the function name from table OAFUNDEF. This function must have a FUNCTYPE of SN, with a CAMHERE value of Y.
CLLI		Alphanumeric up to 16 characters	Common language location identifier. Enter the name of the trunk group CLLI used to make the voice connection for the SN and function pair.

SNVLGRP example

The following figure shows example datafill. In the example, both SN_01 and SN_02 provide the calling card function. SN_01 also provides the AABS function. When routing a call to SN_01 for the calling card function, the voice link from voice link group VL_01 is used to serve the call.

When routing a call to SN_02 for the calling card function, the voice link from voice link group VL_02 is used to serve the call. Similarly, the voice link group VL_05 is used for the AABS function on SN_01.

Figure 138 MAP display example for table SNVLGRP

```
NODEFUNC              CLLI
---------------------------------------------------------------------
SN_01 CALLING_CARD    VL_01
SN_01 AABS_SVC        VL_05
SN_02 CALLING_CARD    VL_02
```

SNVLGRP error messages

The following table lists possible error messages.

Table 47 Error messages for table SNVLGRP

Error message	Explanation
`The node for the voice link must be first datafilled in table OANODINV.`	The user tries to datafill a NODENAME that is not defined first in table OANODINV.

Table 47 Error messages for table SNVLGRP

Error message	Explanation
`Only node ids of PM type OSNM can be datafilled in this table.`	The user tries to datafill a NODENAME that is not of the OSNM type in table OANODINV.
`Only SN functions can be datafilled in this table.`	The user tries to datafill a FUNCNAME that is not of the SN type in table OAFUNDEF.
`Only SN functions with the CAMHERE field set to Y can be datafilled in this table.`	The user tries to datafill a FUNCNAME that is not of the SN type with the CAMHERE field set to Y in table OAFUNDEF.
`This CLLI must be datafilled in Table TRKGRP before datafilling this table.`	The user tries to datafill a CLLI that is not defined first in table TRKGRP.
`Trunk group type for CLLI must be TOPSVL.`	The user tries to datafill a CLLI that is not of trunk group type TOPSVL.
`The voice link trunk group circuit must be datafilled as outgoing in table TRKSGRP.`	The user tries to datafill a CLLI that is not defined as outgoing in table TRKSGRP.
`The CLLI name is used in table XXXXXXXX; it cannot be reused here.`	The user tries to datafill a CLLI that is already used in table OAVLMAP or OSCVLGRP. XXXXXXXX is the name of the table containing the CLLI.

Direct transfers and new subscriber originations datafill

The translations tables for direct transfers (from an SN or operator position) and for new subscriber originations are described in the following table. The tables are listed in the order in which they are to be datafilled.

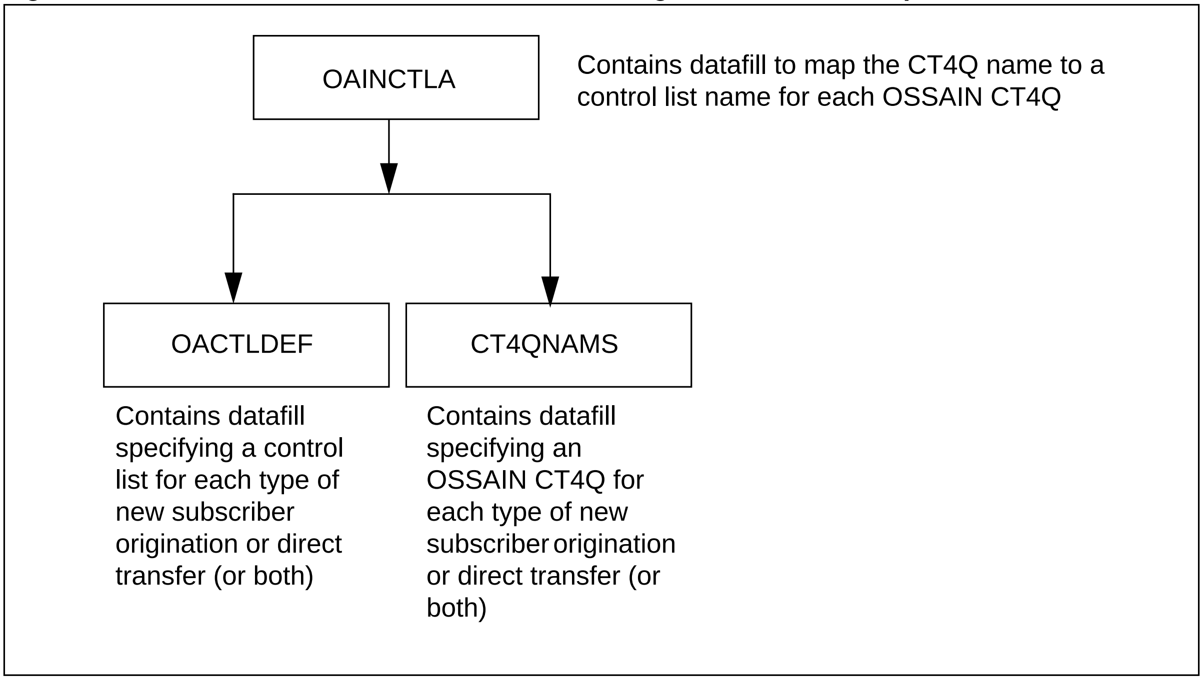
Note: New subscriber originations also require datafill in the QMS criteria tables. These tables allow the QMS PREOPR to refine the call origination type to a final CT4Q that maps to OSSAIN. Please refer to the *Customer Data Schema Reference Manual.*

Table 48 Datafill sequence for direct transfers and new subscriber originations

Table name	Explanation
OACTLDEF	The OSSAIN Control List Definition table defines all OSSAIN control lists.
CT4QNAMS	The Call Type for Queuing Names table specifies the CT4Q name and attributes. This table also associates the call trigger profile index with each CT4Q used by OSSAIN.
OAINCTLA	The OSSAIN Control List Assignment table associates a control list name with each CT4Q used by OSSAIN.

The following figure shows table dependencies for the three tables that require datafill for direct transfers and new subscriber originations.

Figure 139 Direct transfers and new subscriber originations datafill dependencies



OACTLDEF

Table OACTLDEF defines the control lists used in processing OSSAIN calls. A control list contains the name of one OSSAIN function. The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 49 Datafilling table OACTLDEF

Field	Subfield or refinement	Entry	Explanation and action
OACTLNUM		0 to 4094	OSSAIN control list number. Enter the index associated with the control list.
OACTLNAM		Alphanumeric up to 16 characters	OSSAIN control list name. Enter the name of the control list.
NETWRKID		0 to 32766	Network service identifier. Enter the ID to uniquely identify the service in the network for billing.
OAFUNCTS		1 function name	OSSAIN functions. Enter a single function name from table OAFUNDEF for this control list.

OACTLDEF example

The following figure shows example datafill.

Figure 140 MAP display example for table OACTLDEF

```
OACTLNUM    OACTLNAM       NETWRKID     OAFUNCTS
-------------------------------------------------------------------
0           0+AABS         3            (     TOPS_AABS) $
1           DA_NODE        7            (     OSSAIN_DA) $
2           ROUTER         2            (    ROUTE_NODE) $
3           YEL_PAGE       4            ( YELLOW_PAGES) $
```

OACTLDEF error messages

A standard table control error message is displayed when the user tries to datafill a function name before it is defined in table OAFUNDEF. The following table lists other possible error messages.

Table 50 Error messages for table OACTLDEF

Error message	Explanation
`At least one function must be specified in the list.`	The user tries to datafill a control list with less than one function.
`This control list name is in use in table XXXXXXXX.` `You must remove all references to this control list name before you can delete it.`	The user tries to delete a tuple in table OACTLDEF that has a control list name used in one of the following tables: - OAFNDISP - OACNNPRF - OATLKPRF - OACAUPRF - OADSCPRF - OADTFPRF - CT4QNAMS XXXXXXXX is the name of the table using the control list name.

CT4QNAMS

Table CT4QNAMS defines the attributes of each CT4Q name. The CT4Q assigned to a call is determined by the PREOPR ordering of the QMS criteria tables.

The SYSAREA field specifies unique refinements of the data associated with CT4Q names for TOPS and OSSAIN. The SYSAREA field contains a SYSTEM selector subfield with two possible values: TOPSOPR or OSSAIN.

Note 1: TOPS QMS also allows TOPS calls to receive OSSAIN preprocessing. Refer to “OSSAIN preprocessing datafill” on page 350, for more information on preprocessing datafill for table CT4QNAMS.

Note 2: For complete information on table CT4QNAMS, please refer to the *Customer Data Schema Reference Manual*.

Table 51 Datafilling table CT4QNAMS

Field	Subfield or refinement	Entry	Explanation and action
ITRIGIDX		N, Y (0 to 2046)	Initial trigger profile index. Enter Y or N to indicate whether or not to assign an index from table OATPRFIX. If set to Y, also datafill the trigger index from table OATPRFIX.

CT4QNAMS example

The following figure shows example datafill.

Figure 141 MAP display example for table CT4QNAMS

```
CT4QNUM  CT4QNAME       NOAMA    ITRIGIDX   SYSAREA
----------------------------------------------------------------------
0        UNSPEC          N        N         TOPSOPR Y UNSPEC   N  N  N
1        CAMA            N        N         TOPSOPR Y CAMA     N  N  N
2        1_PLUS          N        N         TOPSOPR Y 1+_Q     N  N  N
6        0_MINUS         N        N         TOPSOPR Y 0_MIN_Q  N  N  N
10       0_PLUS          N        N         TOPSOPR Y 0+_Q     N  N  N
14       DA_411          N        N         TOPSOPR Y DA_411_Q N  N  N
620      0_PLUS_OSSAIN   N        Y  0      OSSAIN
621      0_MINUS_OSSAIN  N        Y  0      OSSAIN
622      1_PLUS_OSSAIN   N        Y  0      OSSAIN
```

CT4QNAMS error message

A standard table control error message is displayed when the user datafills a function name before it is defined in table OATPRFIX. The following table displays that error messages.

Table 52 Error messages for table CT4QNAMS

Error message	Explanation
`The trigger index specified is not activated until datafilled in Table OATPRFIX.Table OATPRFIX and its associated trigger tables must be datafilled to activate this trigger index.`	The error message is displayed when the trigger index specified in field ITRIGIDX is not yet datafilled in table OATPRFIX.

OAINCTLA

Table OAINCTLA associates the OSSAIN control list with each CT4Q name. Each CT4Q datafilled in table OAINCTLA must first be assigned in table CT4QNAMS with the SYSTEM selector set to OSSAIN. The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 53 Datafilling table OAINCTLA

Field	Subfield or refinement	Entry	Explanation and action
CT4Q		Alphanumeric up to 32 characters	Call type for queuing. Enter the CT4Q name defined in table CT4QNAMS.
OALISTNM		1 control list name	OSSAIN control list name. Enter the control list name defined in table OACTLDEF.

OAINCTLA example

The following figure shows example datafill.

Figure 142 MAP display example for table OAINCTLA

```
CT4Q                          OALISTNM
-------------------------------------------------------------------
0+YELLOW                      YEL_PAGE
OSS_DA                        DA_NODE
DA_411                        DA_NODE
```

OAINCTLA error messages

A standard table control error message is displayed when the user datafills a control list name before it is defined in table OACTLDEF

Function blocking datafill

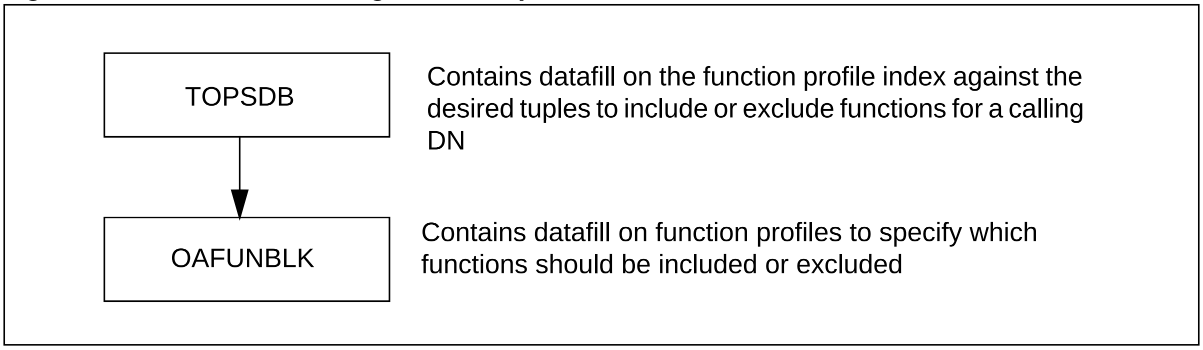
The translations tables for function blocking are described in the following table. The tables are listed in the order in which they are to be datafilled.

Table 54 Datafill sequence for function blocking

Table name	Explanation
OAFUNBLK	The OSSAIN Function Blocking table lists the functions that are included or excluded for a particular index, which is a DN referenced in table TOPSDB.
TOPSDB	The TOPS Database table indexes the DN that maps to available functions in table OAFUNBLK.

The following figure shows table dependencies for the two tables that require datafill for function blocking.

Figure 143 Function blocking datafill dependencies



OAFUNBLK

Table OAFUNBLK lists the functions that are included or excluded (from the available set of functions defined in table OAFUNDEF) for a particular originating DN. A default tuple of index 0 indicates that there are no functions excluded (that is, every defined function is allowed) for the current DN.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 55 Datafilling table OAFUNBLK

Field	Subfield or refinement	Entry	Explanation and action
IDX		0 to 1022	Index. Enter the index associated with the function set.
FNBLKSEL		INCLUDE, EXCLUDE	Function blocking selector. Enter the selector value to specify whether the functions are included or excluded. INCLUDE specifies that only the functions in the list (and no others) are *valid* for the index. EXCLUDE specifies that the functions in the list are *not valid* for the index.
OAFUNCTS		0 to 5 function names	OSSAIN functions. Enter up to 5 function names from table OAFUNDEF (field FUNCNAME).

OAFUNBLK example

The following figure shows example datafill.

Figure 144 MAP display example for table OAFUNBLK

```
IDX       FNBLKSEL            OAFUNCTS
--------------------------------------------------------------------
0         EXCLUDE             $
1         INCLUDE             BRANDING  NODE_AABS   YELLOW_PAGES $
2         INCLUDE             TOPS_BRANDING  TOPS_AABS $
3         EXCLUDE             YELLOW_PAGES $
```

OAFUNBLK error messages

A standard table control error message is displayed when the user datafills a function name before it is defined in table OAFUNDEF. The following table lists other possible error messages.

Table 56 Error messages for table OAFUNBLK

Error message	Explanation
`Tuple referred to by another table - use tabref to get potential table list.`	The user tries to delete a tuple in table OAFUNBLK that has an index in table TOPSDB.

TOPSDB

Table TOPSDB provides an index into several tables, one of which is OAFUNBLK. The field containing the index is also called OAFUNBLK.

Note: For complete information on table TOPSDB, please refer to the *Customer Data Schema Reference Manual*.

TOPSDB example

The following figure shows example datafill.

Figure 145 MAP display example for table TOPSDB

```
TDBKEY   TDBCLIDX  TDBDAOPT  TDBNORM   TDBSERV   OAFUNBLK
-----------------------------------------------------------------
0        0         0         0         0         1
1        1         0         0         0         0
2        2         0         0         0         2
3        4         0         0         0         1
```

Trigger datafill

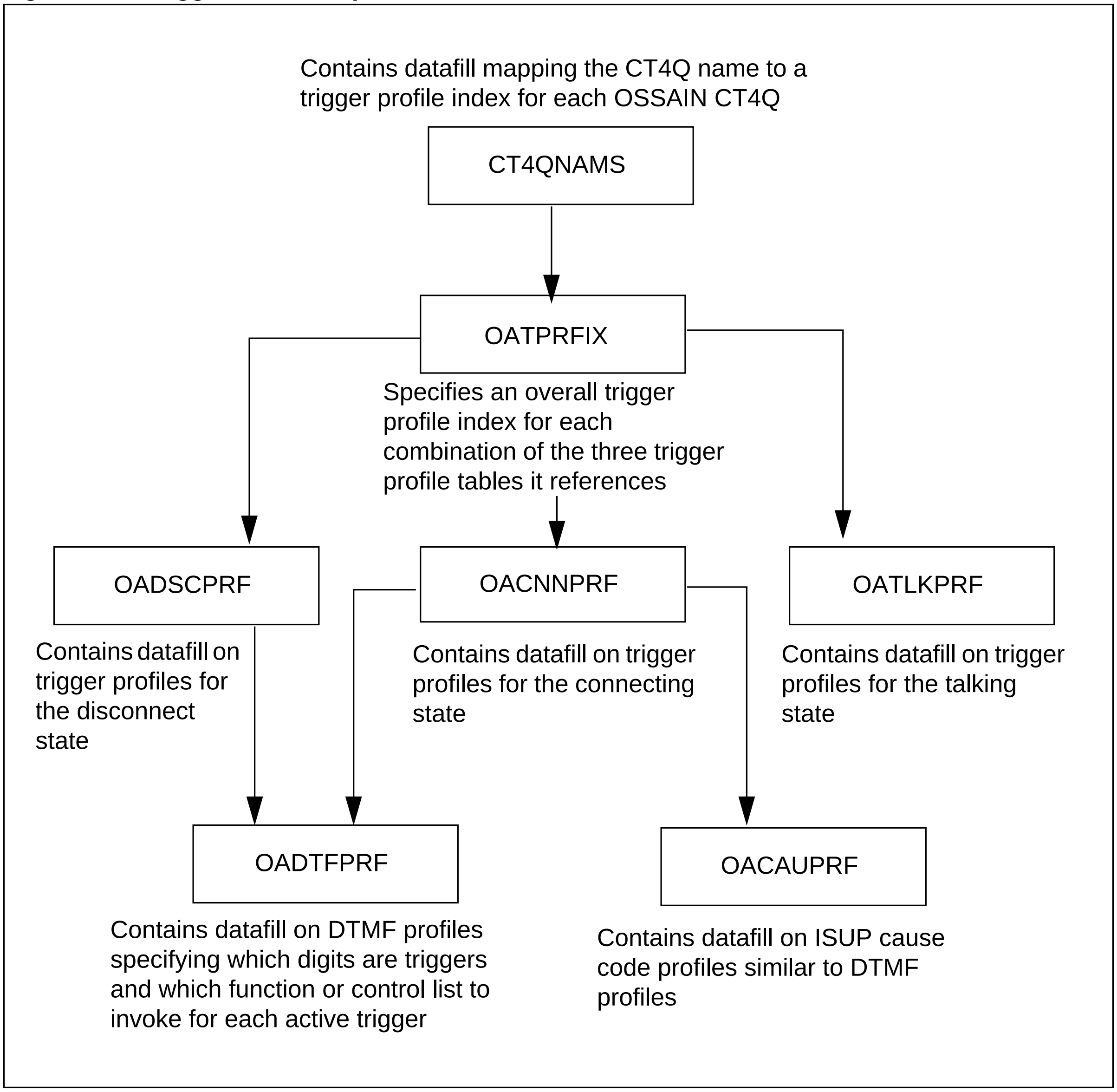
The translations tables for trigger datafill are described in the following table. The tables are listed in the order in which they are to be datafilled.

Table 57 Datafill sequence for triggers

Table name	Explanation
OADTFPRF	The OSSAIN DTMF Profile table contains DTMF profiles used by tables OACNNPRF and OADSCPRF.
OACAUPRF	The OSSAIN Cause Profile table contains ISUP cause profiles used by table OACNNPRF.
OACNNPRF	The OSSAIN Connecting Profile table contains trigger profiles applicable to the connecting trigger detection point (TDP).
OADSCPRF	The OSSAIN Disconnect Profile table contains trigger profiles applicable to the post disconnect TDP.
OATLKPRF	The OSSAIN Talking Profile table contains trigger profiles applicable to the talking TDP.
OATPRFIX	The OSSAIN Trigger Profile Index table associates the call trigger profile index with the connecting/talking/disconnect profile indexes.
CT4QNAMS	The Call Types for Queuing Names table associates a call trigger profile index with each CT4Q used by OSSAIN.
OAINCTLA	The OSSAIN Control List Assignment table associates a control list name with each CT4Q used by TOPS and OSSAIN.

The following figure shows table dependencies for the seven tables that require datafill for triggers.

Figure 146 Trigger datafill dependencies



OADTFPRF

Table OADTFPRF contains a list of DTMF triggers and corresponding actions. The action is a function or control list to which call control is passed as a result of the DTMF trigger event.

Note: If the SN requires call timing information for a call just prior to a trigger event, the GENAMA option (PROFILE field) should not be datafilled against that trigger event. Refer to "OATLKPRF" on page 344, for the stop conversation timing option.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 58 Datafilling table OADTFPRF

Field	Subfield or refinement	Entry	Explanation and action
DTMFIDX		0 to 2046	Dual-tone multifrequency index. Enter the index for the DTMF profile.
PROFILE		See subfields	Profile. This field contains a list of DTMF triggers and corresponding actions. It consists of subfields DIGIT, GENAMA, RLSFWD, RFLTONFL, and ACTSEL.
	DIGIT	0 to 9, STAR, OCTO	DTMF digit. Enter the DTMF digit, STAR, or OCTO for the trigger event. ***Note:*** The value STAR represents the * digit and the value OCTO represents the # digit.
	GENAMA	Y, N	Generate AMA. Enter Y or N to indicate whether or not an AMA record is generated during trigger processing.
	RLSFWD	Y, N	Release forward. Enter Y or N to indicate whether or not to release the forward party during trigger processing. ***Note:*** The RLSFWD subfield must be set to Y when the GENAMA subfield is set to Y.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden. ***Note:*** The RFLTONFL subfield may be set to Y only when the RLSFWD is set to N.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.

Table 58 Datafilling table OADTFPRF

Field	Subfield or refinement	Entry	Explanation and action
HOLDRCVR		Y, N	Hold receiver. Enter Y or N to indicate whether or not to retain the DTMF receiver linked to the calling party when the call transitions to an SN. ***Note:*** HOLDRCVR is not used for a trigger action that leads to an operator function.

OADTFPRF example

The following figure shows example datafill.

Figure 147 MAP display example for table OADTFPRF

```
DTMFIDX          PROFILE                                      HOLDRCVR
---------------------------------------------------------------------
3    (0 N N Y FUNCTION BRANDING) (STAR N N Y FUNCTION ADS_NODE) $  N
4    (0 N N Y FUNCTION YELLOW_PAGES) (1 N N N CTRLLIST YEL_PAGE)
(OCTO N N N FUNCTION SEQUENCE) $ Y
5 (STAR Y Y N FUNCTION DA_OPER) $ N
```

OADTFPRF error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a control list name before it is defined in table OACTLDEF
- when the user tries to datafill a function name before it is defined in table OAFUNDEF

The following table lists other possible error messages.

Table 59 Error messages for table OADTFPRF

Error message	Explanation
`RLSFWD MUST BE SET TO N TO PERMIT RFLTONFL SETTING OF Y`	The user tries to set RFLTONFL to Y when RSLFWD is set to Y.
`RLSFWD must be set to Y when GENAMA is set to Y.`	The user tries to set RLSFWD to N when GENAMA is set to Y.

OACAUPRF

Table OACAUPRF contains a list of ISUP classes, cause codes and corresponding actions. The action is a function or control list to which call control is passed as a result of the ISUP cause code trigger event. The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 60 Datafilling table OACAUPRF

Field	Subfield or refinement	Entry	Explanation and action
IDX		0 to 2046	Index. Enter the index for the cause code profile.
PROFILE		See subfields	Profile. This field contains a list of trigger events and corresponding actions. It consists of subfields RELCLASS, RFLTONFL, and ACTSEL.
	RELCLASS	NORM (1 to 31), CL2, CL3, CL4, CL5, CL6, CL7	Release class. Enter the release class. NORM specifies the normal event class and also requires a cause value in the range 1 to 31. This allows for a unique trigger event for each normal cause code. The other release classes (CL2 through CL7) do not require datafill for an associated cause code. ***Note:*** Refer to the following lists of release classes and NORM cause codes.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.

Release classes

- NORM - Normal Event Class - Cause codes 1-31
- CL2 - Resource Unavailable Class - Cause codes 34-47
- CL3 - Service Option Not Available Class - Cause codes 52-63
- CL4 - Service Option Not Implemented Class - Cause codes 65-79
- CL5 - Invalid Message Class - Cause codes 81-95
- CL6 - Protocol Error Class - Cause codes 97-111
- CL7 - Interworking Class - Cause code 127

NORM cause codes

- 1 - Unallocated Number
- 2 - No Route to Transit Network
- 3 - No Route to Destination
- 4 - Send Special Info Tone
- 5 - Misdialed Trunk Prefix
- 16 - Normal Clearing
- 17 - User Busy
- 18 - No User Responding
- 19 - No Answer From User
- 21 - Call Rejected
- 22 - Number Changed
- 25 - Translations Fail
- 26 - Call Returns
- 27 - Destination out of Service
- 28 - Address Incomplete
- 29 - Facility Rejected
- 30 - Apply Locally
- 31 - Normal Unspecified

Note: This list contains the current known ISUP release cause codes. However, any release cause code in the range 1 to 31 can be datafilled.

OACAUPRF example

The following figure shows example datafill.

Figure 148 MAP display example for table OACAUPRF

```
IDX       PROFILE
----------------------------------------------------------------------
2         (NORM 17 Y FUNCTION ADSNODE) (NORM 19 N FUNCTION ADSNODE)
   (CL2 Y FUNCTION ADSNODE) $
```

OACAUPRF error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a control list name before it is defined in table OACTLDEF
- when the user tries to datafill a function name before it is defined in table OAFUNDEF

OACNNPRF

Table OACNNPRF associates lists of trigger events and corresponding actions with a connecting profile index. The connecting profile index is then associated with a trigger profile index in table OATPRFIX.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 61 Datafilling table OACNNPRF

Field	Subfield or refinement	Entry	Explanation and action
CONNIDX		0 to 2046	Connecting index. Enter the profile index number.
PROFILE		See subfields	Profile. This field contains a list of trigger events and corresponding actions. It consists of subfields TRIGGER, SNCONTRL, RFLTONFL, and ACTSEL.

Table 61 Datafilling table OACNNPRF

Field	Subfield or refinement	Entry	Explanation and action
	TRIGGER	RING, OPLSF, NOANS, NTFY, FLASH, CLGD, PTYD, ANS, TRMT, $	Trigger. Enter the trigger event that applies to the connecting profile, as follows: - RING (ringing) - OPLSF (outpulsing failure) - NOANS (no answer) - FLASH (calling party flashed) - CLGD (calling disconnect) - ANS (answer) - TRMT (treatment) - $ (no trigger event) ***Note:*** NTFY (notify time expires) and PTYD (party disconnect) are not valid connection triggers.
	SNCONTRL	Y, N	Service node control. Enter Y or N to indicate whether or not the SN gets control of the call for the given trigger event. If set to Y, the SN has control and a Session Begin message is sent to the SN for the trigger event. If set to N, the DMS switch remains in control of the call and a Trigger Event Inform message is sent to the SN for the trigger event. ***Note:*** SNCONTRL is not used for a trigger action that leads to an operator function.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden. ***Note:*** If the SNCONTRL subfield is set to N and the call attempts to trigger to an SN and fails, the call *remains* floated even if the RFLTONFL subfield is set to Y.

Table 61 Datafilling table OACNNPRF

Field	Subfield or refinement	Entry	Explanation and action
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function. ***Note:*** Datafilling a TOPSOPER function or control list against the CLGD trigger event is not allowed.
DTMFPRF		Y, N	DTMF profile. Enter Y or N to indicate whether or not to trigger for DTMF events. If set to Y, DTMF triggers are applied to the call and a DTMF profile index in the refinement DTMFIDX must also be datafilled. If set to N, no DTMF triggers are applied to the call.
	DTMFIDX	0 to 2046	DTMF profile index. Enter the index from table OADTFPRF.
CAUSEPRF		Y, N	Cause profile. Enter Y or N to indicate whether or not to trigger for ISUP cause events. If set to Y, ISUP cause triggers are applied to the call and datafill is required in subfield CAUSEIDX. If set to N, no ISUP cause triggers are applied to the call.
	CAUSEIDX	0 to 2046	Cause profile index. Enter the index from table OACAUPRF.

OACNNPRF example

The following figure shows example datafill.

Figure 149 MAP display example for table OACNNPRF

```
CONNIDX    PROFILE                                     DTMFPRF CAUSEPRF
-----------------------------------------------------------------------
0       (NOANS Y Y FUNCTION 0_MINUS_OPER)$                    Y 7   Y 3
1 (FLASH Y N FUNCTION BRANDING) (NOANS Y Y CTRLLIST YEL_PAGE)$ N     Y 2
```

OACNNPRF error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a control list name before it is defined in table OACTLDEF
- when the user tries to datafill a function name before it is defined in table OAFUNDEF
- when the user tries to datafill a DTMF profile index before it is defined in table OADTFPRF
- when the user tries to datafill a cause profile index before it is defined in table OACAUPRF

The following table lists other possible error messages.

Table 62 Error messages for table OACNNPRF

Error message	Explanation
`Not a valid connection trigger.`	The user tries to add or change the TRIGGER subfield using an invalid trigger value. Refer to the TRIGGER subfield for valid values.
`Only one action assignment allowed per trigger.`	The user tries to add or change a tuple with more than one action assigned to the same trigger.

OADSCPRF

Table OADSCPRF associates lists of trigger events and corresponding actions with a post disconnect profile index. The disconnect profile index is then associated with a profile index in table OATPRFIX.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 63 Datafilling table OADSCPRF

Field	Subfield or refinement	Entry	Explanation and action
DISCIDX		0 to 2046	Disconnect index. Enter the profile index number.
PROFILE		See subfields	Profile. This field contains a list of trigger events and corresponding actions. It consists of subfields TRIGGER, SNCONTRL, and ACTSEL.

Table 63 Datafilling table OADSCPRF

Field	Subfield or refinement	Entry	Explanation and action
	TRIGGER	RING, OPLSF, NOANS, NTFY, FLASH, CLGD, PTYD, ANS, TRMT, $	Trigger. Enter the trigger event that applies to the disconnect profile, as follows: - FLASH (calling party flashed) - CLGD (calling disconnect) - $ (no trigger event) ***Note:*** RING (ringing), OPLSF (outpulsing failure), NOANS (no answer), NTFY (notify timer expires), PTYD (party disconnect), ANS (answer), and TRMT (treatment) are not valid disconnect triggers.
	SNCONTRL	Y, N	Service node control. Enter Y or N to indicate whether or not the SN gets control of the call for the given trigger event. If set to Y, the SN has control and a Session Begin message is sent to the SN for the trigger event. If set to N, the DMS switch remains in control of the call and a Trigger Event Inform message is sent to the SN for the trigger event. ***Note:*** SNCONTRL is not used for a trigger action that leads to an operator function.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.
DTMFPRF		Y, N	DTMF profile. Enter Y or N to indicate whether or not to trigger for DTMF events. If set to Y, ISUP cause triggers are applied to the call and datafill is required in subfield DTMFIDX. If set to N, no DTMF triggers are applied to the call.
	DTMFIDX	0 to 2046	DTMF profile index. Enter the index from table OADTFPRF.

OADSCPRF example

The following figure shows example datafill.

Figure 150 MAP display example for table OADSCPRF

```
DISCIDX          PROFILE            DTMFPRF
---------------------------------------------------------------------
0                $                  Y 4
1     (CLGD Y FUNCTION DEBIT)  (FLASH Y FUNCTION DEBIT) $   Y 5
```

OADSCPRF error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a control list name before it is defined in table OACTLDEF
- when the user tries to datafill a function name before it is defined in table OAFUNDEF
- when the user tries to datafill a DTMF profile index before it is defined in table OADTFPRF

The following table lists other possible error messages.

Table 64 Error messages for table OADSCPRF

Error message	Explanation
`Not a valid disconnect trigger.`	The user tries to add or change the TRIGGER subfield using an invalid trigger value. Refer to the TRIGGER subfield for valid values.
`Only one action assignment allowed per trigger.`	The user tries to add or change a tuple with more than one action assigned to the same trigger.

OATLKPRF

Table OATLKPRF associates lists of trigger events and corresponding actions with a talking profile index. The talking profile index is then associated with a profile index in table OATPRFIX.

Note: If the SN requires call timing information (STOPCNVT subfield) for a call just prior to a trigger event, the GENAMA option (GENAMA subfield) should not be datafilled as Y (yes) against that trigger event.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 65 Datafilling table OATLKPRF

Field	Subfield or refinement	Entry	Explanation and action
TALKIDX		0 to 2046	Talking index. Enter the profile index number.
PROFILE		See subfields	Profile. This field contains a list of trigger events and corresponding actions. It consists of subfields TRIGGER, SNCONTRL, STOPCNVT, GENAMA, RELFWD, RFLTONFL, and ACTSEL.
	TRIGGER	RING, OPLSF, NOANS, NTFY, FLASH, CLGD, PTYD, ANS, TRMT, $	Trigger. Enter the trigger event that applies to the talking profile, as follows: - NTFY (notify timer expires) - FLASH (calling party flashed) - CLGD (calling disconnect) - PTYD (party disconnect) - $ (no trigger event) ***Note:*** RING (ringing), OPLSF (outpulsing failure), NOANS (no answer), ANS (answer), and TRMT (treatment) are not valid talking triggers.
	SNCONTRL	Y, N	Service node control. Enter Y or N to indicate whether or not the SN gets control of the call for the given trigger event. If set to Y, the SN has control and a Session Begin message is sent to the SN for the trigger event. If set to N, the DMS switch remains in control of the call and a Trigger Event Inform message is sent to the SN for the trigger event. ***Note:*** SNCONTRL is not used for a trigger action that leads to an operator function.

Table 65 Datafilling table OATLKPRF

Field	Subfield or refinement	Entry	Explanation and action
	STOPCNVT	Y, N	Stop conversation. Enter Y or N to indicate whether or not to stop billable conversation timing for the call when processing the trigger event.
	GENAMA	Y, N	Generate AMA. Enter Y or N to indicate whether or not to generate an AMA record when processing the trigger event.
	RLSFWD	Y, N	Release forward party. Enter Y or N to indicate whether or not to release the forward party when processing the trigger event. ***Note:*** The RLSFWD subfield must be set to Y when the GENAMA subfield is set to Y.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden. ***Note:*** The RFLTONFL subfield may be set to Y only when the RLSFWD is set to N.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.
STAR		See subfields	Star. This field specifies the action for the DTMF star (*) digit and consists of subfields DIGITSEL, STOPCNVT, GENAMA, RLSFWD, RFLTONFL, and ACTSEL.
	DIGITSEL	Y, N	Digit selector. Enter Y or N to indicate whether or not the star digit trigger is enabled. If set to Y, datafill is required in subfields STOPCNVT, GENAMA, RLSFWD, RFLTONFL, ACTSEL, and STRPARTY.
	STOPCNVT	Y, N	Stop conversation. Enter Y or N to indicate whether or not to stop billable conversation timing for the call when processing the star digit trigger event.
	GENAMA	Y, N	Generate AMA. Enter Y or N to indicate whether or not to generate an AMA record when processing the star digit trigger event.

Table 65 Datafilling table OATLKPRF

Field	Subfield or refinement	Entry	Explanation and action
	RLSFWD	Y, N	Release forward party. Enter Y or N to indicate whether or not to release the forward party when processing the star digit trigger event. ***Note:*** The RLSFWD subfield must be set to Y when the GENAMA subfield is set to Y.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden. ***Note:*** The RFLTONFL subfield may be set to Y only when the RLSFWD is set to N.
	STRPARTY	CALLING, CALLED	Specialized tone receiver party. Enter the party to which STR supervision is applied.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.
OCTO		See subfields	Octothorpe. This field specifies the action for the octothorpe (#) digit and consists of subfields DIGITSEL, STOPCNVT, GENAMA, RLSFWD, and ACTSEL.
	DIGITSEL	Y, N	Digit selector. Enter Y or N to indicate whether or not the octothorpe digit trigger is enabled. If set to Y, datafill is required in subfields STOPCNVT, GENAMA, RLSFWD, RFLTONFL, ACTSEL, and STRPARTY.
	STOPCNVT	Y, N	Stop conversation. Enter Y or N to indicate whether or not to stop billable conversation timing for the call when processing the octothorpe digit trigger event.
	GENAMA	Y, N	Generate AMA. Enter Y or N to indicate whether or not to generate an AMA record when processing the octothorpe digit trigger event.

Table 65 Datafilling table OATLKPRF

Field	Subfield or refinement	Entry	Explanation and action
	RLSFWD	Y, N	Release forward party. Enter Y or N to indicate whether or not to release the forward party when processing the octothorpe digit trigger event. ***Note:*** The RLSFWD subfield must be set to Y when the GENAMA subfield is set to Y.
	RFLTONFL	Y, N	Refloat on failure. Enter Y or N to indicate whether or not to refloat the call upon triggering failure. If set to Y, disposition routing (in table OAFNDISP) is overridden. ***Note:*** The RFLTONFL subfield may be set to Y only when the RLSFWD is set to N.
	STRPARTY	CALLING, CALLED	Specialized tone receiver party. Enter the party to which STR supervision is applied.
	ACTSEL	CTRLLIST, FUNCTION	Action selector. Enter CTRLLIST and a control list name (from table OACTLDEF) to select a control list. Enter FUNCTION and a function name (from table OAFUNDEF) to select a function.
HOLDRCVR		Y, N	Hold receiver. Enter Y or N to indicate whether or not to retain the DTMF receiver linked to the calling party when the call transitions to an SN. ***Note:*** HOLDRCVR is not used for a trigger action that leads to an operator function.

OATLKPRF example

The following figure shows example datafill.

Figure 151 MAP display example for table OATLKPRF

```
TALKIDX      PROFILE
                      STAR                               OCTO      HOLDRCVR
-------------------------------------------------------------------
0            $
                       N                                  N         N
1   (FLASH  Y  N  N  N  Y  FUNCTION  PREPAY_NODE)(NTFY  Y  N  N  N  Y  FUNCTION
PREPAY_NODE) $        Y N Y Y N CALLING FUNCTION BRANDING    N        Y
2 (PTYD Y N N N Y CTRLLIST ROUTE_NODE)   $
                     Y N Y Y N CALLING CTRLLIST ROUTE_NODE  N         N
```

OATLKPRF error messages

A standard table control error message is displayed under the following conditions:

- when the user tries to datafill a control list name before it is defined in table OACTLDEF
- when the user tries to datafill a function name before it is defined in table OAFUNDEF. The following table lists other possible error messages.

Table 66 Error messages for table OATLKPRF

Error message	Explanation
`Not a valid talking trigger.`	The user tries to add or change the TRIGGER subfield using an invalid trigger value. Refer to the TRIGGER subfield for valid values.
`Only one action assignment allowed per trigger.`	The user tries to add or change a tuple with more than one action assigned to the same trigger.
`Only one STR digit can be datafilled.`	The user tries to datafill both star (*) and octothorpe (#) trigger digits. Only one DTMF digit event (* or #) can be specified in the same tuple.
`RLSFWD MUST BE SET TO N TO PERMIT RFLTONFL SETTING OF Y`	The user tries to set RFLTONFL to Y when RSLFWD is set to Y.
`RLSFWD must be set to Y when GENAMA is set to Y.`	The user tires to set RLSFWD to N when GENAMA is set to Y.

OATPRFIX

Table OATPRFIX combines the profile indexes from tables OACNNPRF, OADSCPRF, and OATLKPRF into a single call trigger profile index. The single trigger profile index allows for ease in assigning the initial call trigger profile index in table CT4QNAMS. It also allows the SN to update the call floated profile indexes through a single call trigger profile index in the OAP protocol.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 67 Datafilling table OATPRFIX

Field	Subfield or refinement	Entry	Explanation and action
IDX		0 to 2046	Index. Enter the profile index number.
CPROFIDX		N, Y (0 to 2046)	Connecting profile index. Enter Y or N to indicate whether or not to enable the connecting profile index. If set to Y, also datafill the connecting profile index from table OACNNPRF.

Table 67 Datafilling table OATPRFIX

Field	Subfield or refinement	Entry	Explanation and action
TPROFIDX		N, Y (0 to 2046)	Talking profile index. Enter Y or N to indicate whether or not to enable the talking profile index. If set to Y, also datafill the talking profile index from table OATLKPRF.
DPROFIDX		N, Y (0 to 2046)	Disconnect profile index. Enter Y or N to indicate whether or not to enable the disconnect profile index. If set to Y, also datafill the disconnect profile index from table OADSCPRF.

OATPRFIX example

The following figure shows example datafill.

Figure 152 MAP display example for table OATPRFIX

```
IDX             CPROFIDX     TPROFIDX     DPROFIDX
---------------------------------------------------------------
0               Y 4            Y 2            Y 7
1               Y 2            Y 2            Y 3
2               Y 1            Y 2            Y 10
3               N              Y 3            Y 8
```

OATPRFIX error messages

The following table lists possible error messages.

Table 68 Error messages for table OATPRFIX

Error message	Explanation
`Connection profile is not in table OACNNPRF.`	The user tries to datafill a connecting profile index before it is defined in table OACNNPRF.
`Talking profile is not in table OATLKPRF.`	The user tries to datafill a talking profile index before it is defined in table OATLKPRF.
`Disconnect profile is not in table OADSCPRF.`	The user tries to datafill a disconnect profile index before it is defined in table OADSCPRF.

OAINCTLA

Table OAINCTLA associates the OSSAIN control list with each CT4Q name. Each CT4Q datafilled in table OAINCTLA must first be assigned in table CT4QNAMS with the SYSTEM selector set to OSSAIN. Refer to “Direct transfers and new subscriber originations datafill” on page 325, for information on datafilling table OAINCTLA.

OSSAIN preprocessing datafill

TOPS QMS allows calls that are assigned to a TOPS operator or automated system to receive OSSAIN *preprocessing* from an SN. For example, a service such as branding an Automated Coin Toll Service (ACTS) call can be handled by an SN during OSSAIN preprocessing. OSSAIN preprocessing occurs before connecting to the operator or automated system. After a session with the SN is completed, the call resumes its course to TOPS processing.

Note: See page 82 for a list of CO types supported in OSSAIN; see Chapter 4: "OSSAIN enhancements." for a list of additional CO types allowed for OSSAIN preprocessing.

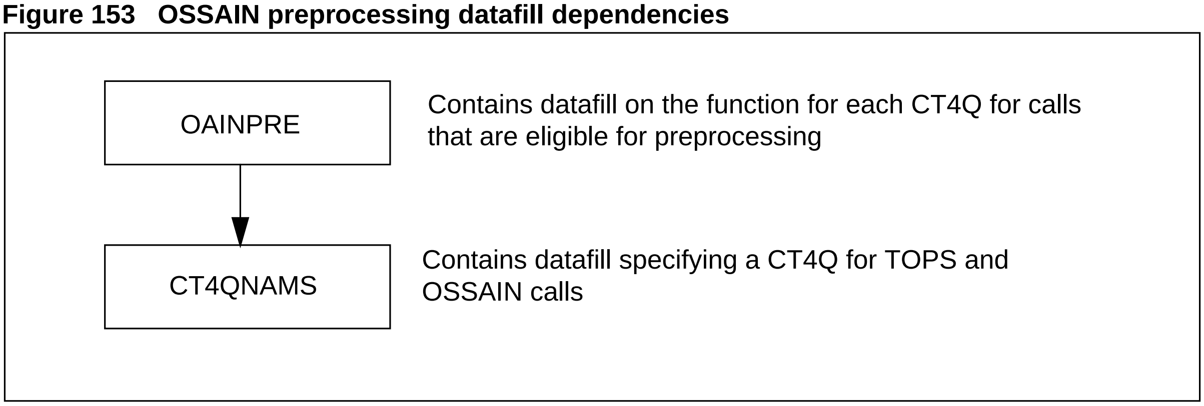
The translations tables for OSSAIN preprocessing are described in the following table. The tables are listed in the order in which they are to be datafilled.

Table 69 Datafill sequence for OSSAIN preprocessing

Table name	Explanation
CT4QNAMS	The Call Type for Queuing Names table specifies the CT4Q name and attributes.
OAINPRE	The OSSAIN Preprocessing table defines an OSSAIN function for each CT4Q name assigned to TOPS calls that receive OSSAIN preprocessing.

The following figure shows table dependencies for the two tables that require datafill for OSSAIN preprocessing.

Figure 153 OSSAIN preprocessing datafill dependencies



CT4QNAMS

The SYSAREA field in table CT4QNAMS specifies whether OSSAIN preprocessing occurs for certain TOPS calls. When the SYSTEM field is set to TOPSOPR and the OAINPRE subfield is set to Y, the CT4Q is eligible for OSSAIN preprocessing. Datafill is also required in the OPRSYS subfield, which lists the set of TOPS operator systems (OPR, MCCS, ABBS, ACTS, ADAS, ALL, NONE) that can receive OSSAIN preprocessing.

Note: The SYSAREA field also requires datafill in the CT4QDISP and AUTOORDB subfields.

CT4QNAMS example

The following figure shows sample tuples in table CT4QNAMS. In the example, calls refining to index 0 (1+COIN) can receive OSSAIN preprocessing for ACTS calls or operator calls. Calls refining to index 1 (DA_411) can receive OSSAIN preprocessing for operator calls.

Note: For complete information on table CT4QNAMS, please refer to the *Customer Data Schema Reference Manual*.

Figure 154 MAP display example for table CT4QNAMS

```
CT4QNUM CT4QNAME       NOAMA    ITRIGIDX   SYSAREA
----------------------------------------------------------------------
6       0_MINUS          N         N         TOPSOPR Y 0_MIN_Q  N  N  N
10      0_PLUS           N         N         TOPSOPR Y 0+_Q     N  N  N
620      0_PLUS_OSSAIN    N         Y 0       OSSAIN
621      0_MINUS_OSSAIN   N         Y 0       OSSAIN
```

OAINPRE

Table OAINPRE defines an OSSAIN function for each CT4Q name assigned to TOPS calls that receive OSSAIN preprocessing. Each CT4Q datafilled in table OAINPRE first must be assigned in table CT4QNAMS.

After the switch performs CT4Q refinement, a call that meets the criteria for OSSAIN preprocessing uses the final CT4Q assignment to index table OAINPRE. From OAINPRE, a function provided by an SN is determined and a session with the SN begins.

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 70 Datafilling table OAINPRE

Field	Subfield or refinement	Entry	Explanation and action
CT4Q		Alphanumeric up to 32 characters	Call type for queuing name. Enter the CT4Q name defined in table CT4QNAMS.
NETWRKID		0 to 32766	Network service ID. Enter the ID to uniquely identify the service in the network for billing.
OAFUNCNM		Alphanumeric up to 16 characters	OSSAIN function name. Enter the function name defined in table OAFUNDEF.

OAINPRE example

The following figure shows example datafill.

Figure 155 MAP display example for table OAINPRE

```
CT4Q                          NETWRKID       OAFUNCNM
-------------------------------------------------------------------
1+COIN                        3              SN_TOLL_BRAND
DA_411                        4              SN_DA_BRAND
1+HOTEL                       5              HOTEL_OAINPRE
```

OAINPRE error messages

The following table lists possible error messages.

Table 71 Error messages for table OAINPRE

Error message	Explanation
`CT4Q does not receive OSSAIN Preprocessing`	The user tries to add or change a tuple that is not datafilled in table CT4QNAMS with SYSTEM set to TOPSOPR and OAINPRE set to Y.

OSSAIN alternate routing datafill

OSSAIN alternate routing allows an SN to *override* the translations route selected by the DMS switch. This capability gives the SN more control in selecting the outgoing route for a call. With alternate routing, the actual DN of the call is *unchanged*. The DN is still used in digit manipulation (based on the new route) before it is outpulsed.

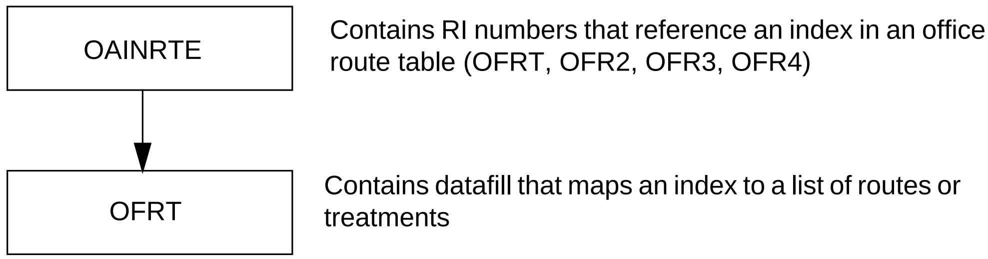
The translations tables for OSSAIN alternate routing are described in the following table. The tables are listed in the order in which they are to be datafilled.

Table 72 Datafill sequence for OSSAIN alternate routing

Table name	Explanation
OFRT	The Office Route tables (OFRT, OFR2, OFR3, and OFR4) specify office routes in the DMS switch.
OAINRTE	The OAIN Route table provides a bridge between the SN routing index (RI) numbers and the DMS switch office routes.

The following figure shows table dependencies for the two tables that require datafill for OSSAIN alternate routing.

Figure 156 OSSAIN alternate routing datafill dependencies



OFRT

Table OFRT is used to map the OFRTIDX from table OAINRTE to a list of alternate routes or treatments.

Note: If users change a tuple in table OFRx, the switch does not ensure that references in table OAINRTE (RIs) have been deleted or changed first. When the SN sends an invalid RI, the switch responds with an error message to the SN. The error message indicates the invalid routing index.

OFRT example

The following figure shows example datafill in table OFRT. In the example, office route 32 (from OAINRTE routing index 0) results in an outgoing trunk group, and office route 1000 (from OAINRTE routing index 3) results in a treatment. Both office routes are valid.

Note: For complete information on the office route tables, please refer to the *Customer Data Schema Reference Manual*.

Figure 157 MAP display example for table OFRT

```
RTE      RTELIST
---------------------------------------------------------------------
32       (N D ISUP2WIT 0 N N) $
1000     (S D VCA) $
```

OAINRTE

Table OAINRTE maps an RI (RTEIDX field) to an office route index (OFRTIDX field). The office route index references an outgoing route in table OFRT (including OFR2, OFR3, and OFR4).

The following table lists the fields, subfields or refinements, valid entries, and explanations.

Table 73 Datafilling table OAINRTE

Field	Subfield or refinement	Entry	Explanation and action
RTEIDX		0 to 2045	Route index. Enter the RI number sent in an OAP message from the SN.
RTETABID		OFRT, OFR2, OFR3, OFR4	Route table index. Enter the name of the office routing table that contains the alternate route referenced by the RI.
OFRTIDX		0 to 1023	Office route index. Enter the office route index that corresponds to the alternate route in the office routing table. This index must already be datafilled in OFRT, OFR2, OFR3, or OFR4.

OAINRTE example

The following figure shows example datafill. In the example, RI 0 maps to index 32 in table OFRT.

Figure 158 MAP display example for table OAINRTE

```
RTEIDX       RTETABID        OFRTIDX
----------------------------------------------------------------------
0            OFRT            32
3            OFRT            1000
16           OFR2            50
235          OFR2            50
```

Note: Table OAINRTE supports only OFRx tables (OFRT, OFR2, OFR3, and OFR4). OSSAIN alternate routing does not support IBN tables and IBN trunks.

OAINRTE warning messages

The following table lists possible warning messages.

Table 74 Warning messages for table OAINRTE

Warning message	Explanation
`Warning: OFRTIDX is not in routing table as RTE.`	The user tries to datafill an office route index that is not datafilled in the office routing table.

Note: If users change a tuple in table OFRx, the switch does not ensure that references in table OAINRTE (RIs) have been deleted or changed first. When the SN sends an invalid RI, the switch responds with an error message to the SN. The error message indicates the invalid routing index.

OSSAIN QMS MIS application datafill

To provide statistics for calls handled by SNs, the OSSAIN QMS MIS application requires datafill in table Queue Management System Management Information System (QMSMIS). This table also supports the TOPS QMS MIS application.

Note 1: For the OSSAIN QMS MIS application, tables OANODNAM (page 282) and OANODINV (page 284) must be datafilled before datafilling table QMSMIS.

Note 2: Table OAINPARM (page 357) contains four parameters used by the OSSAIN QMS MIS application.

QMSMIS

Table QMSMIS contains information on the QMS MIS applications (TOPS and OSSAIN) supported by the DMS switch. For the OSSAIN application, the data connectivity (DATALINK field) must be set to ETHERNET, rather than MPC (multiprotocol controller). Also, the MIS node name must be specified, along with the MIS network ID and MIS processing port.

The following table lists the fields, subfields or refinements, valid entries, and explanations. Only those fields that apply directly to OSSAIN QMS MIS are shown. (The TOPS QMS MIS application is unchanged.) For a description of the other fields, please refer to the *Customer Data Schema Reference Manual*.

Table 75 Datafilling table QMSMIS

Field	Subfield or refinement	Entry	Explanation and action
INDEX		TOPS, OSSAIN, QMSNILAP	Index. Enter the QMS MIS application name. The OSSAIN application provides MIS statistics accumulated for calls handled by SNs.
DATALINK		MPC, ETHERNET	Datalink. Enter the type of datalink over which the MIS statistics are sent. When set to ETHERNET, this field consists of subfields NODENAME, MISNETID, and MISPORT.
	NODENAME	Alphanumeric up to 12 characters	Node name. Enter the name of the MIS node defined in table OANODNAM. This node must also be datafilled in table OANODINV with an ONPMTYPE of OSNM.

Table 75 Datafilling table QMSMIS

Field	Subfield or refinement	Entry	Explanation and action
	MISNETID	0 to 32766	MIS network ID. Enter the unique identifier that is specified in the header of the MIS data stream. ***Note:*** In the OSAC environment, where multiple OSAC host nodes can be connected, the MISNETID can be used to uniquely identify the source of the MIS statistics.
	MISPORT	1024 to 32767	MIS port. Enter the port number to be used when sending an MIS class message to the given node.

QMSMIS example

The following figure shows example datafill.

Figure 159 MAP display example for table QMSMIS

```
INDEX      DATALINK
---------------------------------------------------------------------
OSSAIN     ETHERNET MISNODE  24  7001
TOPS       MPC  1  2  4
```

QMSMIS error messages

The following table lists possible error messages.

Table 76 Error messages for table QMSMIS

Error message	Explanation
`You must set DATALINK to MPC for TOPS MIS nodes.` `You must set DATALINK to ETHERNET for OSSAIN MIS nodes.`	The user tries to datafill a datalink name that does not match the index.
`The node specified must be datafilled with PMTYPE of OSNM in Table OANODINV.`	The user tries to datafill a node name that is not set to the OSNM type in table OANODINV.
`You must OFFL the new node before changing the node name.` `You must OFFL the old node before changing the node name.`	The user tries to change the name of the node without taking the node offline.
`You must OFFL this node before changing the MISPORT.`	The user tries to change the MIS port number without taking the node offline.

Table 76 Error messages for table QMSMIS

Error message	Explanation
`You must OFFL this node before deleting the tuple.`	The user tries to delete the tuple without taking the node offline.
`You must OFFL this node before adding the tuple`	The user tries to add a tuple without taking the node offline.
`The node name specified is in use in Table OASESNPL.`	The user tries to datafill a node name that is used in table OASESNPL (for session pools).

OSSAIN parameter datafill

OSSAIN call processing requires table OAINPARM. Table OAINPARM is automatically allocated during the software load build process and has store for 32 parameters.

OAINPARM parameters

Table OAINPARM consists of two fields: PARMNAME and PARMVAL. The following parameters used in OSSAIN processing are described in this section:

- ADACC_ALT_BILL_INDEX
- ALT_BILL_HANDOFF_METHOD
- CALL_SANITY_TIMER_DURATION
- DEFAULT_TREATMENT
- MAX_ALLOWED_TRANSITIONS
- MAX_NUM_TRIGGERS
- MAX_TRANSFERS_BEFORE_CONNECT
- NOANS_TIMER
- NOTIFY_TIMER
- ON_HOOK_TIMER_DURATION
- QMS_MIS_OAIN_XMIT_TIMEOUT
- QMS_MIS_OAIN_CAM_ON
- QMS_MIS_OAIN_CALL_IN_Q_THRESH
- QMS_MIS_OAIN_REMOTE_SEND
- SEND_BILLSPID_W_CLASSCHG

The following table lists the parameter name, range of values, default value, and an explanation. See “OAINPARM example” on page 361 for example datafill.

Table 77 Datafilling table OAINPARM

Parameter name	Range of values/ units	Default value	Explanation
ADACC_ALT_ BILL_INDEX	N, Y (valid control list identifier)	N	The value of this parameter indicates whether or not ADACC (Automatic Directory Assistance Call Completion) calls that require alternate billing are sent to SNs. If set to Y, a valid control list identifier must also be specified. All of the following conditions must be met when setting this parameter to Y (or else the call will be routed as if the parameter were set to N): - The control list identifier must be datafilled in table OACTLDEF before datafilling this parameter. - The function associated with the control list identifier must be served by an SN, not by an operator or automated system. - The OSSAIN SOC state must be ON. - A sufficient number of recording units must be datafilled in OFCENG parameter OSSAIN_NUM_RU.
ALT_BILL_ HANDOFF_ METHOD	AABS, OSSAIN	AABS	The value of this parameter specifies how operator handoff calls are handled for billing acceptance. If set to AABS, handoff requests will fail, since AABS is no longer supported. If set to OSSAIN, then calls are handed off to an OSSAIN SN. When OSSAIN is selected, a CT4Q name from table CT4QNAMS also must be datafilled. This CT4Q is used to route the call to the SN. The CT4Q name must be datafilled first in table CT4QNAMS along with a value of OSSAIN in the SYSTEM field.
CALL_SANITY_ TIMER_ DURATION	1 to 3600/ seconds	60	The value of this parameter is the office default for the call sanity timer duration. It is used to ensure that SN facilities and switch resources do not remain connected to a dead call.

Table 77 Datafilling table OAINPARM

Parameter name	Range of values/ units	Default value	Explanation
DEFAULT_ TREATMENT	Alphanu- meric up to 8 charac- ters from ta- ble TMTCNTL	RODR	The value of this parameter indicates the treatment name from table TMTCNTL to which the call is routed by default when the value in the MAX_ALLOWED_TRANSITIONS parameter is exceeded.
MAX_ALLOWED _TRANSITIONS	0 to 20/ transitions 1 to 10/ transitions for operator to SN calls	5	The value of this parameter indicates the maximum number of successful transitions for a call. This applies to all transitions including SN to SN, SN or Operator, Operator to SN, and SN to automated system. The number of operator to SN transitions specified by the SN or IWS is limited to a range of 0-10. If the number of operator to SN handoffs exceeds twice the value in this parm, then the parameter value takes precedence. There is a counter for each call, which is zeroed at the beginning of the call, but not when the call is floated. The counter is incremented each time a successful transfer occurs.
MAX_NUM_ TRIGGERS	0 to 100/ triggers	100	The value of this parameter indicates the number of triggers that may occur on an individual call basis.
MAX_ TRANSFERS_ BEFORE_ CONNECT	0 to 5/ transfers	2	The value of this parameter indicates the maximum number of transfers that can occur while attempting to connect a call to an SN for a particular function. Transfers in the switch can result from any of the following conditions: - a QMS deflection or overflow - a blocked function - a failure to send an OAP Session Begin message to an SN after the session for the call has already been selected
NOANS_TIMER	1 to 20/ seconds	20	The value of this parameter indicates the timing duration for answer detection during call floated trigger processing.
NOTIFY_TIMER	1 to 32,400/ seconds	180	The value of this parameter indicates the timing duration that a subscriber can talk during call floated trigger processing.

Table 77 Datafilling table OAINPARM

Parameter name	Range of values/ units	Default value	Explanation
ON_HOOK_ TIMER_ DURATION	1 to 3600/ seconds	60	The value of this parameter is the office default for the on-hook timer duration. It is used to ensure that a subscriber's line does not remain connected to a dead call.
QMS_MIS_ OAIN_XMIT_ TIMEOUT	1 to 255/ seconds	10	The value of this parameter determines the maximum amount of time it takes before an OSSAIN OAP MIS buffer is sent. This timer takes effect during periods of low traffic volume when an MIS buffer could take a significant amount of time to fill. When this timer pops, any accumulated data blocks in the buffer are sent. An information data block is always sent whether or not any queue or session data blocks have accumulated.
QMS_MIS_ OAIN_CAM_ON	N, Y	N	The value of this parameter determines whether or not OSSAIN OAP MIS data blocks are encoded, buffered, and sent to the MIS node.
QMS_MIS_ OAIN_CALL_IN_ Q_THRESH	0 to 32767	5	The value of this parameter determines the number of calls sequentially placed in queue before an OSSAIN OAP MIS data block is encoded, buffered, and sent to the MIS node. Values are from 0 to 32,767. When the value is set to 0, all calls placed in queue generate an MIS queue event data block. When the value is greater than 0, it specifies the number of calls that must be sequentially placed in queue, with no calls exiting the queue, to generate the MIS queue event data block.
QMS_MIS_ OAIN_REMOTE _SEND	N, Y	N	The value of this parameter determines whether or not OSSAIN MIS data is encoded and sent from the OSAC remote switch to the OSAC host switch.
SEND_BILLSPID _W_CLASSCHG	N, Y	N	The value of this parameter determines whether or not the SPID datablocks for the billing party will be sent with the Class Charge Success Response in OAP version 8. This parameter does not affect any other OAP versions.

OAINPARM example

The following figure shows example datafill for each OSSAIN parameter.

Figure 160 MAP display example for table OAINPARM

```
PARMNAME                          PARMVAL
---------------------------------------------------------
ADACC_ALT_BILL_INDEX              Y 7
ALT_BILL_HANDOFF_METHOD           OSSAIN AABS_SN
CALL_SANITY_TIMER_DURATION        60
DEFAULT_TREATMENT                 RODR
MAX_ALLOWED_TRANSITIONS           5
MAX_NUM_TRIGGERS                  15
MAX_TRANSFERS_BEFORE_CONNECT      2
NOANS_TIMER                       20
NOTIFY_TIMER                      180
ON_HOOK_TIMER_DURATION            60
QMS_MIS_OAIN_XMIT_TIMEOUT         15
QMS_MIS_OAIN_CAM_ON               Y
QMS_MIS_OAIN_CALL_IN_Q_THRESH     5
QMS_MIS_OAIN_REMOTE_SEND          Y
SEND_BILLSPID_W_CLASSCHG          N
```

TOPS parameter datafill

OSSAIN call processing adds two parameters to table TOPSPARM. Table TOPSPARM is automatically allocated during the software load build.

TOPSPARM parameters

Table TOPSPARM consists of two fields: PARMNAME and PARMVAL. The following parameters used in OSSAIN processing are described in this section:

- AUTO_OUTPULSE_UPON_CCV_SUCCESS
- DISPLAY_CALLED_NUMBER

The following table lists the parameter name, range of values, default value, and an explanation.

Table 78 Datafilling table TOPSPARM

Parameter name	Range of values/ units	Default value	Explanation
AUTO_ OUTPULSE_ UPON_CCV_ SUCCESS	N, Y	Y	The value of this parameter determines whether or not the DMS switch automatically outpulses the forward number when a valid calling card is entered.
DISPLAY_ CALLED_ NUMBER	NONE, MP, OPP, BOTH	NONE	The value of this parameter indicates the type of TOPS position that the DMS sends the called number to. The parameter values follow: - NONE (The called number is not sent to any TOPS position.) - MP (The called number is sent and displayed at TOPS MP positions.) - OPP (The called number is sent and displayed at OPP-compatible positions.) - BOTH (The called number is sent and displayed at both TOPS MP and OPP-compatible positions.)

TOPSPARM example

The following figure shows example datafill for each parameter.

Figure 161 MAP display example for table TOPSPARM

```
PARMNAME                           PARMVAL
---------------------------------------------------------
AUTO_OUTPULSE_UPON_CCV_SUCCESS     Y
DISPLAY_CALLED_NUMBER              BOTH
```

Chapter 8: OSSAIN software optionality control

All functionality in a product computing module load (PCL) is categorized as either base or optional. Base functionality is available for use immediately. Optional functionality is grouped into commercial units called software optionality control (SOC) options.

As a tool for managing the options in a PCL, SOC provides an interface at the MAP terminal. Users can enable or disable options, track the state of SOC options, and generate reports about SOC options.

This chapter provides a brief description of how OSSAIN implements SOC. For detailed information on how to use the SOC tool, please refer to *Software Optionality Control User's Manual*, 297-8991-901.

OSAN functional group

OSSAIN software belongs to the OSAN (OSSAIN) functional group. The OSAN functional group which is comprised of an OSSAIN base option, an OSSAIN enhancements option, and the OSSAIN sessions-based option. These have a right-to-use (RTU) setting of Y (yes) or N (no) and a state setting of ON or IDLE.

To use the OSAN functionality, table TOPSFTR must be enabled through datafill in addition to enabling the SOC order codes. However, for maintenance on the OSNM nodes (formerly OSAN0002), or OSAC and OSN nodes (formerly OSAN0003), only the SOC is required. For all other initial release functionality like call processing, the TOPSFTR tuple must be enabled in addition to SOC. For additional information on table TOPSFTR, please see page 370.

As of TOPS15, the OSAN OSSAIN functional group is organized into three OSAN option order codes: OSAN0101, OSAN0102, and OSAN0103. Former SOCs were consolidated into the current option codes:

- OSAN0101 includes former SOC codes OSAN0001, OSAN0002, OSAN0003, and OSAN004
- OSAN0102 includes former SOC codes OSAN0005, OSAN0006, and OSAN0007
- OSAN0103 includes former SOC code OSAN0100

OSAN0101: OSSAIN Base

The OSAN0101 SOC option is a state type, and requires SOC OSB00101. To use OSAN0101, its RTU setting must be Y and state setting must be ON.

OSAN0101 provides standalone OSSAIN call processing, including the following features:

- routing and queuing
- trigger processing
- transitions
- Open Automated Protocol (OAP)

Note 1: The former functional groups ENSV0014 (Operator Service AIN) and ENSV0020 (Operator Services AIN Version 3) map to OSAN0101.

Note 2: Service nodes cannot transition from OFFL (offline) to MANB (manual busy) unless OSAN0101 is ON.

OSAN0101 also provides the following features:

- centralized OSSAIN (OSAC) processing and maintenance
- simultaneous connections
- OSSAIN preprocessing of TOPS calls
- sending call timing information to a service node (SN)
- OSSAIN conferencing
- Use of XA-Core Ethernet interface for LAN connectivity
- additional enhancements, including voice link broadcasting

Monitored usage limit

With the SOC consolidation in TOPS15, the monitoring of usage limit (former SOC OSAN0002) is no longer performed.

OSAN0102: OSSAIN Enhancements

The OSAN0102 (OSSAIN Enhancements) SOC option is a state type. It is dependent upon SOC OSAN0101, and provides the following features:

- OSSAIN QMS MIS application and OAP support of QMS MIS messages
- OSSAIN country direct

  Note 1: For an OSSAIN country direct call to be routed to an SN, the TOPS country direct option, ENSV0101, also must be ON. If it is not ON, all country direct calls that attempt to terminate to the TOPS switch are converted to 0+ calls. If ENSV0101 is ON, but OSAN0102 is not ON, country direct calls routed to an SN are sent to treatment.

  Note 2: If OSAN0102 and ENSV0101 are both ON and a country direct call is routed to an SN whose OAP version is less than 5.0, the call is presented to the SN with a CO type of UNSPEC instead of CDIR.

Note 3: OSSAIN country direct is supported for TOPS trunks that support OSSAIN. It is also supported for ISUP IT and ISUP ATC trunks if SOC option OSEA0102 (ISUP to TOPS Enhancements) is ON.

- OSSAIN alternate routing
- voice connection enhancements
- disposition routing enhancements
- refloat on failure trigger enhancements
- Commercial Credit Card Sales Report
- QMS CT4Q refinements for OSSAIN calls that route to an operator

  Note: To perform CT4Q refinements based on ASST ordering, the advanced queuing option, OSB00101 (formerly ADVQ0006), also must be ON. If it is not ON and the call attempts to use ASST refinement ordering, an OAIN302 log is produced to indicate a SOC problem. The call is routed using the CT4Q from table OAFUNDEF as the final operator CT4Q for the call.
- estimate of charges

  Note: OSSAIN estimate of charges requires the option ENSV0101 (Estimate of Call Charges).
- restricts routing of calls with a CO type of SLRN to OSSAIN SNs

  Note: Calls with a CO type of SLRN are only supported over ISUP trunks and require SOC OSEA0102 (ISUP to TOPS) to correctly process the LRN.
- SOC must be on when a CO type of SLRN call is going to OSSAIN or log OAIN302 will be generated and the call will be sent to treatment
- connection to passive operator for DA Automation
- OAP release 8 data blocks
- Short Message Service (SMS) functionality - also requires TOPSFTR parameters SHORT_MESSAGE_SERVICE and OSSAIN_RELEASE_20
- wireless RLT - also requires SOC option OSEA0104 and TOPSFTR parameters RLT_FOR_IS41 and OSSAIN_RELEASE_20
- alternate hosting for passive operator requests - also requires TOPSFTR parameter OSSAIN_RELEASE_20
- OAFUNDEF TOPSAUTO DAS function - also requires TOPSFTR parameter OSSAIN_RELEASE_20
- speech path change requests during passive operator connections - also requires TOPSFTR parameter OSSAIN_RELEASE_20
- authorization code - also requires TOPSFTR parameter OSSAIN_RELEASE_20

- transfer to SN function service - also requires TOPSFTR parameter OSSAIN_RELEASE_22
- DA recall function - also requires TOPSFTR parameter OSSAIN_RELEASE_22

OSAN0102 requires OSAN0101. To use OSAN0102, its RTU setting must be Y and state setting must be ON.

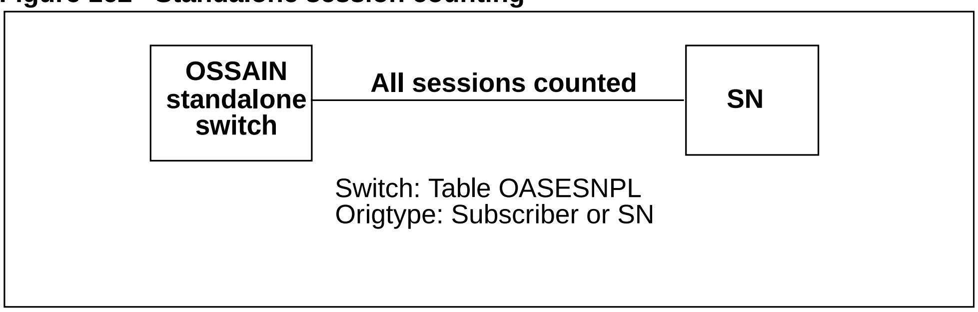
OSAN0103: OSSAIN session pricing

The OSAN0103 (formerly known as OSSAIN Session Limit) SOC option is a usage type. OSAN0103 provides a count of OSSAIN sessions.

Standalone session counting

For standalone sessions, OSAN0103 counts the maximum number of OSSAIN sessions that have an origination type of either SUBSCRIBER or SN datafilled in table OASESNPL (OSSAIN Session Pool). In the following example, all sessions are counted.

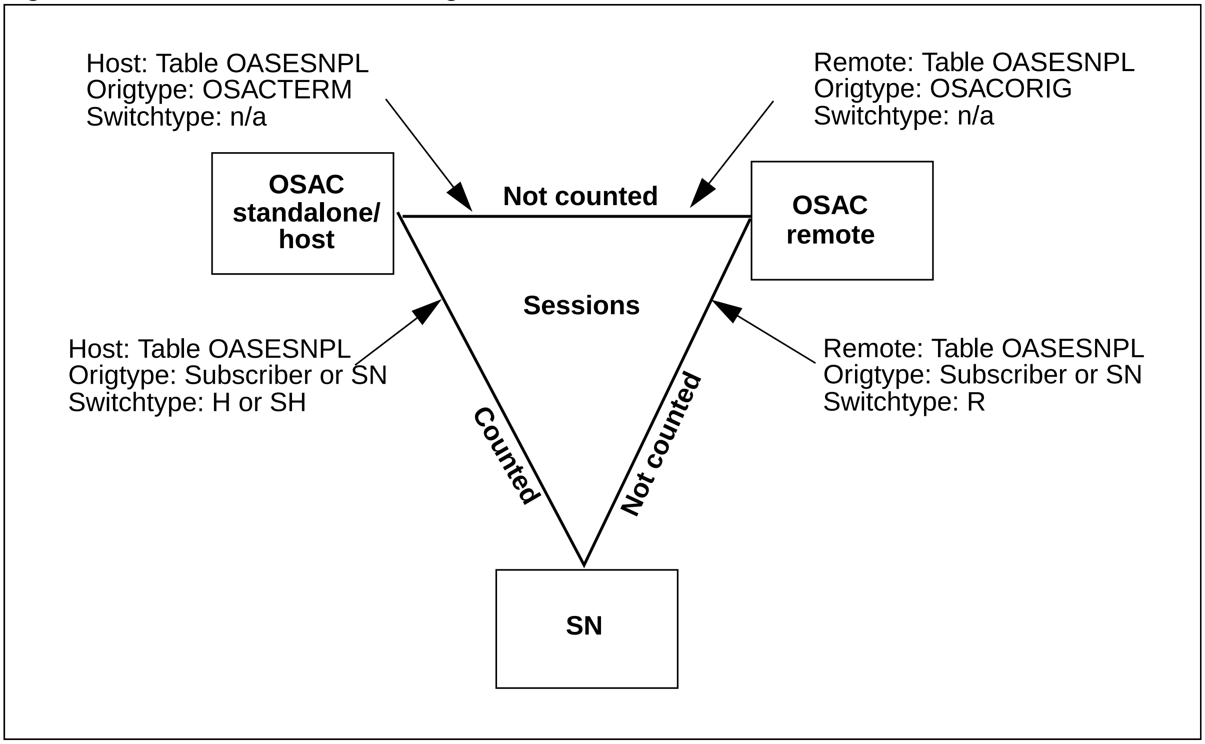
Figure 162 Standalone session counting



OSAC session counting

In the OSAC environment, OSAN0103 counts the maximum number of OSSAIN sessions that have an origination type of either SUBSCRIBER or SN datafilled in table OASESNPL and that have a switch type of H or SH. OSAC host-remote sessions are not counted and OSSAIN sessions that have a switch type of R are not counted. The following figure shows session counting in an OSAC environment.

Figure 163 OSAC session counting



Note: If the session pool is datafilled at OSAC remote, it must also have parallel datafill in table OASESNPL at the OSAC host.

OSAN SOC report example

The following figure shows an example of the three OSAN SOC codes. In the SOC tool, each option must be entered separately with the following command: SELECT OPTION OSAN<xxxx>. All the available SOC options under the group can be displayed by entering SELECT OPTION OSAN.

Figure 164 Example SOC report for OSAN functional group

```
GROUP:OSAN
OPTION    NAME                 RTU STATE   USAGE   LIMIT    UNITS  LAST_CHG
------    ----                 --- -----   -----   -----    -----  --------
OSAN0101  OSSAIN Base            Y    ON       -       -        -  01/01/29
OSAN0102  OSSAIN Enhancements    Y    ON       -       -        -  00/09/06
OSAN0103  OSSAIN Session Prici   N     -       0       0      1OS  98/03/10
```

OSSAIN SOC MAP CI commands

The user interface for SOC consists of CI commands on the MAP terminal. Two commands are described in this section, SELECT and ASSIGN.

SELECT command

The SELECT command displays a SOC report containing various information on SOC options.

OSSAIN SOC reports

The following fields appear in a SOC report:

- GROUP, which is the 3- or 4-character functional group name (OSAN)
- OPTION, which is the 8-character option order code (such as OSAN0101)
- NAME, which is the 20-character name of the option (such as OSSAIN Initial Rel)
- RTU, which specifies the right-to-use (Y or N)
- STATE, which specifies the state (IDLE or ON)
- USAGE, which specifies the current usage of the option
- LIMIT, which shows the usage limit of the option
- UNITS, which specifies the unit of usage as one message (1OA)
- LAST_CHG, which is the date of the last RTU change or limit change

SELECT OPTION

To get a brief report on the OSAN0101 option, use the following command: SELECT OPTION OSAN0101. The following figure shows an example report. In the example, the RTU setting is N.

Figure 165 Example SOC report for OSAN0101 functional group

```
GROUP:OSAN
OPTION    NAME                    RTU STATE   USAGE   LIMIT    UNITS  LAST_CHG
------    ----                    --- -----   -----   -----    -----  --------
OSAN0101  OSSAIN Base               Y    ON       -       -        -  00/11/01
```

OSSAIN SOC session limiting

OSSAIN SOC session limiting is provided by order code OSAN0103. The option is first enabled with a key code for the amount of sessions to be allowed. To assign the session limit, use the following command: ASSIGN LIMIT <limit value> <key code> TO OSAN0103.

The following figure is an example report showing a session limit of 500. The USAGE field shows a count of 200 from table OASESNPL.

Figure 166 Example SOC report showing session limit and usage count

```
GROUP:OSAN
OPTION    NAME                    RTU STATE    USAGE    LIMIT    UNITS   LAST_CHG
------    ----                    --- -----    -----    -----    -----   --------
OSAN0103  OSSAIN Session Prici  Y      -       200      500      1OS    00/11/01
```

Note: SUBSCRIBER session pools with a switch type (SWTCHTYP) of R (remote) in table OASESNPL are *not* considered in the usage of OSAN0103.

Error message for exceeding the SOC LIMIT

If the user tries to change the SOC LIMIT to a value that *exceeds* the count from OASESNPL, the following error message is displayed at the MAP.

Figure 167 Error message for exceeding the SOC LIMIT

```
OSAN0103: OAIN MAXSESNS exceed SOC limit
Warning: NO sessions may be ADDED to OASESNPL until the
OSAN0103: LIMIT is increased to greater than <# of MAXSESNS in OASESNPL>.
USAGE will be set to LIMIT.
```

With this kind of error, the USAGE field defaults to the LIMIT value, as shown in the following figure.

Figure 168 Example SOC report for usage defaulting to limit value

```
GROUP:OSAN
OPTION    NAME                    RTU STATE    USAGE    LIMIT    UNITS   LAST_CHG
------    ----                    --- -----    -----    -----    -----   --------
OSAN0103  OSSAIN Session Prici  Y      -       500      500      1OS   00/11/01
```

Error message for not enabling OSAN0101

If the user tries to assign a session limit to OSAN0103 without first assigning the ON state to OSAN0101, the following error message is displayed at the MAP.

Figure 169 Error message for not enabling OSAN0101

```
Option OSAN0101 must be turned ON in SOC
for OSAN0103 use.
OSAN0103 USAGE will be set to 0.
```

With this kind of error, the USAGE field defaults to zero, as shown in the following figure.

Figure 170 Example SOC report for usage defaulting to zero

```
GROUP:OSAN
OPTION    NAME                   RTU STATE    USAGE    LIMIT    UNITS   LAST_CHG
------    ----                   --- -----    -----    -----    -----   --------
OSAN0103  OSSAIN Session Prici    Y    -         0      500      10S   00/11/01
```

Table TOPSFTR

Table TOPSFTR allows the user to specify whether or not a feature is enabled. TOPSFTR contains tuples corresponding to the TOPS features. Each tuple contains the identification of the specific feature and its setting (i.e., whether or not it is enabled). A setting of Y in the data field column, FTRENABL, indicates that the feature is enabled and N indicates that the given feature is disabled.

Every individual TOPS SOC feature is now controlled by both its parent SOC option and its TOPSFTR entry. For a given TOPS feature to be functional, the parent SOC option must be ON and the corresponding FTRENABL setting in table TOPSFTR must be set to Y.

Warning messages are provided in the SOC procedures to caution the user of conflicting interactions. For example, if a given SOC option is changed from IDLE to ON, then for each SOC feature controlled by that SOC option, a message is displayed listing the state as specified in TOPSFTR. This enables the user to be aware of what features are enabled when the given SOC option is turned ON. Similarly, if a given SOC option is changed from ON to IDLE, a message is displayed listing all SOC features that would no longer be active.

Warning messages are also displayed if there is a conflict in the enable/disable status of a feature in TOPSFTR and the state of the SOC option controlling that feature. For example, when the user attempts to enable a SOC feature in TOPSFTR by setting FTRENABL to Y, a warning message is displayed if the SOC option controlling that feature is IDLE.

The following example shows the OSSAIN-specific tuples:

Figure 171 Map display for table TOPSFTR

```
GFTRNAME                                    FTRENABL
---------------------------------------------------------------------
OSSAIN_INITIAL_RELEASE                          N
OSSAIN_RELEASE_07                               N
OSSAIN_RELEASE_09                               Y
OSSAIN_RELEASE_10                               Y
OSSAIN_RELEASE_11                               N
OSSAIN_RELEASE_12                               N
OSSAIN_RELEASE_20                               Y
OSSAIN_RELEASE_22                               Y
```

Part 6: Billing

Part 6: Billing includes the following chapter:

Chapter 9: “OSSAIN billing,” beginning on page 375.

Chapter 9: OSSAIN billing

This chapter contains basic information on billing and automatic message accounting (AMA) recording for OSSAIN. The TOPS switch supports the following two mutually exclusive billing formats:

- Bellcore AMA Format (BAF) uses the latest Bellcore-defined structure codes, call codes, and module codes. BAF supports fixed definitions per element, but allows variable modules to be appended to a record as needed. Downstream processing for BAF must take into account this variable nature of module codes.
- TOPS Call Detail Recording (TDR) uses a less complex structure for recording TOPS billing data than the Bellcore-defined structure. TDR maintains a fixed definition for each record, which makes downstream processing less complex.

The TOPS portion of the switch can only record in one billing record format for all TOPS calls.

Custom AMA

Custom AMA recording for OSSAIN calls allows AMA information formatted by an SN to be included in AMA records in the DMS switch. The SN uses the Open Automated Protocol (OAP) to append the custom billing information. The DMS switch is unaware of the contents of the billing data.

Custom AMA recording also allows AMA information formatted by an Open Position Protocol (OPP)-compatible position to be included in AMA records in the switch.

Note: The SN also can request the switch to generate an AMA record. For details on how the SN performs AMA operations, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1. For details on how an OPP-compatible position performs AMA operations, please refer to *Open Position Protocol Specification*, Q214-1. For details on formatting custom AMA data, refer to *TOPS Custom AMA*, Q259-1.

Comparison of BAF and TDR

As the call progresses and service is provided, the switch collects and stores OSSAIN billing data to include later in the AMA record. The way the billing data is represented depends on the billing format specified—BAF or TDR.

Note: The billing format is specified using datafill in table CRSFMT (Call Record Stream Format). For more information on this datafill, please refer to *TOPS Call Detail Recording (TDR) User's Guide*, 297-8403-904.

The following table lists the OSSAIN billing data and describes how the data is represented in each format. (Refer to "OSSAIN AMA recording" on page 379 for more information on each format.)

Table 79 OSSAIN billing data comparison of BAF and TDR

OSSAIN billing data	Description	BAF format	TDR format
Service Node Elapsed Time (Minutes, Seconds, Tenths of Seconds)	The elapsed time between SN connect to SN release. This data is collected for each SN connected to the call.	Module 184, Service Provider Elapsed Time field The elapsed time for each SN is recorded because Module 184 can be appended multiple times to the AMA record. Individual SN elapsed time is recorded.	Service Node Accumulated Elapsed Time (Minutes, Seconds, Tenths of Seconds) fields The elapsed time for all SNs is added together and recorded in a single set of data fields. Individual SN elapsed time cannot be determined.
Service Node Number of Transactions	The number of messages exchanged between the switch and the SN. This data is collected individually for each SN associated with a given call.	Module 184, Transactions field The number of transactions for each SN is recorded because Module 184 can be appended multiple times to the AMA record. Individual SN transaction numbers are recorded.	Service Node Accumulated Number of Translations field The number of transactions for all SNs is added together and recorded in a single data field. Individual transaction sets cannot be attributed to an individual SN.
Service Node Data Type	The type of data contained in the custom billing data. This data type is collected for each instance of custom billing data, both large and small.	Modules 186 and 187, Service Node Data Identifier field	Not recorded

Table 79 OSSAIN billing data comparison of BAF and TDR

OSSAIN billing data	Description	BAF format	TDR format
Service Node Data (Large)	OSSAIN custom billing data that is sent from an SN or operator position. OSSAIN custom billing data is data collected by the SN or operator position and sent to the switch to be included in the billing record. The switch has no knowledge of the contents of the data contained in the data block. OSSAIN can send two different sizes, a large block which contains 140 digits and a small block which contains 40 digits. This data field will collect only the large block.	Module 187, Service Node Data - Large field Each large custom billing data is recorded because Module 187 can be appended multiple times to the AMA record.	Service Node Data, Large field Each large custom billing data is recorded in separate TDR records that are linked to the primary record using a sequence number.
Service Node Data (Small)	OSSAIN custom billing data that is sent from an SN or operator position. OSSAIN custom billing data is data collected by the SN or operator position and sent to the switch to be included in the billing record. The switch has no knowledge of the contents of the data contained in the data block. OSSAIN can send two different sizes, a large block that contains 140 digits and a small block that contains 40 digits. This data field will collect only the small block.	Module 186, Service Node Data - Small field Each small custom billing data is recorded because Module 186 can be appended multiple times to the AMA record.	Service Node Data, Small field Each small custom billing data is recorded in separate TDR records that are linked to the primary record using a sequence number.

Table 79 OSSAIN billing data comparison of BAF and TDR

OSSAIN billing data	Description	BAF format	TDR format
Service Node Identifier, Custom Billing	An identifier for the SN that produced the data recorded in either the Service Node Data, Large data field or the Service Node Data, Small data field. This data is collected for each instance of a Service Node Data field that is also collected.	Modules 186 and 187, Service Node ID field Each SN identifier that produced custom billing is recorded because Module 186 and 187 can be appended multiple times to the AMA record.	Service Node Identifier, Custom Billing field Each SN identifier that produced custom billing is recorded in separate TDR records that are linked to the primary record using a sequence number.
Service Node Identifier	An identifier for each SN that is connected to the call.	Module 184, Service Node ID field An identifier for each SN is recorded because Module 184 can be appended multiple times to the AMA record. Individual SN identifiers are recorded.	Service Node Identifier, Last field Only the identifier for the last SN involved in the call is recorded.
Service Node Network Service Identifier	An identifier for the service performed by an SN. Two different types of services can be provided on the OSSAIN platform: - a service that uses the full capabilities of the OSSAIN platform - a service that uses only a limited set of OSSAIN platform capabilities	Module 185, Network Service ID field Only one network service ID is collected and recorded for the call. ***Note:*** See note on page 381 for restrictions.	Service Node Network Service Identifier field Only one network service ID is collected and recorded for the call.
Service Node Sequence Number	A sequence number that is collected for each SN involved in the call. It provides an ordered list of SNs.	Module 184, SN Sequence Number field A sequence number for each SN is recorded because Module 184 can be appended multiple time to the AMA record.	Service Node Number of Nodes field Because each individual SN identifier is not recorded, TDR converts the sequence number of the last SN involved in the call into a count of the number of SNs involved in the call.

OSSAIN AMA recording

This section discusses both the BAF and TDR AMA recording formats for OSSAIN.

BAF AMA records

BAF elements consist of structure codes, call codes, and module codes. For complete details on BAF, please refer to *Bellcore Format Automatic Message Accounting Reference Guide,* 297-1001-830.

Structure codes

A structure code is an identifier that defines a set of data fields in an AMA record and determines the ordering of the fields in that record. OSSAIN calls use the following TOPS structure codes:

- 0106—for Open Numbering Plan Environment (ONPE) calls originating from a subscriber (these calls arrive on TOPS trunks)
- 0107—for ONPE calls not originating from a subscriber (these calls arrive on IT trunks)
- 0751—for calls originating on an intertoll (IT) or Access Tandem to Carrier (ATC) trunk from a corporate entity using the North American dialing plan
- 0752—for calls originating on a Remote Operator Number Identification (RONI) trunk or TOPS trunk directly from an end office using the North American dialing plan
- 0772—for calls that launch an originating line number screening (OLNS) database query

Call codes

A call code is a call type descriptor. It defines the type of call or statistic being recorded. OSSAIN calls use the following set of TOPS call codes:

- call code 189—originating with credit recording service
- call code 190—originating with carrier identification, but no service processing
- call code 191—terminating with carrier identification, but no service processing
- call code 192—originating with call completion service
- call code 193—terminating with call completion service
- call code 194—originating with listing services service
- call code 195—terminating with listing services service
- call code 196—originating with general assistance service
- call code 197—terminating with general assistance service
- call code 198—originating with busy line verification service

- call code 199—terminating with busy line verification service
- call code 215—intercept service

Module codes

A module code is an identifier that defines a set of additional data fields that can be appended to the base AMA record. OSSAIN defines AMA module codes to record call information. In addition to the existing TOPS module codes, OSSAIN calls use the following module codes:

- module 184—Service Node ID
- module 185—Service ID
- module 186—Service Node Formatted - Small
- module 187—Service Node Formatted - Large
- module 305—Authorization Code - commercial credit cards
- module 103—Extended Account Code

Detailed descriptions of these modules and examples of AMA records follow.

Module 184—Service Node ID

The Service Node ID module records information pertinent to the processing involved with an SN. This information includes the SN identifier, the order in which the SN is connected to the call, the elapsed time of the session, and the number of message transactions.

Module 184 is appended for each SN involved in a call. The maximum number of SNs allowed in a call is determined by the MAX_ALLOWED_TRANSITIONS parameter in table OAINPARM.

Note: Refer to Chapter 7: “OSSAIN data schema,” for details on table OAINPARM.

The following table shows the information, field numbers, and number of characters for module 184.

Table 80 Module 184—Service Node ID

Information	Field number	Number of characters
Module Code	88	4
Service Node ID	105	6
SN Sequence Number	140	6
Service Provider Elapsed Time	403	6
Transactions	130	6

The following figure shows an example AMA record with module 184 appended (in bold). In the example, the SERV CODE field records the Service Node ID. (The Service Node ID is datafilled in the NODEID field in table OANODINV.) The SMALL INTERVAL ELAPSED TIME field records the Service Provider Elapsed Time. The TRANSACTIONS field records the number of outgoing and incoming OAP messages to and from the service node.

Note: Refer to Chapter 7: “OSSAIN data schema,” for details on table OANODINV.

Figure 172 Example AMA record for module 184

```
HEX ID:AA STRUCTURE CODE:40752C CALL CODE:192C SENSOR
TYPE:036C SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC
OFFICE ID:0123456C DATE:40707C TIMING IND:00000C STUDY
IND:0200000C SERVICE OBSERVED:0C ORIG NPA:629C ORIG
NUMBER:5201234C CONNECT TIME:1237548C ELAPSED
TIME:000000036C OPERATOR IDS:0009999009999C ACC
OPERATOR WORK TIME:00000C SERVICE FEATURE:000C STATION
SIGNALING IND:2C SCREENING CODE:000C CALLED NUMBER
INPUT:1C CALLING NUMBER SOURCE:1C MODULE CODE:051C
OVERSEAS IND:0C TERM NPA:00201C TERM NUMBER:2201234C
COMPLETION IND:001C RATE IND:1C OSS CALL COMPLETION
CONDITIONS:3111860C MODULE CODE:185C SERV CODE:00033C
MODULE CODE:184C SERV CODE:00003C SN SEQUENCE
NUMBER:00001C SMALL INTERVAL ELAPSED TIME:03145C
TRANSACTIONS:00007C MODULE CODE:184C SERV CODE: 00010C
SN SEQUENCE NUMBER:00002C SMALL INTERVAL ELAPSED
TIME:12345C TRANSACTIONS:00002C MODULE CODE:000C
```

Module 185—Service ID

The Service ID module records the Network Service ID of the OSSAIN service provided. The module is included in all AMA records for calls using an OSSAIN service.

Note: The trigger profile index assigned to a call does not require an OSSAIN CT4Q and the associated control list, function, and network service id. Therefore, the network service id recorded by the module will not be available for those TOPS calls not utilizing an OSSAIN control list. The operating companies that change from routing TOPS calls through an OSSAIN CT4Q should prepare their downstream processes for this change in AMA.

Module 185 is not replicative; that is, there is only one instance of this module for a record. The Network Service ID is datafilled in the NETWRKID field in table OACTLDEF.

Note: Refer to Chapter 7: “OSSAIN data schema,” for details on table OACTLDEF.

The following table shows the information, field numbers, and number of characters for module 185.

Table 81 Module 185—Service ID

Information	Field number	Number of characters
Module Code	88	4
Network Service ID	105	6

The following figure shows an example AMA record with module 185 appended (in bold). In the example, the SERV CODE field records the Network Service ID.

Figure 173 Example AMA record for module 185

```
HEX ID:AA STRUCTURE CODE:40752C CALL CODE:192C SENSOR
TYPE:036C SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC
OFFICE ID:0123456C DATE:40707C TIMING IND:00000C STUDY
IND:0200000C SERVICE OBSERVED:0C ORIG NPA:629C ORIG
NUMBER:5201234C CONNECT TIME:1237548C ELAPSED
TIME:000000036C OPERATOR IDS:0009999009999C ACC
OPERATOR WORK TIME:00000C SERVICE FEATURE:000C STATION
SIGNALING IND:2C SCREENING CODE:000C CALLED NUMBER
INPUT:1C CALLING NUMBER SOURCE:1C MODULE CODE:051C
OVERSEAS IND:0C TERM NPA:00201C TERM NUMBER:2201234C
COMPLETION IND:001C RATE IND:1C OSS CALL COMPLETION
CONDITIONS:3111860C MODULE CODE:185C SERV CODE:00033C
MODULE CODE:184C SERV CODE:00003C SN SEQUENCE
NUMBER:00011C SMALL INTERVAL ELAPSED TIME:03145C
TRANSACTIONS:00007C MODULE CODE:000C
```

Module 186—Service Node Formatted - Small

The Service Node Formatted - Small module records custom billing data from an SN or an OPP-compatible position. Using OAP or OPP, the SN or OPP-compatible position can request the switch to append its formatted billing data to the billing record. The DMS switch is unaware of the contents of the billing data. The knowledge of the data is in the SN and the downstream processor.

Module 186 can be appended to an AMA record multiple times, but there is an 800-byte limit on the total number of bytes that the custom AMA modules (186 and 187) can use.

Note: Two characters comprise a byte. Each module 186 requires 26 bytes (52 characters).

The following table shows the information, field numbers, and number of characters for module 186.

Table 82 Module 186—Service Node Formatted - Small

Information	Field number	Number of characters
Module Code	88	4
Service Node ID	105	6
Service Node Data Identifier	613	2
Service Node Data - Small	614	40

Field 613

Field number 613, Service Node Data Identifier, is described in the following table.

Table 83 Field 613, Service Node Data Identifier

Characters	Meaning
1	0 = Not used 1 = BCD 2 = EBCDIC 3 = Hexadecimal 4 = ASCII 5-9 = Reserved
2	Sign (Hex C)
Note: For OSSAIN, the Service Node Data Identifier is always set to 3 (hexadecimal).	

Field 614

Field number 614, Service Node Data - Small, is described in the following table.

Table 84 Field 614 Service Node Data - Small

Characters	Meaning
40	Data from service node

The following figure shows an example AMA record with module 186 appended (in bold). In the example, the SERVICE NODE DATA IDENTIFIER field specifies the data format of the SERVICE NODE DATA SMALL field. (Hexadecimal is the only data format supported at this time.) The SERV CODE field records the Service Node ID.

Note: The size of the SERVICE NODE DATA SMALL field is 20 bytes. For custom billing data having less than 20 bytes, the data is padded to the right with #Fs to fill the 20-byte field.

Figure 174 Example AMA record for module 186

```
HEX ID:AA STRUCTURE CODE:40752C CALL CODE:192C SENSOR
TYPE:036C SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC
OFFICE ID:0123456C DATE:40707C TIMING IND:00000C STUDY
IND:0200000C SERVICE OBSERVED:0C ORIG NPA:629C ORIG
NUMBER:5201234C CONNECT TIME:1237548C ELAPSED
TIME:000000036C OPERATOR IDS:0009999009999C ACC
OPERATOR WORK TIME:00000C SERVICE FEATURE:000C STATION
SIGNALING IND:2C SCREENING CODE:000C CALLED NUMBER
INPUT:1C CALLING NUMBER SOURCE:1C MODULE CODE:051C
OVERSEAS IND:0C TERM NPA:00201C TERM NUMBER:2201234C
COMPLETION IND:001C RATE IND:1C OSS CALL COMPLETION
CONDITIONS:3111860C MODULE CODE:185C SERV CODE:00033C
MODULE CODE:184C SERV CODE:00003C SN SEQUENCE
NUMBER:00001C SMALL INTERVAL ELAPSED TIME:03145C
TRANSACTIONS:00007C MODULE CODE:186C SERV CODE:00010C
SERVICE NODE DATA IDENTIFIER:3C SERVICE NODE DATA SMALL:
000048454C5021204920414D2054524150454421 MODULE
CODE:000C
```

Module 187—Service Node Formatted - Large

The Service Node Formatted - Large module records custom billing data from an SN or an OPP-compatible position. Using OAP or OPP, the SN or OPP-compatible position can request the switch to append its formatted billing data to the billing record. The DMS switch is unaware of the contents of the billing data. The knowledge of the data is in the SN and the downstream processor.

Module 187 can be appended to an AMA record multiple times, but there is an 800-byte limit on the total number of bytes that the custom AMA modules (186 and 187) can use.

Note: Two characters comprise a byte. Each module 187 requires 76 bytes (152 characters).

The following table shows the information, field numbers, and number of characters for module 187.

Table 85 Module 187—Service Node Formatted - Large

Information	Field number	Number of characters
Module Code	88	4
Service Node ID	105	6
Service Node Data Identifier	613	2
Service Node Data - Large	615	140

Field 615

Field number 615, Service Node Data - Large, is described in the following table.

Table 86 Field 615, Service Node Data - Large

Characters	Meaning
140	Data from service node

The following figure shows an example AMA record with module 187 appended (in bold). In the example, the SERVICE NODE DATA IDENTIFIER field specifies the data format of the SERVICE NODE DATA LARGE field. (Hexadecimal is the only data format supported at this time.) The SERV CODE field records the Service Node ID.

Note: The size of the SERVICE NODE DATA LARGE field is 70 bytes. For custom billing data having less than 70 bytes, the data is padded to the right with #Fs to fill the 70-byte field.

Figure 175 Example AMA record for module 187

HEX ID:AA STRUCTURE CODE:40752C CALL CODE:192C SENSOR
TYPE:036C SENSOR ID:0123456C REC OFFICE TYPE:036C REC
OFFICE ID:0123456C DATE:40707C TIMING IND:00000C STUDY
IND:0200000C SERVICE OBSERVED:0C ORIG NPA:629C ORIG
NUMBER:5201234C CONNECT TIME:1237548C ELAPSED
TIME:000000036C OPERATOR IDS:0009999009999C ACC
OPERATOR WORK TIME:00000C SERVICE FEATURE:000C STATION
SIGNALING IND:2C SCREENING CODE:000C CALLED NUMBER
INPUT:1C CALLING NUMBER SOURCE:1C MODULE CODE:051C
OVERSEAS IND:0C TERM NPA:00201C TERM NUMBER:2201234C
COMPLETION IND:001C RATE IND:1C OSS CALL COMPLETION
CONDITIONS:3111860C MODULE CODE:185C SERV CODE:00033C
MODULE CODE:184C SERV CODE:00003C SN SEQUENCE
NUMBER:00001C SMALL INTERVAL ELAPSED TIME:03145C
TRANSACTIONS:00007C **MODULE CODE:187C SERV CODE:00010C**
SERVICE NODE DATA IDENTIFIER:3C SERVICE NODE DATA
LARGE:0000005748592041524520594F55204445434F44494E4720
544849532044415441 3F204745542041204C494645212020
4C495645204C4F4E4720414E442050524F5350455221 MODULE
CODE:000C

Module 305—Authorization Code - commercial credit card

Module code 305 is supported for generating alphanumeric authorization codes for commercial credit card queries.

The following table provides information for module code 305.

Table 5 Module code 305

Information	Field number	Number of characters
Significant Digits Next Field	55	4 BCDs
Account / Authorization Code	324	15 EBCDIC characters

OSSAIN calls using commercial credit card authorization codes will append module code 305. The module consists of the following fields:

- SIGNIFICANT DIGITS NEXT FIELD (field 55) consists of 4 BCD characters. Characters 1-3 are set to the number of EBCDIC characters in the ACCOUNT / AUTHORIZATION CODE field. Character 4 is the sign character and is set to hex C.
- ACCOUNT / AUTHORIZATION CODE (field 324) contains 1-15 EBCDIC characters. The field is left justified and padded with blanks. It contains the authorization code for commercial credit cards.

Figure 176 Example AMA record for module 305

```
HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:192C  SENSOR
TYPE:036C  SENSOR ID:0619351C  REC OFFICE TYPE:036C  REC
OFFICE ID:0619351C  DATE:40225C  TIMING IND:00000C
STUDY IND:0200000C  SERVICE OBSERVED:0C  ORIG NPA:620C
ORIG NUMBER:2619999C  CONNECT TIME:0815302C  ELAPSED
TIME:000000000C  OPERATOR IDS:0009999009999C  ACC
OPERATOR WORK TIME:00000C  SERVICE FEATURE:000C  STATION
SIGNALING IND:2C  SCREENING CODE:000C  CALLED NUMBER
INPUT:1C CALLING NUMBER SOURCE:1C  MODULE CODE:316C
LOCAL INDICATOR:1C MODULE CODE:311C  ORIG CALL TYPE:002C
MODULE CODE:051C  OVERSEAS IND:0C  TERM NPA:00231C  TERM
NUMBER:2619999C  COMPLETION IND:008C  RATE IND:1C  OSS
CALL COMPLETION CONDITIONS:3111110C  MODULE CODE:305C
SIG DIGITS NEXT FIELD:006C  COMMERCIAL CREDIT CARD
AUTHCODE:F1F2E9D7F9F840404040404040404040
MODULE CODE:184C  SERV CODE:00001C  SN SEQUENCE
NUMBER:00001C SMALL INTERVAL ELAPSED TIME:00346C
TRANSACTIONS:00005C  MODULE CODE:185C SERV CODE:00001C
MODULE CODE:104C  TRUNK FACILITY ID:104630000C MODULE
CODE:000C
```

Module 103—Extended Account Code

Module code 103 is supported for generating account codes which are greater than 4 digits (module code 301 supports 4 digit account codes via the tops account code functionality).

The following table provides information for module code 103.

Table 6 Module code 103

Information	Field number	Number of characters
Significant Digits Next Field	55	4 BCDs
Account Code	126	16 characters

OSSAIN calls using extended account codes will append module code 103 if AMAOPTS parameter CDAR is ON. The module consists of the following fields:

- SIGNIFICANT DIGITS NEXT FIELD (field 55) consists of 4 BCD characters. Characters 1-3 are set to the number of digits in the ACCOUNT CODE field. Character 4 is the sign character and is set to hex C.
- ACCOUNT CODE (field 126) contains 1-14 digits. It contains the extended account code.

Figure 177 Example AMA record for module 103

```
*HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:192C
SENSOR TYPE:036C SENSOR ID:0619351C  REC OFFICE
TYPE:036C  REC OFFICE ID:0619351C DATE:40420C  TIMING
IND:00000C  STUDY IND:0200000C  SERVICE OBSERVED:0C ORIG
NPA:620C  ORIG NUMBER:2615555C  CONNECT TIME:1219524C
ELAPSED TIME:000000000C  OPERATOR IDS:00099990099999C
ACC OPERATOR WORK TIME:00000C  SERVICE FEATURE:000C
STATION SIGNALING IND:2C  SCREENING CODE:000C  CALLED
NUMBER INPUT:1C CALLING NUMBER SOURCE:1C  MODULE
CODE:316C  LOCAL INDICATOR:1C MODULE CODE:311C  ORIG
CALL TYPE:002C  MODULE CODE:051C  OVERSEAS IND:0C TERM
NPA:00201C  TERM NUMBER:2208888C  COMPLETION IND:008C
RATE IND:1C OSS CALL COMPLETION CONDITIONS:3111110C
MODULE CODE:720C PARTY IDENTIFIER:002C  LOCATION ROUTING
NUMBER:FFFFFFFFFFFF SERVICE PROVIDER
IDENTITY:FFFFFFFFFF  LOCATION:FFFFFFFFFFFFFFFF
SUPPORTING INFORMATION:1010000C  MODULE CODE:184C  SERV
CODE:00102C SN SEQUENCE NUMBER:00001C  SMALL INTERVAL
ELAPSED TIME:02025C TRANSACTIONS:00005C  MODULE
CODE:185C  SERV CODE:01234C MODULE CODE:103C  SIG DIGITS
NEXT FIELD:010C  ACCT CODE:000001234567890C  MODULE
CODE:104C TRUNK FACILITY ID:104670000C MODULE CODE:000C
```

TDR AMA records

The following table summarizes the TDR data fields that apply to OSSAIN service nodes. It lists the field names, field type, range, and associated templates. For complete details on TDR, refer to *TOPS Call Detail Recording (TDR) User's Guide*, 297-8403-904.

Table 87 TDR data fields for SN data

TDR data field name	AMADUMP/ CALLDUMP field name	Field type	Range	Associated templates
Service Node Accumulated Elapsed Time (Minutes, Seconds, Tenths of Seconds)	SNELAPTIMEMIN, SNELAPTIMESEC SNELAPTIME10TH	Bits	0 to 59 (minutes, seconds) 0 to 9 (tenths of seconds	BLV / interrupt template Call completion template Call transfer to carrier template Charge adjust template Combined template General assistance template IN interworking template Intercept template Listing services template
Service Node Accumulated Number of Transactions	SN#TRANS	Bits	0 to 65535	BLV / interrupt template Call completion template Call transfer to carrier template Charge adjust template Combined template General assistance template IN interworking template Intercept template Listing services template
Service Node Data, Large	SNDATALRG	Digits	0 to 139 (large)	OSSAIN custom billing template
Service Node Data, Small	n/a	Digits	0 to 39 (small)	None
Service Node Identifier, Custom Billing	SNIDCUST	Bits	0 to 65535	OSSAIN custom billing template

Table 87 TDR data fields for SN data

TDR data field name	AMADUMP/ CALLDUMP field name	Field type	Range	Associated templates
Service Node Identifier, Last	SNIDLAST	Bits	0 to 65535	BLV / interrupt template Call completion template Call transfer to carrier template Charge adjust template Combined template General assistance template IN interworking template Intercept template Listing services template
Service Node Network Service Identifier	SNNETID	Bits	0 to 65535	BLV / interrupt template Call completion template Call transfer to carrier template Charge adjust template Combined template General assistance template IN interworking template Intercept template Listing services template

Table 87 TDR data fields for SN data

TDR data field name	AMADUMP/ CALLDUMP field name	Field type	Range	Associated templates
Service Node Number of Nodes	SN#NODES	Bits	0 to 65535	BLV / interrupt template Call completion template Call transfer to carrier template Charge adjust template Combined template General assistance template IN interworking template Intercept template Listing services template
Commercial Credit Card Authcode	CCCAUTHCODE	EBCDIC chars	0-15	BLV / interrupt template Call completion template Charge adjust template Combined template General assistance template Listing services template
Account Code / Authorization Code Number	ACCCODE	Digits	0-14	BLV / interrupt template Call completion template Charge adjust template Combined template General assistance template Listing services template

Figure 178 shows a TDR AMA record using the Call Completion template, followed by the OSSAIN Custom Billing template.

Figure 178 Example TDR AMA records for OSSAIN

```
*RECCODE:F0  TMPLTVER:00  TMPLTID:001  ACTTMPLID:0  RECLENGTH:068
 WORDLAYOUTIND:0  ASSOCTDR:1  TOOLGEN:0  SEQNUM:00003
 ORIGNUM:00000000006193208655  LRNCLG:0000000000  SPIDCLG:
 OLNSSERVEQP:000  INTRKGRP:0136  CLGSRC:1  INTRKMEM:0001  SERVOBS:0
 TRAFSAMP:0  DATEYR:9  DATEMO:01  DATEDAY:29  SERVFEAT:0  TIMEMIN:04
 TIMESEC:32 TIME10TH:9 TIMEHR:14 SCRNCODE:000 SUBBILLIND:0 ORIGNUMIND:1
 ELAPTIMEMIN:00000  ELAPTIMESEC:30  ELAPTIME10TH:0  OPRWKTIMEMIN:00
 OPRWKTIMESEC:00 OPRWKTIME10TH:0 OPER#LAST:9999 ACVAL:0 OPERTEAMLAST:000
 ACCODE:00000000000000  BILLTYPID:0  CALLCRDFMT:0  CALLCRDSEQCNT:000
 ALTBILLNUM:00000000000000000000000  LIDBRESP:00  CALLCRDSUBACT:00
 LRNBILL:0000000000  SPIDBILL:     RAO:000  OSSACTION:00  MULTFACT:00001
 AMTCHARGE:00000  AMTDEPOSIT:00000  CHARGEIND:0  COINCRDT:0  RATEIND:0
 HOTELRM:        HOTELNM:      ICSRC:00  ICAGREEMNT:0  IC:0000
 SNELAPTIMEMIN:01  SNELAPTIMESEC:38  SNELAPTIME10TH:5  SNIDLAST:00079
 SNNETID:08655 SN#TRANS:00010 SN#NODES:00001 TERMNUM:00000000002012208655
 LRNCLD:0000000000  SPIDCLD:     OSSCCSCECI:1  OSSCCSCSTI:1  OSSCCSCRI:1
 OUTTRKGRP:0140  OSSCCSCATI:2  OUTTRKMEM:0001  OSSCCSCNPI:1  TERMNUMIND:1
 PERSONIND:0  OVSNPAIND:1  COMPLIND:1  ICCALLEVNTSTAT:00  CCDATEDAY:00
 CCDATEMO:00  CCDATEYR:0  CCTIMEHR:00  CCTIMEMIN:00  CCTIMESEC:00
 CCTIME10TH:0  CCELAPTIMEMIN:00000  CCELAPTIMESEC:00  CCELAPTIME10TH:0


*RECCODE:F0  TMPLTVER:00  TMPLTID:007  ACTTMPLID:0  RECLENGTH:068
 WORDLAYOUTIND:0  ASSOCTDR:1  TOOLGEN:0  SEQNUM:00003  SNIDCUST:00079
 SNDATALRG:000000000000000000000000000000000000000000000000000000000000000
 000000000000000000000000000000000000012345678ABCD1234567890ABCDEF123456FF
 FFFF
```

TDR fixed templates

TDR currently provides the following ten fixed call templates:

- BLV / interrupt template
- Call completion template
- Call transfer to carrier template
- Charge adjust template
- Combined template
- General assistance template
- IN interworking template
- Intercept template
- Listing services template
- OSSAIN custom billing template

For details on template headers, template versioning, template padding and truncation, and definitions for single-fixed and multi-fixed templates, refer to *TOPS Call Detail Recording (TDR) User's Guide*, 297-8403-904.

TOPS Charge Adjust service

Custom AMA data can be appended to the AMA record for the TOPS Charge Adjust service. When the SN wants to append AMA with the TOPS Charge Adjust AMA record, the SN must send the custom AMA data *prior to* transferring to an OPP-compatible position that provides the service.

When the OPP-compatible position wants to append AMA with the TOPS Charge Adjust AMA record, the position must send the custom AMA data *prior to* providing the service.

Note: The OPP-compatible position can provide a maximum of 62 Charge Adjust services unless it requests a GEN AMA before the limit is reached.

NO AMA

Using an OAP message, an SN can mark a call as NO AMA. This allows a call or portion of a call to be marked not billable in the AMA record generated by the switch. Once the SN or operator marks a call as NO AMA, the call remains in this state until the SN or operator marks it as requiring billing.

Part 7: OA&M

Part 7: Operation, administration, and maintenance includes the following chapters:

Chapter 10: “OSSAIN maintenance,” beginning on page 395.

Chapter 11: “OSSAIN logs,” beginning on page 453.

Chapter 12: “OSSAIN operational measurements,” beginning on page 517.

Chapter 10: OSSAIN maintenance

This chapter discusses the maintenance activities and user interface for OSSAIN. Maintenance topics include automated processes; maintenance states; related datafill, logs, and OMs; MAP commands; and alarms.

The chapter is organized into the following four broad areas of OSSAIN maintenance:

- *OSSAIN node maintenance* for the following three node types:
  - OSNM (Operator Services Node Maintained)
  - OSN (Operator Services Node)
  - OSAC (Centralized OSSAIN Node)
- *OSSAIN session pool maintenance* for session pools supported by the three node types.
- *Additional utilities* for the following three non-menu commands:
  - OSSAINCI
  - QCALL
  - QVIEW
- *OSSAIN alarms* that the switch generates for OSSAIN nodes and session pools.

OSSAIN node maintenance

The user interface for OSSAIN node maintenance resembles the interface for peripheral module (PM)-type applications. Maintenance activities are similar among each of the three node types. This section discusses common maintenance as well as maintenance that varies depending on the node type.

Defining nodes in table OANODINV

All OSSAIN node types (SNs and switches) must be datafilled in table OANODINV (OSSAIN Node Inventory) before any maintenance actions can be performed. For details on all the fields and values, refer to Chapter 7: “OSSAIN data schema.”

The SN is datafilled as an *OSNM node type* when it has a maintenance relationship with the switch where it is datafilled. This relationship can exist in either a standalone OSSAIN network (one switch) or in an OSAC network (host switch and remote switch or switches).

Each switch in the OSAC network is datafilled as an *OSAC node type*. In an OSAC network, the OSAC nodes communicate to perform centralized node maintenance and to facilitate call processing in the OSAC environment.

A centralized SN is datafilled as an OSNM node type at the OSAC host switch and as an *OSN node type* at the OSAC remote switch. An OSN node does *not* have a maintenance relationship with the OSAC remote switch. However, the OSAC remote switch does monitor the state of the data connection with the OSN.

Note: An MIS node is datafilled as an OSNM node type in table OANODINV. For details on MIS node maintenance, refer to Chapter 4: “OSSAIN enhancements,” on page 194.

Automated audit process

The switch performs an automated audit process, which verifies that the node is in service (INSV) or in service trouble (ISTB). In the OSAC environment, the audit also verifies parallel datafill for the session pool on the node.

Datafill in three fields in table OANODINV controls the automated audit process, as follows:

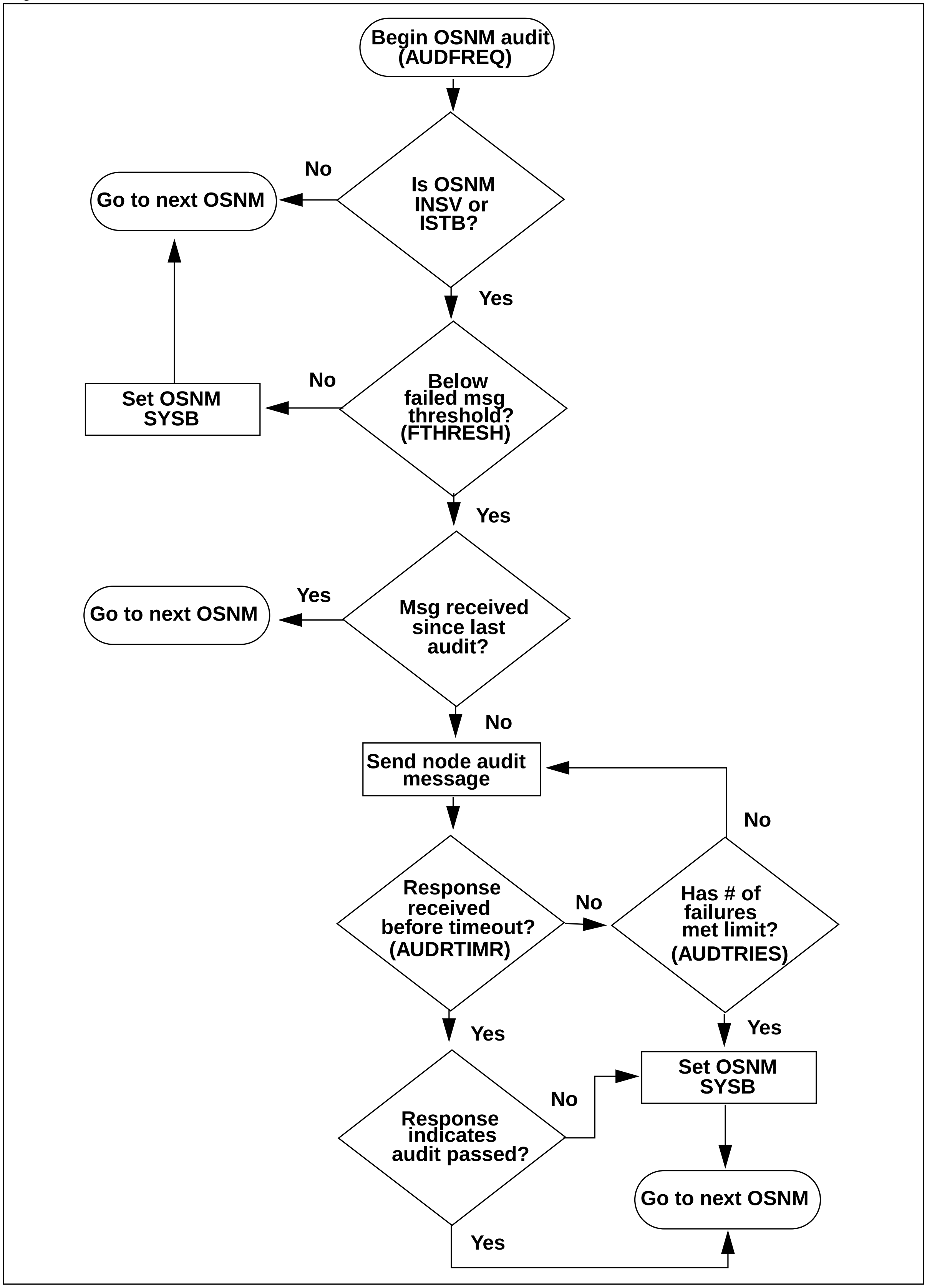
- AUDTRIES (audit tries), which specifies the number of times the switch sends an audit without receiving a response (in other words, the number of consecutive audit failures). Once this limit is met, the switch takes the node out of service.
- AUDRTIMR (audit response timer), which specifies the number of seconds the switch waits for a response to an audit from the node.
- AUDFREQ (audit frequency), which specifies the number of seconds that elapse between node audits.

The automated audit process also checks the percentage of unsuccessfully processed messages from the node against the failed message threshold (FTHRESH) datafilled in table OANODINV against the node. If the percentage exceeds the threshold, then the node is declared a babbler and the audit initiates action to make the node system busy.

OSNM node audit flow

The following flowchart illustrates the steps of an OSNM node audit flow.

Figure 179 OSNM node audit flow



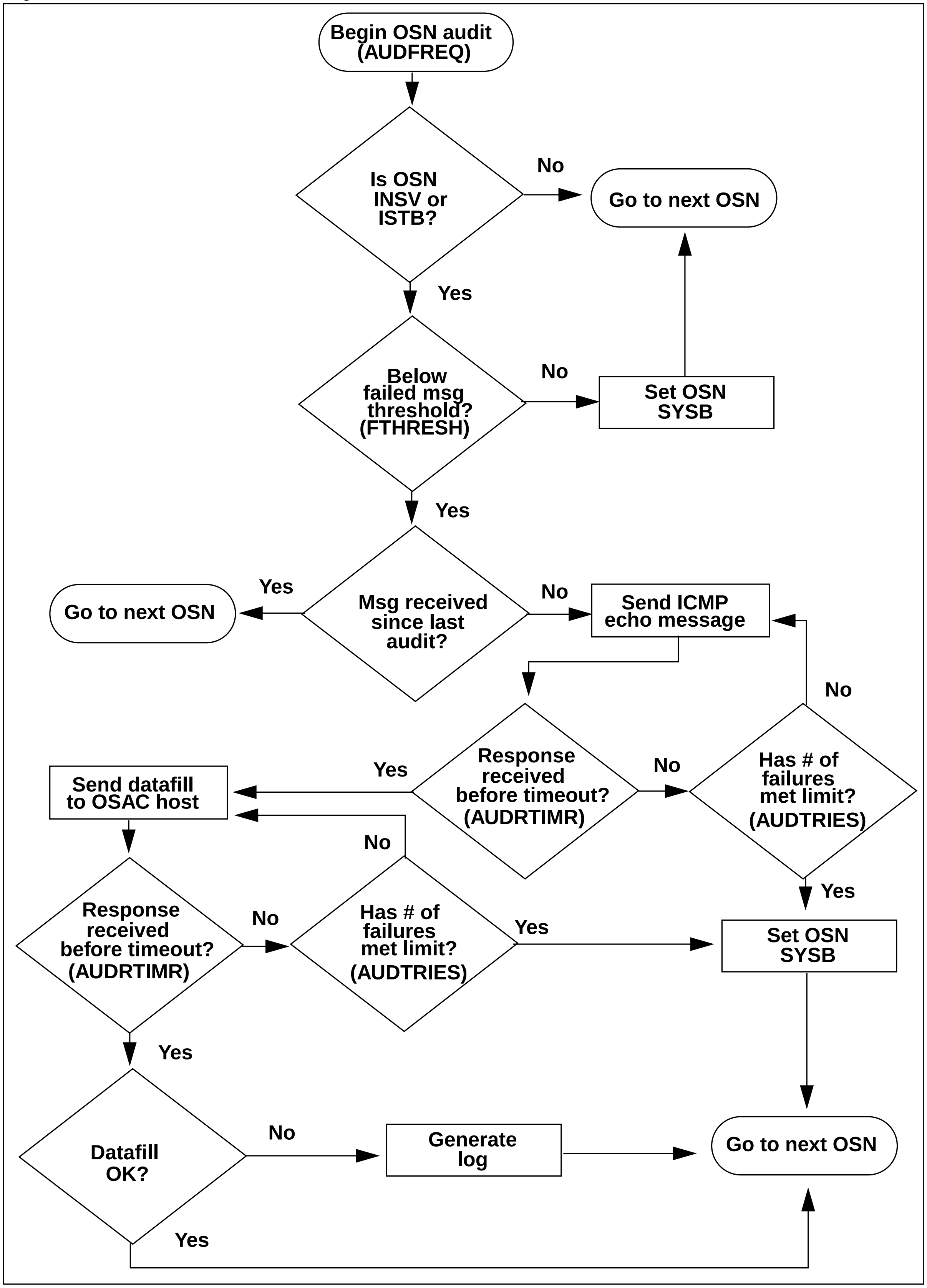
When the switch sets the state of the OSNM to system busy (SYSB), all session pools supported by the OSNM are unavailable for call processing. The switch takes down all calls with sessions to that OSNM node.

If the OSNM responds to the node audit message with an error, the switch sets the state of the OSNM to SYSB and proceeds to the next OSNM.

OSN node audit flow

The following flowchart illustrates the steps of an OSN node audit flow.

Figure 180 OSN node audit flow



If the OSAC host switch responds to the datafill check, the response message includes the following:

- whether the datafill is parallel
- whether the OSN is INSV or ISTB from the view of the OSAC host switch (to the OSAC host, the OSN is an OSNM)

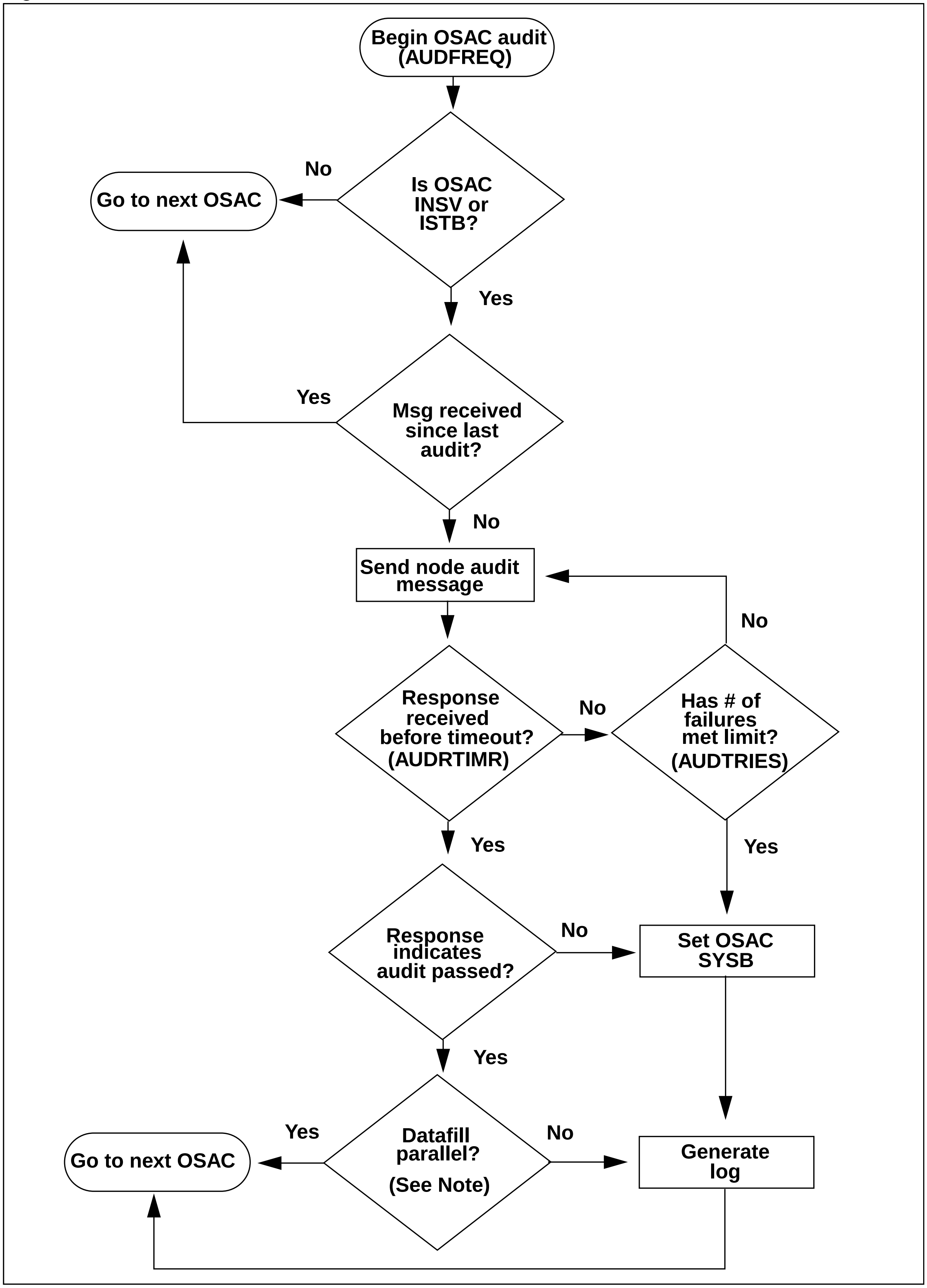
First, if the datafill is not parallel, the OSAC host switch generates a log report, but the connection state of the OSN stays in service. If the state (from the view of the host) is not INSV or ISTB, then the OSAC remote switch sets the connection state of the OSN to SYSB (even if the datafill is parallel).

When the OSAC remote sets the connection state of the OSN to SYSB, all session pools supported by the OSN are unavailable for call processing. The OSAC remote switch takes down all calls with sessions to that OSN. However, this action only affects calls at the OSAC remote switch; it does not affect calls at other OSAC switches that have sessions with the centralized SN.

OSAC node audit flow

The following flowchart illustrates the steps of an OSAC node audit flow.

Figure 181 OSAC node audit flow



Note: If the OSAC responds with an error that indicates a failed parallel datafill check, the switch generates a log report, but the connection state of the OSAC stays in service. The switch proceeds to the next OSAC node. However, if the error indicates anything other than a failed parallel datafill check, the switch sets the connection state of the OSAC node to SYSB. All INSV session pools on the OSAC node change to the CBSY state and all associated calls are taken down.

Automated RTS process

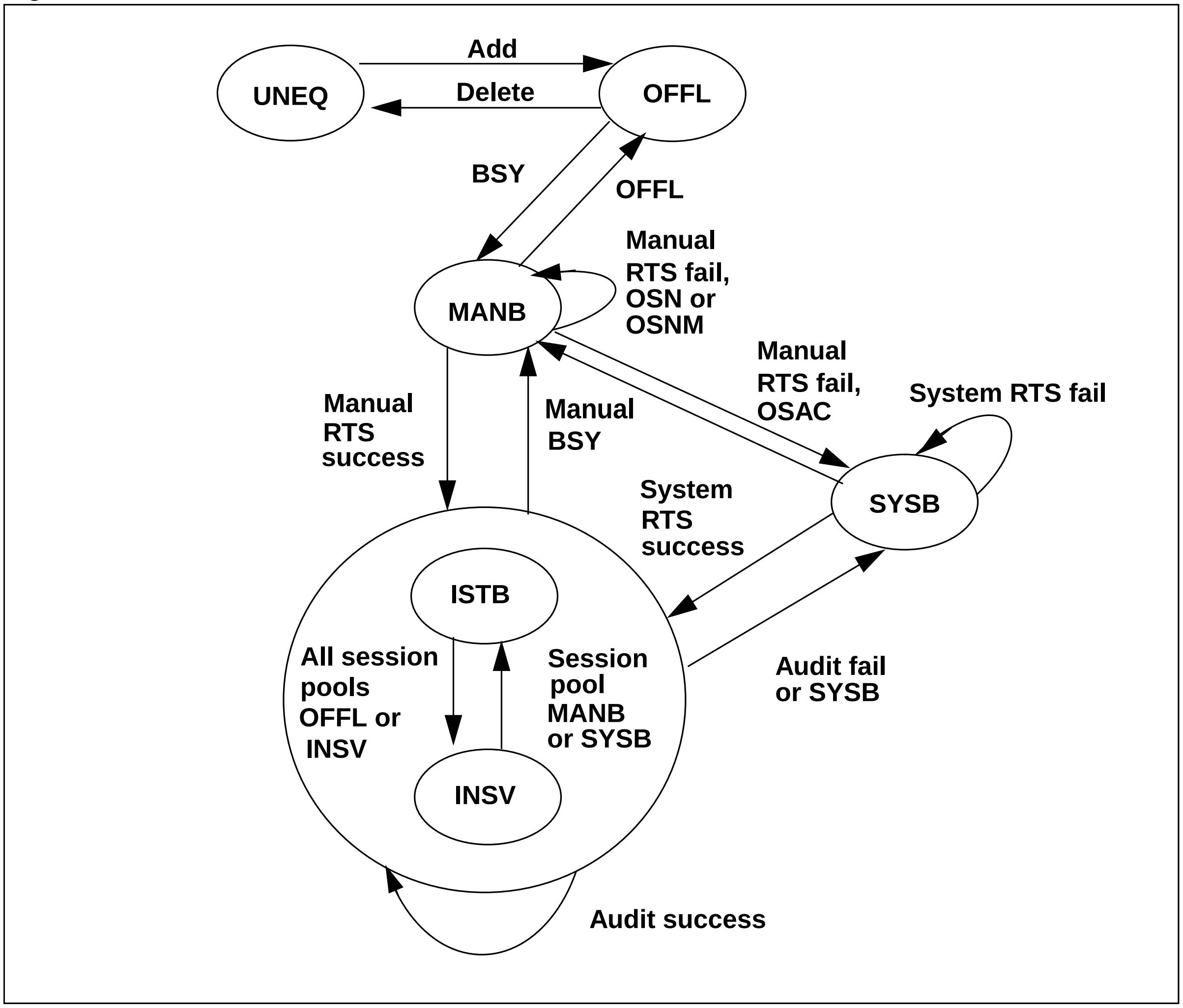
The automated RTS process tries to bring into service any system busy nodes. Approximately every 20 seconds, the automated RTS process scans the states of the nodes. If a node is in a SYSB state, and it is not marked as babbling, the switch attempts to RTS the node. If the RTS fails, the node remains SYSB.

For OSAC nodes, the automated RTS process scans the states of the connections to the OSAC nodes. If a connection is in a SYSB state, the switch attempts to RTS the OSAC. If the RTS fails, the OSAC remains SYSB.

State transitions

The following figure shows the state transitions of a node. Each transition type is described after the figure.

Figure 182 Node state transitions



UNEQ

Unequipped. The node has not been datafilled in table OANODINV.

OFFL

Offlined. The node has been datafilled, but has not been manually busied for the first time or has been offlined from the MANB state.

MANB

Manual busy. The node has been manually busied from the MAP. When a node goes MANB, all service processing provided by that node is stopped. In the CM, all in-service session pools datafilled on that node are taken out of service (CBSY). Call processing messages cannot be sent to the OSN node.

SYSB

System busy. The node has been placed in a busy state by the system. This occurs under one of the following conditions:

- when an automated audit fails
- when an OSNM node makes an unsolicited busy request
- when the OSAC host data connection is not in service

When a node goes SYSB, all service processing provided by that node is stopped. In the CM, all in-service session pools datafilled on that node are taken out of service (CBSY). Call processing messages cannot be sent to the OSN node.

INSV

In service. The node is in this state when it is first brought into service by either a manual or automated RTS and all session pools the node supports are either successfully brought into service or are OFFL. For OSAC nodes, the state of the connection is in service.

ISTB

In service trouble. The node is in this state when it is first brought into service by either a manual or automated RTS. The node remains ISTB until all session pools are brought into service. If any session pool cannot be brought into service or is MANB, the node is kept in the ISTB state. For OSAC nodes, the state of the connection is ISTB until all session pools are brought into service.

Unsolicited busy request from an OSNM node

If an OSNM node needs to take itself down for any reason, it can send an *unsolicited busy request* to the switch. When the switch receives the unsolicited request, it places the OSNM node in a SYSB state.

For an unsolicited busy request to be valid at the switch, the OSNM must be in either the INSV or ISTB state and have no outstanding BSY maintenance requests.

If the switch determines that the unsolicited busy request from the OSNM is valid, the switch performs the following tasks:

- It cancels any outstanding audit requests on the OSNM.
- It sets the maintenance state of the OSNM to SYSB.
- It sends a success message to the OSNM.

When an OSNM becomes SYSB at the switch, all calls currently being processed at the OSNM are taken down and any INSV session pools are set to CBSY.

Note: The automated RTS process tries to return to service an OSNM node that initiated an unsolicited busy request.

Centralized SNs in an OSAC network

This section discusses maintenance interactions of centralized SNs (that is, nodes that are datafilled as OSNM at the OSAC host switch and as OSN at the OSAC remote switch) and centralized session pools.

Centralized SNs

When an OSNM is brought into service at the OSAC host switch, the OSAC host informs all OSAC remotes of the state change. Each OSAC remote determines if it has the node datafilled as an OSN node. If it does and the node is SYSB, the OSAC remote initiates the OSN node RTS process for the node. Once the OSN is in service, the OSAC remote attempts to RTS the session pools supported by the OSN that are CBSY.

When an OSNM is busied at the OSAC host switch, the OSAC host informs all OSAC remotes of the state change. Each OSAC remote determines if it has the node datafilled as an OSN node. If it does and the node is ISTB or ISNV, the OSAC remote busies the node. This causes each of the in-service OSN session pools supported by the node to go CBSY. Any calls in the OSAC remote using the affected session pools are taken down.

Centralized session pools

When a centralized session pool is brought into service at the OSAC host switch, the OSAC host informs all OSAC remotes of the state change. Each OSAC remote determines if it has the session pool datafilled. If it does and the session pool is SYSB, the remote switch tries to RTS the session pool.

When a centralized session pool is busied at the OSAC host switch, the OSAC host informs all OSAC remotes of the state change. Each OSAC remote determines if it has the session pool datafilled. If it does and the session pool is INSV, the OSAC remote busies the session pool. Any calls in the OSAC remote using the centralized session pool are taken down.

Related logs

The switch generates standard PM state transition logs when a node changes maintenance states or when faults are detected (or both). The following PM logs are supported for node maintenance:

- PM100—DIAG FAIL: The node failed out-of-service diagnostics.
- PM102—SYSB: The node changed state to SYSB because of a maintenance-detected fault.
- PM103—OFFL: The node changed state to OFFL because of a manual action or because the node was added to table OANODINV.
- PM104—UNEQ: The node was deleted from table OANODINV.

- PM105—MANB: The node changed state to MANB because of a manual action.
- PM106—INSV: The node changed state to INSV from MANB because of a manual action, or from SYSB because of an automated (system) action.

OSNM-specific log reports

The following log reports are specific to OSNM nodes:

- PM128—ISTB: The OSNM has a fault condition but can still communicate with the host switch.
- OAIN700—OSNM report: The OSNM node requested a log for the reason specified in the report.
- EXT106—Minor alarm: The OSNM requested a minor alarm.
- EXT107—Major alarm: The OSNM requested a major alarm.
- EXT108—Critical alarm: The OSNM requested a critical alarm.

OSAC-specific log report

The OSAC host switch generates an OSAC601 log if the check for parallel datafill fails. Refer to Chapter 11: “OSSAIN logs,” for more information on logs.

Related OM groups

When the node is datafilled in table OANODINV, the switch uses the following OM groups to monitor maintenance events from the node:

- OAINNODE
- OANODEDC
- OAPMERRN
- OAPMTYPN
- OAPNMTC (for OSNM nodes)
- OSNNODE (for OSN nodes)
- OSACND (for OSAC nodes)

Refer to Chapter 12: “OSSAIN operational measurements,” for more information on OM groups.

Protocols

Messaging uses the following protocols in node maintenance:

- OAP is used for maintenance messages between the switch and OSNMs, and for node state inform messages between the OSAC remote switch and centralized SNs.
- User Datagram Protocol (UDP) is used to encapsulate OAP messages.

- Internet Control Message Protocol (ICMP) echo messages are used to ping nodes.
- OAP Node Connectivity Test messaging replaces ICMP echo messages in an OSAC remote in the RTS sequence for an OSN, if the OSN is at OAP 9 or higher and the OSAC host is at SN07 or higher.
- OSAC proprietary protocol is used for maintenance messages between the OSAC switches.

User interface—OAINPMDIR directory

The OAINPMDIR MAP level allows the user to post OSSAIN nodes and perform manual maintenance on the posted node or nodes.

For OSN node types, manual maintenance is on the *data connection* to the posted OSN. A maintenance action does not cause any messages to be sent to the posted OSN; rather, the action changes the connection state of the OSN (for example, from OFFL to MANB).

The OAINPMDIR MAP level is accessed from the PM level. The user interface consists of a MAP banner and MAP commands. The banner is updated when a node changes state. The following figure shows an example MAP display for an OSNM node.

Figure 183 OAINPMDIR MAP display for OSNM node

```
 CM     MS      IOD     Net     PM     CCS     Lns     Trks     Ext    APPL
  .     .       .       .       .      .       .       .        .      .


OSNM                                SysB    ManB     OffL    CBsy  ISTb  InSv
   0 Quit        PM                   0       0        0       0     0     60
   2 Post_       OSNM                 0       0        0       -     0     4
   3 ListSet
   4            OSNM  7
   5            Nodename
   6 Tst_       InSv
   7 Bsy_
   8 RTS_
   9 OffL_
  10
  11
  12 Next
  13
  14 QueryPM
  15
  16
  17 Info
  18 SesnPool

  TIME   15:31 >
```

The next subsections provide details, such as parameter definitions, error responses, and actions associated with each OAINPMDIR MAP command:

- QUIT (page 408)
- POST (page 409)
- LISTSET (page 410)
- TST (page 410)
- BSY (page 413)
- RTS (page 416)
- OFFL (page 419)
- NEXT (page 420)
- QUERYPM (page 420)
- INFO (page 422)
- SESNPOOL (page 422)
- ABTK (page 423)

QUIT

Exits user from OAINPMDIR MAP level to previous level.

Table 88 QUIT parameters

Parameter	Value	Definition
<n levels>	<number>	Quits the specified number of MAP levels.
<incrname>	PM, MTC, MAPCI	Specifies the MAP level increment that precedes the current increment in nesting.
<ALL>	ALL	Quits all MAP levels and returns user to the CI level.

The following table lists common error responses, explanations, and actions.

Table 89 QUIT responses and actions

Response	Explanation	User action
`QUIT—Unable to quit requested number of levels Last parameter evaluated was: 1`	The QUIT command failed. User entered an invalid level number or the number exceeds the number of MAP levels to quit.	Re-enter the QUIT command using the correct level number.

POST

Posts a specific node or posts the node by state. For OSN nodes, this command posts the connection of an OSN node or posts the OSN by connection state.

Table 90 POST parameters

Parameter	Value	Definition
ALL PMS	Literal “AllPMS”	Posts all PMs known to the switch.
ALL	Literal “All”	Posts all PMs of the PM type currently posted.
PM Number	0 - 255	Posts the specified PM number of the PM type currently posted.
PM Type	OSNM, OSN, OSAC	Posts the specified PM type.
ALL	Literal “All”	Posts all PMs of the specified PM type.
PMNo	{0 - 255}	Posts the specified PM number of the previously specified PM type.
State	SYSB, MANB, OFFL, ISTB, INSV	From the post set, posts PMs that are currently in the specified state.

The following table lists common error responses, explanations, and actions.

Table 91 POST responses and actions

Response	Explanation	User action
`EITHER incorrect optional parameter(s) OR too many parameters: Failed to create new Post set`	User entered incorrect parameters for the POST command.	Use HELP POST to determine the correct syntax for specifying the post set.
`<Node type> <number> is unequipped EITHER incorrect optional parameter(s) OR too many parameters: Failed to create new Post set`	The specified node and number is not a datafilled node. This applies whether specifying the PM number alone, or both the PM type and PM number.	Use HELP POST to determine the correct syntax for specifying the post set. Check table OANODINV for information on which nodes are equipped.
`No PM posted`	User posted only the PM type, without specifying a particular node.	Use the POST command with desired PM number, or with PM type and PM number, to post a set of nodes.

LISTSET

Lists the set of nodes created when posting multiple nodes.

Table 92 LISTSET parameters

Parameter	Value	Definition
No parms or ALL	No value or “ALL”	Lists the entire post set.
PM Type	OSNM, OSN, OSAC	Lists the specific node in the post set.

The following table lists common error responses, explanations, and actions.

Table 93 LISTSET responses and actions

Response	Explanation	User action
`No PM posted` `Post set is empty`	User did not post a node.	Re-enter the POST command with the correct parameter.

TST

The TST command is available with an OSNM, OSN, or OSAC node posted at the MAP. A node must be ManB to be tested.

Formerly TST sent an ICMP echo request (ping) and then a test request message, or only an ICMP echo request. Now, when TST is entered for a node, a ping is no longer sent. The commands are not changed.

For OSN, a Node Connectivity Test message is sent to the OSN node in place of the ping to verify connectivity. The purpose of this operation is to verify application-layer connectivity between the switch and a service node. This substitution occurs only if the OSAC remote is at SN09 or higher, the OSAC host is at SN07 or higher, and the OSN is at OAP 9 or higher.

For OSNM and OSAC, the original ping test was immediately followed by an OAP or OSAC message to the same node. The subsequent message also verified connectivity. The redundant ping in the test sequence for OSNM and OSAC nodes is now removed.

When TST PING is entered for a node on a platform that does not support ping from SOS, the following response message is provided: “Use TST without PING.”.

The test request message for OSNM nodes asks the service node to test itself. For OSN and OSAC nodes, the test request message is a parallel datafill check between remote and host.

Note: For OSNM nodes, ensure that the TSTTIME subfield value in table OANODINV is large enough to allow for complete node testing.

Table 94 TST parameters

Parameter	Value	Definition
No parms	No value	Tests the posted node.
<PING>	PING	Sends only an ICMP echo request to the posted node, and verifies reply.
<ALL>	ALL	Tests all nodes in the posted set.

The following table lists common error responses, explanations, and actions.

Table 95 TST responses and actions

Response	Explanation	User action
`TST Aborted`	The test was aborted by a manual ABTK command.	None
`TST Failed` `Could not send PING message`	The switch was unable to send an ICMP echo request to the posted node.	Check OAIN logs. For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface.
`TST Failed` `No reply from OSSAIN node`	For OSNM nodes, the node did not respond to the out-of-service tests or did not have enough time to complete the tests. For OSN or OSAC nodes, the node did not respond to the connectivity test or the OSAC host did not respond to the request to verify datafill.	For OSNM nodes, check that the TSTTIME in table OANODINV is large enough to allow the node to complete its diagnostics. If the timer value is large enough, troubleshoot the node to determine the reason for failure. For OSN or OSAC nodes, use the TST command with the PING parameter to verify connectivity with the OSN and with the OSAC host. If the ping test succeeds, check OAIN and OAP logs on the OSAC node.
`TST Failed` `Invalid message received, check OAP logs`	The node returned an invalid message.	Check OAP logs to determine the reason for failure. Troubleshoot the node to determine the reason for failure.

Table 95 TST responses and actions

Response	Explanation	User action
`TST Failed` `Could not send to the OSSAIN node`	The switch was unable to send a message to the node or the OSAC remote was unable to send datafill information to the OSAC host.	Check OAIN logs. For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface.
`TST Failed` `Command failed, check OAIN/OAP logs`	The node returned an error or reject message.	Troubleshoot the node to determine the reason for failure. Check OAIN and OAP logs.
`TST Failed` `Node datafill checks failed`	The OSAC host found a mismatch for OSN datafill on the OSAC remote.	Check the OSAC601 log on the OSAC host to determine which table has incorrect datafill. Enter the correct datafill.
`TST Failed` `<node failure reason text>`	The node returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`TST Failed` `Invalid PING message received`	The switch received an invalid reply in response to an ICMP message.	Troubleshoot for data network problems between the switch and the far-end node. For nodes datafilled to use the EIU, determine if the node can be pinged from the switch using the SNPINGCI tool. ***Note:*** For more information on the SNPINGCI tool, please refer to *Nonmenu Commands Reference Manual*, 297-1001-820.
`TST Failed` `Host is not in service`	The connection to the OSAC host that supports the OSN is not in service.	Bring the connection to the OSAC host into service and re-enter the TST command.
`TST Failed` `Maintenance in progress`	A maintenance task is already in progress for the node.	Re-enter the TST command.
`Request invalid: OSSAIN <node ID> IS <state>`	The node must first be in the MANB state. For OSN and OSAC nodes, the connection must first be in the MANB state.	MANB the posted node and re-enter the TST command. ***Note:*** Manually busying the node may cause call outages.

Table 95 TST responses and actions

Response	Explanation	User action
`Use TST without PING`	TST PING has been used on a switch which does not support pings from MAP.	Use command TST without the PING option. Ping from another platform.

BSY

- For OSNM nodes—Sets the state of the posted node to MANB. If the node is OFFL or SYSB, a busy message is not sent to the node. Busying an in-service OSNM node causes a node busy inform message to be sent to all OSAC remote nodes. Each OSAC remote will system busy its associated OSN node.

  Once the OSNM node completes its busy tasks, it sends a response to switch maintenance. The OSNM is responsible for taking its own session pools out of service. In-service session pools supported by this node are placed in the CBSY state.

- For OSN nodes—Sets the connection state of the posted OSN to MANB. Busying an OSN takes down calls in the OSAC remote that have sessions with the OSN. However, it does not affect the state of the centralized OSNM or the connection between other OSAC remotes and the centralized OSNM.

- For OSAC nodes—Sets the connection to the posted OSAC node to MANB at the OSAC switch where the BSY command was issued. The connection state of the OSAC switch that receives the BSY request is set to SYSB.

  In an OSAC remote switch, any calls using sessions on a busied OSAC host switch are taken down. The OSAC host switch frees those sessions for use by other OSAC nodes. In an OSAC remote switch, in-service OSN nodes that depend on the OSAC connection go SYSB.

Note: The BSY command is valid in the OFFL, INSV, ISTB, or SYSB states.

Table 96 BSY parameters

Parameter	Value	Definition
No parms	No value	Manually busies the posted node, prompting the user to confirm.
<NOWAIT>	NOWAIT	Manually busies the posted node, prompting the user to confirm. Frees up MAP for other purposes while the command is executing; no responses are seen at the MAP.
<ALL>	ALL	Manually busies all nodes in the posted set.

When the user issues the BSY command, the following warning is displayed at the MAP and a confirmation is requested.

Figure 184 BSY warning message

```
Inservice Session Pools on this node will be affected.
Please confirm (“YES,” “Y,” “NO,” or “N”):
```

If confirmation is NO, the BSY action is not performed; if confirmation is YES, the BSY action is performed and a message is displayed indicating the result.

Note: When busying a node that is OFFL or SYSB, a confirmation request is not needed because in-service calls are not affected.

The following table lists common error responses, explanations, and actions.

Table 97 BSY responses and actions

Response	Explanation	User action
`BSY Aborted`	The BSY was aborted by a manual ABTK command.	None
`BSY Failed` `No reply from OSSAIN node`	The node did not respond to the busy message.	Verify connectivity with the node.
`BSY Failed` `Could not send to the OSSAIN node`	The switch was unable to send a message to the node.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`BSY Failed` `Invalid message received, check OAP logs`	The node returned an invalid message.	Check OAP logs for error messages and re-enter the BSY command.
`BSY Failed` `Command failed, check OAIN/OAP logs`	The node returned an error or reject message.	Troubleshoot the node to determine the reason for failure. Check OAIN and OAP logs.
`BSY Failed` `Invalid node/session pool state`	The state of the OSAC connection at the target OSAC node is not INSV or ISTB.	None
`BSY Failed` `<node failure reason text>`	The node returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`BSY Failed` `Maintenance in progress`	A maintenance task is already in progress at the target OSAC node. The OSAC connection is no longer in service.	None
`Request invalid: OSSAIN <node ID> IS MANB`	The node is already in the MANB state. For OSN and OSAC nodes, the connection is already in the MANB state.	None

RTS

- For OSNM nodes—Brings the state of the posted OSNM into service. If RTS is successful, the OSNM node sends a positive response and the switch places the node in the ISTB state. Next, an RTS request message is sent to each session pool in the CBSY state. If all session pools return to service, then the OSNM goes INSV. Otherwise, the node state stays ISTB.

  When an OSNM node comes into service, a node RTS inform message is sent to all OSAC remote nodes. Each OSAC node will attempt to RTS its associated OSN node (if the OSN is SYSB).

  Note: Ensure that the RTSTIME subfield value in table OANODINV is large enough to allow the OSNM to return to service.

- For OSN nodes—Brings the connection state of the posted OSN into service. The connection state of the OSN first must be in the MANB state. When RTSing the connection state of an OSN, the remote exchanges messages with the host to verify parallel datafill and to verify that the service node is in service at the host. If these tests pass, the remote then verifies its connectivity with the OSN node using node connectivity test request if the OSAC remote is at SN09 or higher, the OSAC host is at SN07 or higher, and the OSN is at OAP 9 or higher. If a success response is received from the node, the switch continues with the RTS process. If an error response or no response is received from the node, the RTS fails.

  Note: If the OSAC remote or host is below SN07, or if the service node is below OAP release 9, then the remote uses a ping (ICMP echo) to verify connectivity with the OSN. Ping has the following limitation. Nodes send ping replies to whatever IP address the request came from. The processing does not involve OAP, and does not involve any checks by the SN of the IP address it is configured to use for the switch. Therefore, it is possible for a ping connectivity test to pass and for the OSAC remote to successfully RTS the OSN even if the SN is configured with the wrong IP address for the remote. If this happens, calls from the remote to the OSN will be unsuccessful. The problem is fixed in SN07, but the fix requires that the SN be at OAP release 9 and that the OSAC host and remote both be at SN07 (or higher).

- For OSAC nodes—Brings the connection to the posted OSAC node into service. RTS also checks for matching datafill between OSAC nodes. RTS can be performed only while the connection is in the MANB state. The RTS command must be issued from *both* the OSAC remote node and the OSAC host node before the connection between them can come into service. Also, the connection state of the OSAC node will be ISTB until all OSAC session pools return to service.

  Note: It is not unusual for an OSAC connection to initially transition to SYSB when the RTS command is entered from both switches at close to the same time. When this happens, the connection normally transitions to ISTB and then to INSV within less that a minute. There is no need to troubleshoot the SYSB reason when this happens.

After the OSAC connection is successfully brought into service, the OSAC remote switch will attempt to RTS system busy OSN nodes that depend on the OSAC connection.

Table 98 RTS parameters

Parameter	Value	Definition
No parms	No value	Returns to service the posted node.
<Nowait>	NOWAIT	Frees up MAP for other purposes while the command is executing; no responses are seen at the MAP.
<ALL>	ALL	Performs RTS operation on ALL posted nodes.

The following table lists common error responses, explanations, and actions.

Table 99 RTS responses and actions

Response	Explanation	User action
`RTS Aborted`	The RTS was aborted by a manual ABTK command.	None
`RTS Failed` `Could not send PING message`	For OSN nodes, the OSAC remote was unable to send an ICMP message to the OSN. For OSAC nodes, the OSAC node was unable to send an ICMP message to the target OSAC node.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`RTS Failed` `No reply from OSSAIN node`	For OSNM nodes, the node did not respond to the out-of-service tests or did not have enough time to complete the tests. For OSN or OSAC nodes, the node did not respond to the connectivity test or the OSAC host did not respond to the request to verify datafill.	For OSNM nodes, check that the RTSTIME in table OANODINV is large enough to allow the OSNM to complete its RTS process. If the timer value is large enough, troubleshoot the OSNM to determine the reason for failure. For OSN or OSAC nodes, use the TST command with the PING parameter to verify connectivity with the OSN and with the OSAC host. If the ping test succeeds, check OAIN and OAP logs on the OSAC node. Investigate the data network components.

Table 99 RTS responses and actions

Response	Explanation	User action
`RTS Failed` `Invalid message received, check OAP logs`	The node returned an invalid message.	Check OAP logs to determine the reason for failure. Troubleshoot the node to determine the reason for failure.
`RTS Failed` `Could not send to the OSSAIN node`	The switch was unable to send a message to the node or the OSAC remote was unable to send datafill information to the OSAC host.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`RTS Failed` `Command failed, check OAIN/ OAP logs`	The node returned an error or reject message.	Troubleshoot the node to determine the reason for failure. Check OAIN and OAP logs.
`RTS Failed` `Node datafill checks failed`	The OSAC host found a mismatch for OSN datafill on the OSAC remote.	Check the OSAC601 log on the OSAC host to determine which table has incorrect datafill. Enter the correct datafill.
`RTS Failed` `<node failure reason text>`	The node returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`RTS Failed` `Host is not in service`	The connection to the OSAC host that supports the OSN is not in service.	Bring the connection to the OSAC host into service and re-enter the RTS command.
`RTS Failed` `Maintenance in progress`	A maintenance task is already in progress for the node.	Re-enter the RTS command.
`RTS Failed` `Invalid node/session pool state`	The state of the OSAC connection at the target OSAC node must be SYSB.	RTS the connection from both OSAC nodes.
`RTS Failed` `Node has failed babbling node checks. Node must be offlined to clear fault`	The OSNM or OSN node exceeded the failed message threshold value (table OANODINV).	Verify that the babbling condition has been corrected. Offline and busy the node before re-entering the RTS command.

Table 99 RTS responses and actions

Response	Explanation	User action
`RTS Failed Incompatible protocol version`	The connection to the OSN node cannot be returned to service because the node does not communicate at a compatible level of the OAP protocol.	Upgrade the service node to a level supported by the switch.
`Request invalid: OSSAIN <node ID> IS <state>`	The node must first be in the MANB state. For OSN and OSAC nodes, the connection must first be in the MANB state.	MANB the posted node and re-enter the RTS command. ***Note:*** Manually busying the node may cause call outages.
`OSNM: SOC option OSAN0101 is not enabled.`	For OSNM nodes, the OSAN0101 SOC option is not enabled.	Enable OSAN0101 and re-enter the RTS command.
`OSAC: SOC option OSAN0101 is not enabled.`	For OSAC nodes, the OSAN0101 SOC option is not enabled.	Enable OSAN0101 and re-enter the RTS command.

OFFL

- For OSNM nodes—Sets the state of the posted OSNM node to OFFL. No OAP messages are sent to or received from the node.
- For OSN nodes—Sets the connection state of the posted OSN to OFFL.
- For OSAC nodes—Sets the connection to the posted OSAC node to OFFL. No OSAC messages are sent to or received from the target OSAC node.

Note: The OFFL command is valid only in the MANB state.

Table 100 OFFL parameters

Parameter	Value	Definition
No parms	No value	Offlines the posted node.
<ALL>	ALL	Offlines all posted nodes.

The following table lists common error responses, explanations, and actions.

Table 101 OFFL responses and actions

Response	Explanation	User action
`Request invalid: OSSAIN <node ID> IS <state>`	The node must first be in the MANB state.	MANB the posted node and re-enter the OFFL command.

NEXT

Posts the next node in the post set.

Table 102 NEXT parameters

Parameter	Value	Definition
No parms	No value	Posts the next node in the post set.
PM Type	OSNM, OSN, OSAC	Steps to the next PM type in the post set.

The following table lists common error responses, explanations, and actions.

Table 103 NEXT responses and actions

Response	Explanation	User action
`End of post set`	The post set is empty or no more nodes are in the post set.	None

QUERYPM

Displays basic information about the posted node.

Table 104 QUERYPM parameters

Parameter	Value	Definition
No parms	No value	Displays information about a posted node.
<FLT>	FLT	Displays fault information about a posted node.

The MAP displays the following response when the user enters the QUERYPM command. Actual data values depend on datafill in table OANODINV.

Note: The HOSTNAME field is displayed only for OSN nodes.

Figure 185 QUERYPM response

```
NODENAME: BRANDING_01
NODEID: 1
PROTOCOL: UDP
IPADDR: 57.192.1.98
ONODTYPE: OSN NODE
ONODLOC: 2 A 33
ONODSITE: SITEA
HOSTNAME: OSAC HOST 1
```

The following table lists common error responses, explanations, and actions.

Table 105 QUERYPM responses and actions

Response	Explanation	User action
`No PM posted`	User did not first post a node.	Re-enter the POST command with the correct parameter.

The following table lists common error responses, explanations, and actions when the <FLT> parameter is used.

Table 106 QUERYPM <FLT> responses and actions

Response	Explanation	User action
`Failed message threshold exceeded`	The switch received more than the maximum percentage of failed messages from the OSNM or OSN node. This maximum in set in table OANODINV.	Troubleshoot the node to determine the reason for failure.
`Busied at source`	For OSNM nodes, the switch received an unsolicited busy request from the node. For OSN nodes, the connection is SYSB because the associated OSNM node was busied at the OSAC host. For OSAC nodes, the connection was busied at the far end OSAC node.	Troubleshoot the node to determine the reason for failure.
`Node audit failed`	For OSNM nodes, the switch busied the node because it received no response to node audit requests. For OSN nodes, the switch busied the connection, either because it received no response to connectivity tests with the posted OSN, or because the associated OSNM node is out of service. For OSAC nodes, the switch busied the connection because it received no response to audit requests sent to the posted OSAC node.	Use the TST command to verify connectivity with the node. Check OAP, OAIN, and ITN or XAC logs at the far end OSAC node. If the far-end node is OK, investigate the data network.

Table 106 QUERYPM <FLT> responses and actions

Response	Explanation	User action
`Task aborted`	For OSN or OSAC nodes, the connection is SYSB because the RTS attempt was aborted.	BSY the connection and re-enter the RTS command.
`RTS attempt failed`	For OSN or OSAC nodes, the connection is SYSB due to failing the RTS attempt.	BSY the connection and re-enter the RTS command.
`OSAC host out of service`	For OSN or OSAC nodes, the connection is SYSB because the OSAC host that supports the OSN is out of service.	Troubleshoot the connection to the OSAC host.
`Session pool out of service`	The node is ISTB because one or more of its session pools is either MANB or SYSB.	Troubleshoot the MANB and SYSB session pools supported by the node.

INFO

Displays information about the session pools datafilled on the posted node. There are no parameters.

The MAP displays the following response when the user enters the INFO command. The information applies to all session pools datafilled on the posted node. It includes the session pool ID number, session pool name, the maximum number of sessions, and the current state of the session pool.

Figure 186 INFO response

```
                    Session Pool Info
          ID          NAME                 MAX      STATE

          0           Branding_0           100      INSV
          1           Branding_1           150      INSV
          2           YellowPg_1           50       INSV
```

The following table lists common error responses, explanations, and actions.

Table 107 INFO responses and actions

Response	Explanation	User action
`No PM posted`	User did not first post a node.	Re-enter the POST command with the correct parameter.

SESNPOOL

Displays the SESNPOOLDIR MAP sublevel. There are no parameters or associated error responses.

ABTK

Stops the current maintenance action on a posted node. ABTK is a non-menu command. There are no parameters or associated error responses.

Session pool maintenance

This section describes the maintenance of session pools. Information is provided for both SN session pools and host-remote session pools.

Defining session pools in table OASESNPL

All session pools must be datafilled in table OASESNPL before any maintenance actions can be performed. For details on all the fields and values, refer to Chapter 7: “OSSAIN data schema.”

Automated audit process

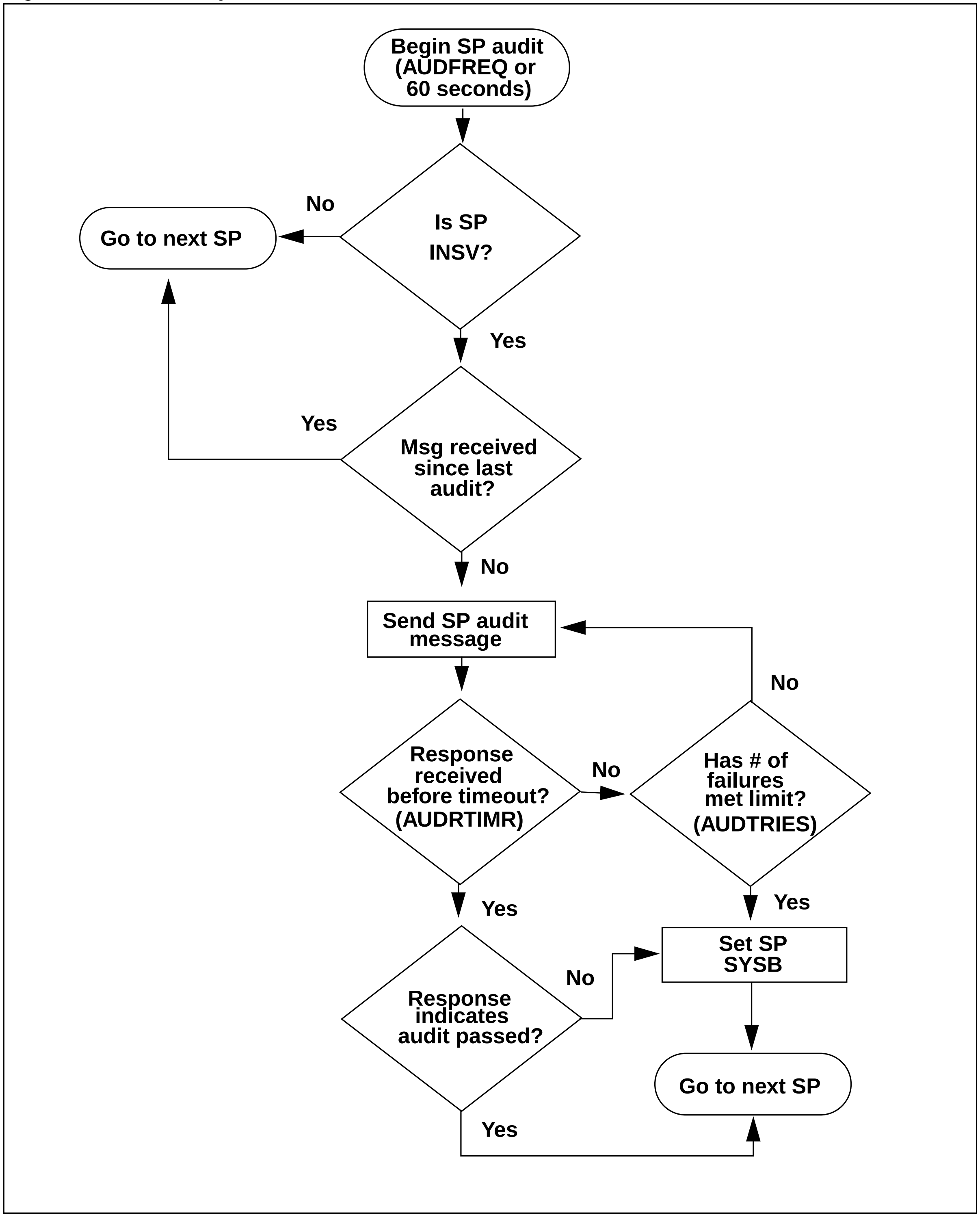
The switch performs an automated audit process, which verifies that the session pool is in service (INSV). In the OSAC environment, the audit also verifies parallel datafill for the session pool.

Datafill in three fields in table OANODINV controls the automated audit process, as follows:

- AUDTRIES (audit tries), which specifies the number of times the switch sends an audit without receiving a response (in other words, the number of consecutive audit failures). Once this limit is met, the switch takes the session pool out of service.
- AUDRTIMR (audit response timer), which specifies the number of seconds the switch waits for a response to an audit from a session pool.
- AUDFREQ (audit frequency), which specifies the minimum number of seconds that elapse between session pool audits. As of SN07, the session pool audit executes at most once a minute regardless of the datafill, so the datafilled AUDFREQ affects the frequency of session pool audits only if the datafilled frequency is greater than 60.

The following flowchart illustrates the steps of the session pool (SP) audit flow.

Figure 187 Session pool audit flow



The session pool audit varies slightly for each type of node, as follows:

- OSNM session pools

  The switch sends a session pool audit message to the OSNM session pool. If the OSNM session pool does not respond to the message, the switch sets the state of the OSNM session pool to SYSB and proceeds to the next OSNM session pool.

- OSN session pools

  The OSAC remote switch sends a request message to the OSAC host to perform a parallel datafill check on the OSN session pool. If the OSAC host does not respond to the datafill check, the OSAC remote sets the state of the OSN session pool to SYSB.

  If the OSAC host switch responds to the datafill check, the response message includes the following:

  — whether the OSN session pool is INSV from the view of the OSAC host switch (to the OSAC host, the OSN is an OSNM)

  — whether the datafill is parallel

  First, if the OSN session pool is not INSV, the switch sets the state of the OSN session pool to SYSB and proceeds to the next OSN session pool. If the OSN session pool is INSV, but the datafill is not parallel, the switch generates a log report, but the state of the OSN session pool stays in service.

- OSAC (host-remote) session pools

  The switch sends a session pool audit message to the target host-remote session pool. If the host-remote session pool does not respond to the message, the switch sets the state of the host-remote session pool to SYSB and proceeds to the next host-remote session pool.

  If the host-remote session pool responds with an error that indicates a failed parallel datafill check, the switch generates a log report, but the connection state of the host-remote session pool stays in service. The switch proceeds to the next host-remote session pool. If the error indicates anything other than a failed parallel datafill check, the switch sets the connection state of the host-remote session pool to SYSB. All INSV session pools on the OSAC node change to the CBSY state and all associated calls are taken down.

  Note: Both the OSAC host switch and the OSAC remote switch can generate session pool audits.

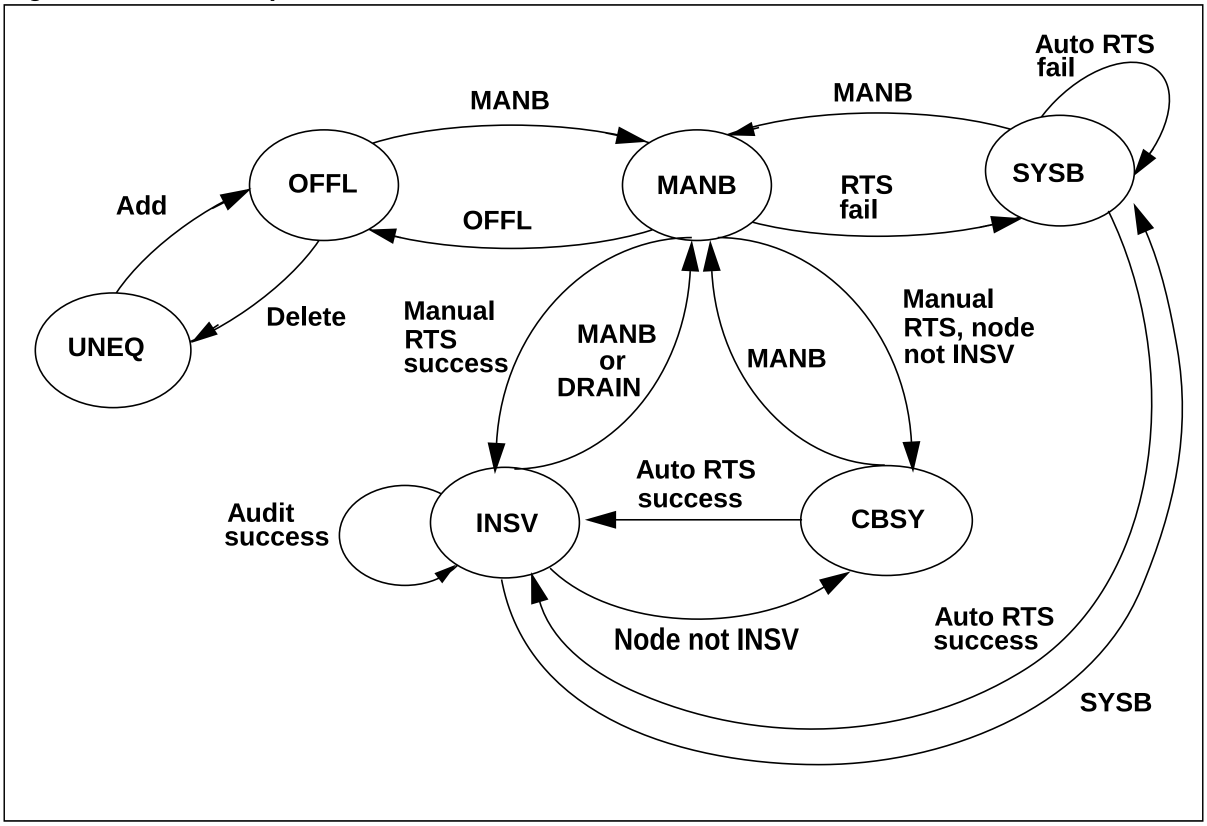
Automated RTS process

The automated RTS process tries to bring into service any system busy session pools. Approximately every 20 seconds, the automated RTS process scans the states of the session pools. If a session pool is in a SYSB state, the switch attempts to RTS the session pool. If the RTS fails, the session pool remains SYSB.

State transitions

The following figure shows the state transitions for session pools. Each transition type is described after the figure.

Figure 188 Session pool state transitions



UNEQ

Unequipped. The session pool has not been datafilled in table OASESNPL.

OFFL

Offlined. The session pool enters this state by one of two events:

- The session pool is created through datafill.
- The session pool is set to OFFL by manual action.

MANB

Manual busy. The session pool has been busied by manual action.

SYSB

System busy. The session pool has been placed in the busy state by the system.

For OSNM session pools, this occurs under one of the following conditions:

- when the session pool fails to recover from the CBSY state when the node is returned to service
- when a session pool audit fails
- when the session pool is busied by the OSNM node

For OSN session pools, this occurs under one of the following conditions:

- when the session pool fails to recover from the CBSY state when the node is returned to service
- when a session pool audit fails
- when the OSAC host informs the OSAC remote that the associated OSNM session pool is out of service

For OSAC (host-remote) session pools, this occurs under one of the following conditions:

- when the session pool fails to recover from the CBSY state when the node is returned to service
- when a session pool audit fails
- when the far end OSAC switch busies the session pool

INSV

In service. The session pool is in the INSV state when it has successfully passed an RTS request. The node that supports the session pool must be INSV or ISTB. For OSN session pools, the connection to the OSAC host that supports the centralized SN must be INSV or ISTB and the associated OSNM session pool must be INSV.

CBSY

C-side busy. The session pool enters this state when the node that supports it goes out of service. When the node goes back into service, either by manual or automatic action, all session pools in the CBSY state are also returned to service by an automated RTS. If a session pool fails to return to service, it is placed in the SYSB state.

Unsolicited busy request from a session pool

If a session pool on an OSNM node needs to take itself down for any reason, it can send an *unsolicited busy request* to the switch. When the switch receives the request, it sets the session pool state to SYSB.

For an unsolicited busy request to be valid at the switch, the session pool must be in the INSV state and have no outstanding BSY maintenance requests.

If the switch determines that the request is valid, the switch performs the following tasks:

- It cancels any outstanding audit requests on the session pool.
- It sets the maintenance state of the requesting session pool to SYSB.
- It returns a success message to the requesting session pool.

When a session pool becomes SYSB at the switch, all calls currently being processed at the session pool are taken down.

Note: The automated RTS process will try to return to service a session pool that initiated an unsolicited busy request.

Throttle request from a session pool

If a session pool on an OSNM node needs to change (increase or decrease) the number of sessions that are available, the session pool can send a *throttle request* to the switch.

For a change request to be valid at the switch, the session pool must be in the INSV state and have no outstanding maintenance requests. Also, the number of available sessions requested by the session pool must be *less than or equal to* the maximum number of sessions in table OASESNPL (MAXSESN field).

If the switch determines that the request is valid, the switch performs the following tasks:

- It immediately changes the number of available sessions in the session pool.
- It sends a success message to the requesting session pool.

Note 1: For a throttle request to be valid, the session pool must be an OSNM session pool and the TRIGEVNT field must be set to N in table OASESNPL.

Note 2: Throttle does not override any maintenance tasks.

Note 3: Users can view the number of available sessions and maximum sessions at the MAP. Refer to “User interface—SESNPOOLDIR directory” on page 432.

Increasing the number of available sessions

As soon as the switch determines that the request is valid, it changes the number of available sessions in the session pool. Because the change is immediate, it is possible that the switch could send a session begin message to a new session pool *before* it sends the change request success message. So once the session pool requests an increase, it must be able to immediately accept the increase in call traffic.

Decreasing the number of available sessions

Decreasing the number of available sessions does not take down existing calls—even if the new number of available sessions is less than the number of existing calls in the session pool. The decrease only changes the number of new calls that can use the session pool.

Interactions with RTS

When the switch returns a session pool to service, it sets the number of available sessions to the maximum value in table OASESNPL. However, the session pool can request a different value when it replies to the RTS request. The new value cannot be *greater* than the maximum value in table OASESNPL.

If the switch determines that the request is valid, it changes the number of available sessions to the requested number. If the request is not valid, the switch uses the value in table OASESNPL.

Related logs

Following is a list of OAIN logs for session pool state changes or failures:

- OAIN500—FAIL: The session pool test was performed (using the TST command) and failed.
- OAIN502—SYSB: The session pool changed state to SYSB because of a maintenance detected fault.
- OAIN503—OFFL: The session pool changed state to OFFL because of a manual request or being added to table OASESNPL.
- OAIN504—UNEQ: The session pool changed state to unequipped because of being deleted from table OASESNPL.
- OAIN505—MANB: The session pool changed state to MANB because of a manual action.
- OAIN506—INSV: The session pool changed state to INSV from MANB because of a manual request, or from SYSB because of a system request.
- OAIN507—CBSY: The node that the session pool is dependent on went out of service.

OSNM-specific log reports

The following log reports are specific to OSNM session pools:

- PM128—ISTB: The OSNM has a fault condition but can still communicate with the host switch.
- OAIN701—Session pool report: The session pool requested a log report.
- EXT106—Minor alarm: The session pool requested a minor alarm.
- EXT107—Major alarm: The session pool requested a major alarm.
- EXT108—Critical alarm: The session pool requested a critical alarm.

OSAC-specific log report

The OSAC host switch generates an OSAC601 log if the check for parallel datafill fails. Refer to Chapter 11: “OSSAIN logs,” for more information on logs.

Related OM groups

When the session pool is datafilled in table OASESNPL, the switch uses the following OM groups to monitor maintenance events from the session pool:

- OAPSPMTC
- OASESNPL
- OASNPLDC
- OSNSP (SN session pools)
- OSACSP (host-remote session pools)

Refer to Chapter 12: “OSSAIN operational measurements,” for more information on these OM groups.

Protocol

Messaging uses the following protocols in session pool maintenance:

- OAP is used for maintenance messages between the switch and SN session pools, and for session pool state inform messages between the OSAC remote switch and centralized SNs.
- OSAC proprietary protocol is used for maintenance messages between the OSAC switches.
- UDP is used to encapsulate OAP and OSAC messages.

User interface—SESNPOOLDIR directory

The SESNPOOLDIR MAP level allows the user to post session pools and perform manual maintenance on the posted session pools. Session pools can be posted by node, by state, or individually.

SESNPOOLDIR is accessed from the OAINPMDIR level. For a posted session pool that is currently in service, this level also displays another field. This field tracks the number of sessions that are available to receive new calls as well as the maximum number of sessions datafilled for the session pool in table OASESNPL.

Note: Maintenance actions performed on OSN session pools do not directly affect centralized session pools. These actions only affect the view of the session pool from the OSAC remote switch.

The user interface consists of a MAP banner and MAP commands. The banner is updated when a session pool changes state. The following figure shows an example MAP display for an OSNM session pool.

Figure 189 SESNPOOLDIR MAP display example for OSNM session pool

```
 CM     MS      IOD     Net     PM     CCS     Lns     Trks      Ext    APPL
   .    .       .       .     13OSNM  .       .       .         .      .


SesnPool                        SysB    ManB     OffL    CBsy  ISTb  InSv

   0 Quit        PM              12       2       21      0    17     42
   2 Post_       OSNM            0        0        13      -     0      2
   3 ListSet
   4           OSNM   7
   5           Nodename
   6 Tst_      InSv
   7 Bsy_
   8 RTS_        Status         SysB    ManB     OffL    CBsy         InSv
   9 OffL_       SP              0       0      75      0              2
  10
  11            OSNM  7  SP  42   INSV             39 of 50
  12 Next       Size of Post set:  1
  13
  15
  16
  17 Drain
  18 Info

  TIME   15:31 >
```

The next subsections provide details, such as parameter definitions, error responses, and actions associated with each SESNPOOLDIR MAP command:

- QUIT (page 433)
- POST (page 434)
- LISTSET (page 434)
- TST (page 435)
- BSY (page 437)
- RTS (page 439)
- OFFL (page 442)
- NEXT (page 442)
- DRAIN (page 442)
- INFO (page 444)

QUIT

Exits user from SESNPOOLDIR MAP level to previous level.

Table 108 QUIT parameters

Parameter	Value	Definition
<n levels>	<number>	Quits the specified number of MAP levels.
<incrname>	PM, MTC, MAPCI	Specifies the MAP level increment that precedes the current increment in nesting.
<ALL>	ALL	Quits all MAP levels and returns user to the CI level.

The following table lists common error responses, explanations, and actions.

Table 109 QUIT responses and actions

Response	Explanation	User action
`QUIT—Unable to quit requested number of levels Last parameter evaluated was: 1`	The QUIT command failed. User entered an invalid level number or the number exceeds the number of MAP levels to quit.	Re-enter the QUIT command using the correct level number.

POST

Posts a specific session pool by session pool ID number, all session pools by state, or all session pools assigned to a node.

Table 110 POST parameters

Parameter	Value	Definition
<SP ID number>	0 to 4094	Posts the specified session pool.
ALL	Literal “All”	Posts all session pools.
<State>	INSV, SYSB, CBSY, MANB, OFFL	Posts the session pools in a specific state.
<node ID>	0 - 255	Posts the session pools assigned to a specific node.

The following table lists common error responses, explanations, and actions.

Table 111 POST responses and actions

Response	Explanation	User action

`EITHER incorrect optional parameter(s) OR too many parameters: Failed to create new Post set`
`Invalid session pool number`
`Could not create post set`
	The specified session pool is not datafilled in table OASESNPL.	Re-enter the POST command using the proper session pool ID number.
		Check table OASESNPL to verify the session pool ID number.

LISTSET

Lists the set of session pools created when posting multiple session pools.

Table 112 LISTSET parameters

Parameter	Value	Definition
No parms or ALL	No value or “ALL”	Lists the entire post set.

The following table lists common error responses, explanations, and actions.

Table 113 LISTSET responses and actions

Response	Explanation	User action
`No session pools posted`	User did not post a session pool.	Re-enter the POST command with the correct parameter.

TST

- For OSNM session pools and OSAC host-remote session pools—Sends a session pool test message to the posted session pool. The TST command can only be used when the session pool is MANB. For OSN session pools, the TST command sends a parallel datafill check request to the OSAC host node.
- For OSNM session pools—Table OANODINV, subfield TSTTIME, defines the amount of time that the session pool is allowed to complete the TST sequence. If the session pool fails to respond during the TSTTIME interval, the session pool fails the test sequence. Ensure that the TSTTIME period is longer than the longest test period required for a session pool.

Table 114 TST parameters

Parameter	Value	Definition
No parms	No value	Tests the posted session pool.
<ALL>	ALL	Tests all session pools in the posted set.

The following table lists common error responses, explanations, and actions.

Table 115 TST responses and actions

Response	Explanation	User action
`TST Aborted`	The test was aborted by a manual ABTK command.	None
`TST Failed` `Could not send to Session Pool`	The switch was unable to send a message to the session pool.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`TST Failed` `Invalid message received, check OAP logs`	The session pool returned an invalid message.	Check OAP logs for error messages and re-enter the TST command.
`TST Failed` `Command failed, check OAIN/OAP`	The session pool returned an error or reject message.	Troubleshoot the node to determine the reason for failure. Check OAIN and OAP logs.
`TST Failed` `Sesn pool datafill checks failed`	The OSAC host found a mismatch for OSN datafill on the OSAC remote.	Check the OSAC601 log on the OSAC host to determine which table has incorrect datafill. Enter the correct datafill.

Table 115 TST responses and actions

Response	Explanation	User action
`TST Failed` `<session pool failure reason text>`	The session pool returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`TST Failed` `Host is not in service`	The connection to the OSAC host that supports the OSN session pool is not in service.	Bring the connection to the OSAC host into service and re-enter the TST command.
`TST Failed` `Maintenance in progress`	A maintenance task is already in progress for the session pool.	Re-enter the TST command.
`TST Failed` `No reply from Session Pool`	For OSNM session pools and OSAC host-remote session pools, the session pool did not respond to the TST request. For OSN session pools, the OSAC host did not respond to the parallel datafill check request.	Investigate the state of the node. If it is out of service, then a TST on its session pools will fail. Attempt to return the node to service. Use the node TST to verify connectivity with the node. For OSNM session pools, check the TSTTIME value in table OANODINV. If TSTTIME is large enough, troubleshoot the session pool to determine the reason for failure. For OSN session pools, verify that the connection to the OSAC host node is in service. Check OAIN and OAP logs at both OSAC nodes. For OSAC host-remote session pools, verify that the OSAC connection is in service. Check OAIN and OAP logs at both OSAC nodes. The problem could be in the data network.

BSY

- For OSNM session pools—Sets the session pool to the MANB state and sends a busy message to the posted session pool. BSY also sends a session pool busy inform message to all OSAC remote nodes. OSAC remote nodes then BSY their associated OSN session pool.
- For OSN session pools—Sets the session pool to MANB, and sends an OAP session pool busy inform message to the SN that supports the session pool *for SN originated session pools only.* The session pool busy inform is not sent for other OSN session pools such as subscriber originated.
- For OSAC host-remote session pools—Sets the session pool to MANB and sends an OSAC busy message to the host-remote session pool.

Table 116 BSY parameters

Parameter	Value	Definition
No parms	No value	Manually busies the posted session pool after counting and displaying the active calls; and if active calls are present, prompts the user to confirm.
<FORCE>	FORCE	Manually busies the posted session pool, overriding any checks for active calls.
<NOWAIT>	NOWAIT	Manually busies the posted session pool after counting and displaying the active calls; and if active calls are present, prompts the user to confirm. Frees up the MAP for other purposes while the command is executing; no responses are seen at the MAP.
<ALL>	ALL	Manually busies all session pools, overriding any checks for active calls.

If the user issues the BSY command and the posted session pool has calls active, the following warning is displayed at the MAP and a confirmation is requested:

Figure 190 BSY warning message

```
Session Pool <SPno.> has <no.> calls active.
Please confirm (“YES,” “Y,” “NO,” or “N”):
```

If confirmation is NO, the BSY action is not performed; if confirmation is YES, the BSY action is performed.

Note: When busying a session pool that is OFFL or SYSB, a confirmation request is not needed since in-service calls will not be affected.

Note: It is recommended that a session pool be drained if it is necessary to remove it from service. Refer to page 442 for information about the DRAIN command.

The following table lists common error responses, explanations, and actions.

Table 117 BSY responses and actions

Response	Explanation	User action
`BSY Aborted`	The BSY was aborted by a manual ABTK command.	None
`BSY Failed` `Could not send to Session Pool`	The switch was unable to send a message to the session pool.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`BSY Failed` `Invalid message received, check OAP logs`	The session pool returned an invalid response.	Check OAP logs for error messages and re-enter the BSY command.
`BSY Failed` `Command failed, check OAIN/OAP logs`	The session pool returned an error or reject message.	Troubleshoot the node to determine the reason for failure. Check OAIN and OAP logs.
`BSY Failed` `Invalid node/session pool state`	For OSAC host-remote session pools, the state of the host-remote session pool is not in service at the target OSAC node.	None
`BSY Failed` <session pool failure reason text>	The session pool returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`BSY Failed` `Maintenance in progress`	A maintenance task is already in progress for the host-remote session pool.	None
`BSY Failed` `No reply from Session Pool`	The session pool did not respond to the busy message.	For OSNM session pools, check that the TSTTIME in table OANODINV is large enough to allow the node to complete its BSY process. If the timer value is large enough, troubleshoot the node to determine the reason for failure.

RTS

- For OSNM session pools—Sends a return-to-service message to the posted session pool. Upon a successful response from the session pool, the session pool is marked INSV.

  When an OSNM session pool comes into service, a session pool RTS inform message is sent to all OSAC remote nodes. Each OSAC node will attempt to RTS its associated OSN session pool (if the OSN is SYSB). If the session pool fails to return a success response in the allotted period of time (from table OANODINV, field RTSTIME) then the session pool is marked SYSB. Ensure that the RTSTIME period in table OANODINV is longer than the longest RTS period required for a session pool.

- For OSN session pools—Sends a parallel datafill check request to the OSAC host node. Upon a successful response from the OSAC host, the session pool is marked INSV, and an OAP session pool RTS inform message is sent to the SN that supports the session pool *for SN originated session pools only*. The session pool RTS inform is not sent for other OSN session pools such as subscriber originated.
- For OSAC host-remote session pools—Sends a return-to-service message to the posted session pool. Upon a successful response from the session pool, the session pool is marked INSV. If the session pool fails to return a success response, then the session pool is marked SYSB.

For all session pools, if the node is not in service, then in-service session pools are marked CBSY.

Table 118 RTS parameters

Parameter	Value	Definition
No parms	No value	Returns to service the posted session pool.
<Nowait>	NOWAIT	Frees up MAP for other purposes while the command is executing; no responses are seen at the MAP.
<ALL>	ALL	Performs RTS operation on ALL posted session pools.

The following table lists common error responses, explanations, and actions.

Table 119 RTS responses and actions

Response	Explanation	User action
`RTS Aborted`	The RTS was aborted by a manual ABTK command.	None

Table 119 RTS responses and actions

Response	Explanation	User action
`RTS Failed` `Could not send to Session Pool`	The switch was unable to send a message to the session pool.	For nodes datafilled to use the EIU, verify that EIUs are in service and check ITN logs. For nodes datafilled to use XA-Core Ethernet interface, check that interface. Also check OAIN logs.
`RTS Failed` `Invalid message received, check OAP logs`	The session pool returned an invalid message.	Check OAP logs for error messages and re-enter the RTS command.
`RTS Failed` `Command failed, check OAIN/ OAP logs`	The session pool returned an error or reject message.	For OSNM session pools and OSAC host-remote session pools, troubleshoot the node to determine the reason for failure. For OSN session pools, check OAIN and OAP logs at the OSAC host node.
`RTS Failed` `Sesn pool datafill checks failed`	The OSAC host found a mismatch for OSN datafill on the OSAC remote.	Check the OSAC601 log on the OSAC host to determine which table has incorrect datafill. Enter the correct datafill.
`RTS Failed` `<session pool failure reason text>`	The session pool returned an error message with reason text.	Troubleshoot the node to determine the reason for failure.
`RTS Failed` `Host is not in service`	The connection to the OSAC host that supports the OSN session pool is not in service.	Bring the connection to the OSAC host into service and re-enter the RTS command.
`RTS Failed` `Maintenance in progress`	A maintenance task is already in progress for the session pool.	Re-enter the RTS command.
`RTS Failed` `Invalid node/session pool state`	The state of the host-remote session pool at the target OSAC node must be SYSB.	RTS the OSAC host-remote session pool at the OSAC host node. Then RTS the session pool at the OSAC remote node.

Table 119 RTS responses and actions

Response	Explanation	User action
`RTS Failed` `No reply from Session Pool`	For OSNM session pools and OSAC host-remote session pools, the session pool did not respond to the out-of-service request. For OSN session pools, the OSAC host did not respond to the verify datafill check.	For OSNM session pools, check that the RTSTIME in table OANODINV is large enough to allow the node to complete its RTS process. If the timer value is large enough, troubleshoot the node to determine the reason for failure. For OSN session pools and OSAC host-remote session pools, check OAIN and OAP logs at both OSAC nodes.

OFFL

Sets the state of the posted session pool to OFFL. No messages are sent to the session pool.

Table 120 OFFL parameters

Parameter	Value	Definition
No parms	No value	Offlines the posted session pool.
<ALL>	ALL	Offlines all posted session pools.

There are no associated error responses.

NEXT

Posts the next session pool in the post set. There are no parameters.

The following table lists common error responses, explanations, and actions.

Table 121 NEXT responses and actions

Response	Explanation	User action
`End of post set`	The post set is empty or no more session pools are in the post set.	None

DRAIN

Makes all idle sessions in a session pool unavailable for call processing and does not present new calls to the session pool. Active calls are allowed to progress normally. There are no parameters.

After an active call is completed on a session pool that has DRAIN in effect, the session that was used for that call is not made available for another call. Disposition routing (table OAFNDISP) is used for any new calls destined for the session pool. Gradually, all sessions in the session pool become unavailable and the session pool automatically transitions to the MANB state.

Note 1: For OSNM session pools, the DRAIN command overrides the throttle request by a session pool.

Note 2: For OSN session pools, the DRAIN command is not supported.

DRAIN is no longer in effect for a session pool under the following conditions:

- The session pool goes SYSB, CBSY, or MANB.
- The session pool is busied and returned to service.
- A restart is performed.

For OSAC session pools, DRAIN works differently if the OSAC host or remote is at SN06 or lower than if the host and/or remote is at SN07 or higher. Prior to SN07:

- If the DRAIN command was issued in an OSAC host for an OSAC session pool, the host simply informed the OSAC remote and reported to the MAP user that DRAIN succeeded. The MAP display in the host did not display the drain flag, and the session pool did not transition automatically to MANB in the host even after the last call finished using the pool.
- If the remote received a message from the host indicating that a MAP user in the host had issued the DRAIN command for an OSAC session pool, the remote transitioned its end of the session pool to MANB if no calls were currently using sessions in the pool. If calls were using sessions in the pool, the remote displayed the drain flag and stopped selecting sessions in the pool for new calls, but it did not automatically change the pool's maintenance state from INSV even after the last call stopped using the pool.
- If the DRAIN command was issued in an OSAC remote for an OSAC session pool, the remote's functionality was as described in the previous bullet.

An OSAC host at SN07 or higher will still inform the remote when the DRAIN command is issued for an OSAC session pool, but it will also display the drain flag and will automatically transition the pool to MANB when the last call stops using the pool.

An OSAC remote at SN07 or higher will stop selecting the pool for new calls if it is notified that the DRAIN command was issued at the host, but it will not display the drain flag, nor will it automatically transition its end of the pool to MANB even if no calls are using the pool.

An OSAC remote at SN07 or higher will automatically transition the pool to MANB once the last call stops using the pool, if the DRAIN command is issued in the remote.

A success response to the DRAIN command does not indicate that the pool is draining. It does not indicate that the pool is drained.

The following table lists common error responses, explanations, and actions.

Table 122 DRAIN responses and actions

Response	Explanation	User action
`Request invalid: SP <SPno.> IS <state>`	The session pool is not in an INSV state.	Repeat the DRAIN command when the session pool is INSV.

INFO

Displays information about the posted session pool. There are no parameters.

The MAP displays the following response when the user enters the INFO command. The information applies to all session pools datafilled on the posted node. It includes the session pool ID number, session pool name, the maximum number of sessions, and the current state of the session pool.

Figure 191 INFO response

```
                     Session Pool Info
          ID         NAME                  MAX      STATE
          0          HOST_2_REM1           100      INSV
          1          HOST_2_REM2           150      INSV
          2          HOST_2_REM3           50       INSV
```

The following table lists common error responses, explanations, and actions.

Table 123 INFO responses and actions

Response	Explanation	User action
`No session pools posted`	User did not post a session pool.	Re-enter the POST command with the correct parameter.

Additional utilities for OSSAIN

In addition to the node and session pool directories, the following three directories provide non-menu commands:

- OSSAINCI
- QCALL
- QVIEW

Users access the directory by entering the directory name.

OSSAINCI directory

The following table lists an explanation of commands supported from the OSSAINCI directory.

Table 124 OSSAINCI commands

Command	Explanation
LEAVE	Exits user from OSSAINCI increment.
SELECTSP	Adds the specified session pool to the list of objects in the data communications trace list.
RELEASESP	Removes the specified session pool from the list of objects in the data communications trace list.

Table 124 OSSAINCI commands

Command	Explanation
SELECTNODE	Adds the specified node and session pools on that node to the list of objects in the data communications trace list.
RELEASENODE	Removes the specified node and session pools on that node from the list of objects in the data communications trace list.
RELEASEALL	Removes all objects from the data communications trace list.
TRACEON	Starts data communications tracing. All messages processed for selected nodes and session pool are displayed as OAIN logs.
TRACEOFF	Stops data communications tracing.
CLEAR	Removed in TOPS14.
DISPLAY	Removed in TOPS14.
SESSIONTOVID	Displays the VID (virtual terminal identifier) corresponding to the specified session pool and session.
HELP	Displays the list of valid commands available under the OSSAIN CI increment.

The following paragraphs provide details, such as the command syntax, responses, and actions associated with each OSSAINCI command.

LEAVE

The following table lists responses and actions.

Table 125 LEAVE responses and actions

Response	System action	User action

`>OSSAINCI`
`OSSAIN:`
`>LEAVE`
`CI:`
`>`
	The system exits the OSSAIN increment and returns the user to the CI prompt.	None

SELECTSP

The following table lists responses and actions.

Table 126 SELECTSP responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >SELECTSP 0 SESSION POOL ADDED TO TRACE LIST	The system adds the specified session pool to the trace list.	Activate tracing using the TRACEON command.

RELEASESP

The following table lists responses and actions.

Table 127 RELEASESP responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >RELEASESP 0 SESSION POOL REMOVED FROM TRACE LIST	The system removes the specified session pool from the trace list.	None

SELECTNODE

The following table lists responses and actions.

Table 128 SELECTNODE responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >SELECTNODE 0 NODE ADDED TO TRACE LIST	The system adds the specified node and its session pools to the trace list. No action is taken if the node or session pools are already in the trace list.	Activate tracing using the TRACEON command.

RELEASENODE

The following table lists responses and actions.

Table 129 RELEASENODE responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >RELEASENODE 0 NODE REMOVED FROM TRACE LIST	The system removes the specified node and its session pools from the trace list. No action is taken if the node is not in the trace list.	None

RELEASEALL

The following table lists responses and actions.

Table 130 RELEASEALL responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >RELEASEALL ALL ITEMS HAVE BEEN REMOVED FROM THE TRACE LIST	The system removes all nodes and session pools from the trace list. No action is taken if no objects are in the trace list.	None

TRACEON

The following table lists responses and actions.

Table 131 TRACEON responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >TRACEON OSSAIN DATA COMMUNICATIONS TRACE ENABLED	The system starts data communications tracing. When a message is processed by data communications, it is checked to see if its header information includes a selected node or session pool. If it does, the message is displayed through an OAIN log report.	Open OAIN logs to view traced messages.

TRACEOFF

The following table lists responses and actions.

Table 132 TRACEOFF responses and actions

Response	System action	User action
>OSSAINCI OSSAIN: >TRACEOFF OSSAIN DATA COMMUNICATIONS TRACE DISABLED	The system stops dumping of traced messages as OAIN logs.	None

DISPLAY

This command was removed as part of the TOPS14 changes. Since SOC has its own usage registers for counting messages, separate counters for each function are no longer required. The removal of this command also eliminated a source of blocking in the OSSAIN software.

CLEAR

This command was removed as part of the TOPS14 change to remove the Display command.

SESSIONTOVID

The following table lists responses and actions.

Table 133 SESSIONTOVID responses and actions

Response	System action	User action

`>OSSAINCI`
`OSSAIN:`
`>SESSIONTOVID 0 0`
`NN = 009A  TN = 0001`
	None	None

HELP

The following table lists responses and actions.

Table 134 HELP responses and actions

<table>
<tr><th>Response</th><th>System action</th><th>User action</th></tr>
<tr><td><pre>
>OSSAINCI

OSSAIN:

>HELP

            OSSAIN Command Increment

This command increment provides access to the following commands.

OSSAIN Message Trace:

 SELECTSP - Add a session pool to the trace list.
 RELEASESP - Remove a session pool from the trace list.
 SELECTNODE - Add a node and all session pools on the node to the trace
list.
 RELEASENODE - Remove a node an all session pools on the node form the
trace list.
 RELEASEALL - Remove all session pools and nodes from trace list.
 TRACEON - Turn OSSAIN message tracing on.
 TRACEOFF - Turn OSSAIN message tracing off.

Miscellaneous:

  SESSIONTOVID - Convert a session to is corresponding VID.

</pre></td><td>None</td><td>None</td></tr>
</table>

QCALL directory

The QCALL directory details the refinement and call assignment information of one particular call having a unique set of characteristics. The following table lists a brief explanation of the command supported from the QCALL directory.

Table 135 QCALL commands

Command	Explanation
START	Displays the final CT4Q assignment information based on the call criteria. For CT4Qs assigned to OSSAIN, the OSSAIN control list is displayed. For CT4Qs assigned to an operator, the call queue and QMS service are displayed.

The following example shows a sample QCALL report.

Table 136 QCALL report example

Response

```
STARTing simulated call queue assignment processing

  Initial CT4Q = UNSPEC
   CT4Q after CT4QCLAS refinement = UNSPEC
   CT4Q after CT4QORIG refinement = UNSPEC

  The final CT4Q is: DEBIT
  The OA Control List is: DEBIT_LIST
```

QVIEW directory

The QVIEW directory details the refinement and call assignment information of a whole set of calls with all of their possible characteristics. The following table lists a brief explanation of the command supported from the QVIEW directory.

Table 137 QCALL commands

Command	Explanation
START	Displays the final CT4Q assignment information based on the call criteria. For CT4Qs assigned to OSSAIN, the OSSAIN control list is displayed. For CT4Qs assigned to an operator, the CallQ and QMS service are displayed.

The following example shows a sample QVIEW report. In the example, the first part shows assignment info for a CT4Q routing to an OSSAIN control list called DEBIT_CTL. The second part shows assignment info for a CT4Q assigned a call queue and QMS service.

Table 138 QVIEW report example

Response

```
                ** QVIEW REPORT on ACTIVE table for PREOPR Ordering

                CO: OA                    -->>> CT4Q: 0_DEBIT

OLDCT4Q            TABLE             CRITERION          NEWCT4Q         ASSIGNMENT INFO

0_DEBIT            ORIG               T1_ORIG           DEBIT_0            DEBIT_CTL
0_DEBIT            ORIG               T1_ORIG           DEBIT_1            DEBIT_CTL
0_DEBIT            ORIG               T1_ORIG           DEBIT_2            DEBIT_CTL

0_DEBIT            OAINCTLA                                                DEBIT_CTL


                CO: OA                  -->>> CT4Q: INWARD_121

OLDCT4Q            TABLE             CRITERION          NEWCT4Q         ASSIGNMENT INFO

INWARD_121         ORIG               T1_ORIG           MP1_INW       CQ53      T_TA
INWARD_121         ORIG               T2_ORIG           MP1_INW       CQ63      T_TA
INWARD_121         ORIG               T3_ORIG           MP1_INW       CQ83      T_TA

INWARD_121         TQMSFCQA                                            CQ8      T_TA

--------------------------------------------------------
- SUMMARY REPORT on ACTIVE table for PREOPR Ordering
--------------------------------------------------------

            CT4Q TABLE                         NUMBER OF REFINEMENTS
             ----------                         -----------------------
             CT4QORIG                                     3

CT4Qs NOT assigned a call queue or control list:

               CT4Q                                      TABLE
              ------                                    ------

CALLQs NOT assigned a QMS service:
```

OSSAIN alarms

Two sets of alarms are described in this section, as follows:

- alarms requested by OSNM nodes or OSNM session pools
- OSSAIN node state transition alarms

OSNM node and session pool alarms

An OSNM node and its session pools can request that the switch generate an alarm and a corresponding EXT log report. The following table lists each type of alarm and EXT log.

Note 1: The alarm does not use a timer; it must be turned off manually by the user or by a request from the OSNM or session pool.

Note 2: Refer to Chapter 11: “OSSAIN logs,” for more information on EXT log reports.

Table 139 OSNM alarms

Alarm name	Corresponding log report
SN_CRITICIAL	EXT108
SP_CRITICAL	EXT108
SN_MAJOR	EXT107
SP_MAJOR	EXT107
SN_MINOR	EXT106
SP_MINOR	EXT106

OSSAIN node state transition alarms

The following table shows the relationship of the state of the OSSAIN node to the alarm generated at the switch.

Table 140 OSSAIN alarms

OSSAIN node state	Alarm generated
SYSB (system busy)	Major
MANB (manual busy)	Minor
ISTB (in service trouble)	Minor
UNEQ (unequipped)	No alarm
OFFL (offline)	No alarm
INSV (in service)	No alarm

As soon as the node causing the alarm is in the MANB state, the alarm is reduced in severity to a minor alarm. If the node is returned to service, the alarm is cleared.

Chapter 11: OSSAIN logs

This chapter provides information on logs for OSSAIN. Each log is given a brief description, an example log report, an action, and a list of any associated OM registers.

Logs associated with OSSAIN are of the following types:

- OAIN (beginning page 453)
- OAP (beginning page 490)
- OSAC (beginning page 496)
- EXT (beginning page 501)
- AUD (beginning page 503)
- PM (beginning page 505)
- DAS (beginning page 508)
- TCCI (beginning page 510)
- QMIS (beginning page 512)
- TOPS (beginning page 513)

Note: For complete information on all log reports for the DMS switch, refer to *Log Report Reference Manual*.

OAIN logs

OAIN logs are related to OSSAIN call processing or maintenance. The following table lists each OAIN log in order and the page in this chapter where its description begins.

Table 141 Location of OAIN log descriptions

OAIN log	Page number
OAIN200	page 455
OAIN201	page 456
OAIN203	page 456
OAIN205	page 456

Table 141 Location of OAIN log descriptions

OAIN log	Page number
OAIN206	page 457
OAIN207	page 457
OAIN208	page 458
OAIN209	page 458
OAIN210	page 458
OAIN300	page 459
OAIN301	page 459
OAIN302	page 460
OAIN303	page 461
OAIN304	page 461
OAIN305	page 462
OAIN500	page 463
OAIN502	page 464
OAIN503	page 464
OAIN504	page 465
OAIN505	page 465
OAIN506	page 465
OAIN507	page 466
OAIN600	page 466
OAIN601	page 467
OAIN602	page 467
OAIN603	page 468
OAIN604	page 468
OAIN605	page 469
OAIN606	page 469
OAIN607	page 473
OAIN608	page 475
OAIN609	page 476
OAIN610	page 477
OAIN611	page 477

Table 141 Location of OAIN log descriptions

OAIN log	Page number
OAIN612	page 478
OAIN613	page 479
OAIN614	page 479
OAIN615	page 480
OAIN616	page 480
OAIN617	page 481
OAIN618	page 482
OAIN619	page 483
OAIN620	page 484
OAIN621	page 485
OAIN622	page 486
OAIN623	page 487
OAIN624	page 488
OAIN625	page 488
OAIN700	page 489
OAIN701	page 489

OAIN200

This log is generated when an attempt to find the disposition for an OSSAIN call fails because no entry for the current function name is present in table OAFNDISP.

Figure 192 Example log report for OAIN200

```
OAIN200 JAN24 07:46:17 8701 INFO BAD OAFNDISP DATAFILL
DATAFILL TABLE OAFNDISP WITH BRANDING
```

Action

Datafill table OAFNDISP with the function name indicated in the log.

OM register

None

OAIN201

This log is generated at the OSAC host switch (or standalone switch) when the SN logical voice channel is not datafilled in table OAVLMAP. The failure is reported to the SN, at which time the node can select another logical voice channel for the voice connection.

Figure 193 Example log report for OAIN201

```
OAIN201 JAN24 07:46:17 8701 INFO BAD OAVLMAP DATAFILL
DATAFILL TABLE OAVLMAP WITH NODE_1 10
```

Action

Datafill table OAVLMAP with the node name and logical channel number indicated in the log.

OM register

None

OAIN203

This log is generated when an attempt to transfer to a control list for an OSSAIN call fails because no entry for the requested control list is present in table OACTLDEF.

Figure 194 Example log report for OAIN203

```
OAIN203 JAN24 07:46:17 8701 INFO Bad OACTLDEF Datafill
DATAFILL TABLE OACTLDEF WITH INDEX 0
```

Action

Datafill table OACTLDEF with the control list name indicated in the log.

OM register

This log is associated with OM group OAPCALP3, register XFRCTRE.

OAIN205

This log is generated when an attempt to select a CT4Q assigned to OSSAIN processing fails because the control list is not assigned in table OAINCTLA.

Figure 195 Example log report for OAIN205

```
OAIN205 FEB28 07:46:17 8701 INFO NO OSSAIN CONTROL LIST
CT4Q: NEW_SERVICE1
```

Action

Determine whether the CT4Q should be datafilled as an OSSAIN CT4Q in table CT4QNAMS. If so, datafill the corresponding control list name (from table OACTLDEF) in table OAINCTLA.

OM register

None

OAIN206

This log is generated when a preopr, post-auto, recall, or asst refinement of a TOPS CT4Q results in an OSSAIN CT4Q.

Figure 196 Example log report for OAIN206

```
OAIN206 JAN24 07:46:17 8701 INFO Invalid CT4Q Refinement
OLD_CT4Q: 0_MINUS_TOPS      NEW_CT4Q: 0_MINUS_OSSAIN
REFINEMENT: POST-AUTO
```

Action

Check the refinement tables to find the invalid refinement.

OM register

None

OAIN207

This log is generated when an attempt is made to select a CT4Q assigned to receive OSSAIN preprocessing that has no corresponding datafill in table OAINPRE.

Figure 197 Example log report for OAIN207

```
OAIN207 JUN27 10:41:21 5678 INFO NO OAINPRE FUNCTION
CT4Q: NEW_SERVICE1
```

Action

Determine whether the CT4Q should be datafilled to receive OSSAIN preprocessing in table CT4QNAMS. If so, datafill table OAINPRE with an appropriate function name from table OAFUNDEF.

OM register

None

OAIN208

This log is generated when the DMS switch cannot make the voice link connection because datafill is missing in table SNVLGRP.

Figure 198 Example log report for OAIN208

```
OAIN208 JUN27 10:41:21 5678 INFO BAD SNVLGRP DATAFILL
MISSING VOICE LINK INFORMATION FOR
SN: SN_01
FUNCTION: CALLING_CARD
```

Action

Datafill table SNVLGRP with voice link information for the SN and function pair indicated in the log report.

OM register

None

OAIN209

This log is generated for the following conditions:

- when the switch cannot translate a DN obtained from table OAFNDISP used in disposition routing
- when the DN length is greater than 10 digits for North American translations
- when translations routes the call to treatment

Figure 199 Example log report for OAIN209

```
OAIN209 MAY24 07:46:17 8701 DEFAULT DN TRANSLATIONS FAILURE
CHECK TRANSLATIONS FOR FOLLOWING DN DATAFILLED IN OAFNDISP
FN: CLG_CARD
DN: 201-220-1234
```

Action

Verify translations for the specified DN. If the DN is incorrect, change the DN in table OAFNDISP against the specified function.

OM register

None

OAIN210

This log is generated when non-Nortel DMS/DAS protocol is used to connect to a passive operator.

Figure 200 Example log report for OAIN210

```
OAIN210 JAN24 07:46:17 8701 INFO DAS PROTOCOL NOT SUPPORTED
DAS Protocol: IBM
```

Action

Ensure that Standard Nortel-DMS/DAS protocol is used when connecting to a passive operator.

OM register

None

OAIN300

This log is generated when a message arrives that is not expected. This does not include invalid messages from an SN. (Invalid messages from an SN are handled by the OAP protocol.)

Figure 201 Example log report for OAIN300

```
OAIN300 JAN24 07:46:17 8701 TBL OSSAIN UNEXPECTED MSG
CALLID: 0302 0011
SN:   NAV_LISTING_SVCS     SNID: OSN 6
SNVL: CKT OSSAINVL 22
FN:   YELLOW_PAGES       SESSPL: 4 LISTING_SERVICES
POS:
CLG:  CKT  BELLIC 34        CLD: CKT ITOG 11
FROM: CKT  BELLIC 34
TEXT: TERMINAL FORCE RELEASED
MSG:  0033B27BB00BEA18B00BEA300005FD0004150021B00BEA48000BCE14
```

Action

Collect logs and check the circuit specified in the FROM field in the log.

OM register

None

OAIN301

This log is generated at the standalone switch if there is a resource problem. This would include the following scenarios:

- conference port unavailable
- OSSAIN recording units unavailable
- OSAC recording units unavailable (generated at the OSAC host switch)
- service change over an SN transition failed

Figure 202 Example log report for OAIN301

```
OAIN301 FEB28 07:46:17 8701 TBL OSSAIN RESOURCE PROBLEM
CALLID: 0302 0011
TROUBLE: SERVICE CHANGE FAILED
```

Action

The action depends on the trouble indicated as follows:

- conference port unavailable

  Check the state of the conference three port and conference six port circuits. Also, check to see if more conference circuits are required to support office traffic. For more information on hardware resources, refer to Chapter 6: “OSSAIN engineering.”

- OSSAIN recording unit unavailable

  Check the value of office parameter OSSAIN_NUM_RU in table OFCENG. It may need to be increased. For more information, refer to Chapter 6: “OSSAIN engineering.”

- OSAC recording unit available

  Check the value of office parameter OSAC_NUM_RU in table OFCENG. It may need to be increased. For more information, refer to Chapter 6: “OSSAIN engineering.”

- service change over a SN transition failed

  Errors can include called party attached and call orig of country direct. For the called party attached error, the SN needs to release the called party and then request service change via OAP. For the call orig error, no action is required or possible.

OM register

None

OAIN302

This log is generated when OSSAIN call processing attempts to use an OSSAIN service that does not have its SOC (software optionality control) option setting in the ON state.

Figure 203 Example log report for OAIN302

```
OAIN302 FEB28 07:46:17 8701 INFO OSSAIN SOC NOT ON
CALLID: 0302 0111
CT4Q: NEW_SERVICE1
SOCID OSAN0101
```

Action

Verify that the SOC option indicated in the log has the ON state assigned. For more information on OSSAIN SOC codes, refer to Chapter 8: “OSSAIN software optionality control.” Also check datafill in the QMS refinement tables and table CT4QNAMS to determine whether calls should receive OSSAIN service.

OM register

None

OAIN303

This log is generated when OSSAIN trigger processing attempts to route a call to a TOPS automated system (such as ACTS or MCCS) or to a treatment. OSSAIN trigger events can route calls only to a TOPS operator or OSSAIN SN. OAIN303 also is generated if the call attempts to trigger to an operator while the call is in an OSSAIN conference.

Figure 204 Example log report for OAIN303

```
OAIN303 FEB28 07:46:17 8701 INFO TRIGGER ROUTE ERROR
CALLID: 0302 0011
PROFIDX: 21
TRIGEVNT: PTYD
ACTION: ACTS
TEXT: TRIGGER TO TOPS AUTO ERROR
```

Action

Check the OSSAIN datafill for the trigger profile index (PROFIDX field) and trigger event (TRIGEVNT field) indicated in the log. Ensure that this trigger event routes only to a control list or function that maps to a TOPS operator or an OSSAIN SN.

OM register

None

OAIN304

This log is generated when OSSAIN trigger processing attempts to inform the SN of a trigger event but fails to obtain a session to the SN.

Figure 205 Example log report for OAIN304

```
OAIN304 FEB28 07:46:17 8701 INFO INVALID REQUEST SESSION
CALLID: 0302 0011
REASON: call_deflected
TEXT: Invalid call queue definition for Trigger Event Inform
```

Action

Check the OSSAIN function datafill (table OAFUNDEF) and call queue datafill (table QMSCQDEF) to ensure that the call queue datafilled against the function does not support queuing.

OM register

None

OAIN305

This log is generated when a CT4Q that is eligible for OSSAIN preprocessing routes to a TOPS automated system or operator instead of to an OSSAIN SN.

Figure 206 Example log report for OAIN305

```
OAIN305  JUN24 11:47:38 2133 INFO OSSAIN ROUTE ERROR
    CALLID: 0302 0011
    CT4Q: 0_PLUS_TOPS
    FUNCTION: AABS_VSN
    TEXT:  PREPROCESSING TO TOPSAUTO ERROR
```

Action

Ensure that the CT4Q using OSSAIN preprocessing routes only to a function provided by an SN. Check datafill in the following tables for the CT4Q and the function specified in the log:

- CT4QNAMS
- OAINPRE
- OAFUNDEF

OM register

None

OAIN306

This log is generated to indicate a call routed to treatment because too many DA recalls have already been performed and the call was attempting to go to yet another DARECALL function.

Figure 207 Example log report for OAIN306

```
OAIN306 FEB28 07:46:17 8701 INFO TRMT: MAX DA RECALLS REACHED
CALLID: 0302 0011
FUNCTION: DA_RECALL
CALLING: CKT T907TI00 1
```

Action

Check table VROPT maximum_da_recalls to ensure count is as desired.
Note: This will include TOPS operator DA recalls also.

Check table OAFUNDEF to ensure the function datafill DARECALL is set as desired.

Change transfer or trigger information to route to different function.

Maintain a count in context block to route appropriately.

OM register

None

OAIN500

This log is generated when a session pool test command fails.

Figure 208 Example log report for OAIN500

```
OAIN500 JUN27 10:41:21 5678 DIAG FAIL OSNM 0
SP Name: Branding_1
SP ID: 0
Reason Text: No reply from session pool
```

Action

None

OM register

This log is associated with OM group OASNPOOL, register TSTFAIL.

OAIN502

This log is generated when a session pool goes system busy.

Figure 209 Example log report for OAIN502

```
OAIN502 JUN27 10:41:21 5678 SYSB OSNM 0
SYSB from INSV
SP Name: Branding_1
SP ID: 0
Reason Text: RTS failed
```

Action

Consult the SN session pool application information provided by the SN vendor.

OM register

This log is associated with the following registers in OM group OASNPOOL:

- SPSYB
- AUDFAIL
- RTSFAIL

OAIN503

This log is generated when a session pool is placed in the offline state.

Figure 210 Example log report for OAIN503

```
OAIN503 JUN27 10:41:21 5678 OFFL OSNM 0
OFFL from MANB
SP Name: Branding_1
SP ID: 0
Reason Text: Manual action
```

Action

None

OM register

None

OAIN504

This log is generated when a session pool is deleted from table OASESNPL.

Figure 211 Example log report for OAIN504

```
OAIN504 JUN27 10:41:21 5678 UNEQ OSNM 0
UNEQ from OFFL
SP Name: Branding_1
SP ID: 0
Reason Text:
```

Action

None

OM register

None

OAIN505

This log is generated when a session pool is placed in the MANB state.

Figure 212 Example log report for OAIN505

```
OAIN505 JUN27 10:41:21 5678 MANB OSNM 0
MANB from INSV
SP Name: Branding_1
SP ID: 0
Reason Text: Manual action
```

Action

None

OM register

This log is associated with OM group OASNPOOL, register SPMANB.

OAIN506

This log is generated when a session pool goes in service because of a manual action.

Figure 213 Example log report for OAIN506

```
OAIN506 JUN27 10:41:21 5678 RTS OSNM 0
INSV from MANB
SP Name: Branding_1
SP ID: 0
Reason Text: Manual action
```

Action

None

OM register

This log is associated with OM group OASNPOOL, register SPINSV.

OAIN507

This log is generated when an in-service session pool goes C-side busy because the SN on which it is dependent goes out of service.

Figure 214 Example log report for OAIN507

```
OAIN507 JUN27 10:41:21 5678 CBSY OSNM 0
CBSY from INSV
SP Name: Branding_1
SP ID: 0
Reason Text: System action
```

Action

Determine the reason for the SN going out of service. Once it goes in service, the session pool, if left in the CBSY state, automatically returns to service.

OM register

This log is associated with OM group OASNPOOL, register SPCBSY.

OAIN600

This log is generated when the on-hook sanity timer expires. When the subscriber on-hook timer expires, the other subscribers’ on-hook timers and hook status information are evaluated to determine if the call sanity timer should be initiated. The on-hook timer itself does not take the call down; however, if the call sanity timer expires, the call is taken down.

Figure 215 Example log report for OAIN600

```
OAIN600 JAN24 07:46:17 8701 INFO ON-HOOK TIMER EXPIRED
CALLID: 0302 0011
SN:   NAV_LISTING_SVCS     SNID: OSN 6
SNVL: CKT OSSAINVL 22
FN:   YELLOW_PAGES       SESSPL: 4 LISTING_SERVICES
PTY:  CALLING                    CKT BELLIC 34
```

Action

Check the on-hook sanity timer value for the session pool specified in the log. Increase the value if it is insufficient. If it is sufficient, check the service node logic associated with the function provider to ensure that it is not inadvertently holding the connection to the subscriber.

OM register

None

OAIN601

This log is generated when the call sanity timer or queued sanity timer expires. When the call sanity timer expires, the SN and its associated facilities (voice link) are disconnected and the call is taken down. When the queued sanity timer expires, the call is taken down.

Figure 216 Example log report for OAIN601

```
OAIN601 JAN24 07:46:17 8701 INFO CALL SANITY TIMER EXPIRED
CALLID: 0302 0011
SN:   NAV_LISTING_SVCS     SNID: OSN 6
SNVL: CKT OSSAINVL 22
FN:   YELLOW_PAGES       SESSPL: 4 LISTING_SERVICES
```

Action

When a session pool (SESSPL field) is specified in the log, the call sanity timer (while at the SN) has expired. Check the call sanity timer value for the session pool specified. Increase the value if it is insufficient. If it is sufficient, check the service node logic associated with the session pool.

When no session pool (SESSPL field) is specified in the log, the queued sanity timer has expired. Check the states of the session pools serving the call queue for the function indicated in the log.

Note: No values appear in the SN, SNID, SNVL, and SESSPL fields when the queued sanity timer expires.

OM register

None

OAIN602

This log is generated at the OSAC host switch (or standalone switch) when the service node voice channel to connect is in use.

Figure 217 Example log report for OAIN602

```
OAIN602 JAN24 07:46:17 8701 INFO VOICE CKT IN USE
SN:   NODE_1         SNVL: CKT OSSAINVL 24
```

Action

None

OM register

None

OAIN603

This log is generated at the OSAC host switch (or standalone switch) when the service node voice channel to connect is busy. The switch reports the error to the SN, at which time it can select a new voice channel.

Figure 218 Example log report for OAIN603

```
OAIN603 JAN24 07:46:17 8701 INFO VOICE CKT BUSY
SN:   NODE_1         SNVL: CKT OSSAINVL 24
```

Action

Post the voice circuit at the MAP and bring it into service.

OM register

None

OAIN604

This log is generated during invalid transfer attempts, which include the following examples:

- transferring a call to an operator or AABS while the call is participating in an OSSAIN conference
- transferring a country direct call to a DA operator
- transferring a call to a DA operator when the calling party has already gone onhook

Figure 219 Example log report for OAIN604

```
OAIN604 JAN24 07:46:17 8701 INFO Invalid Transfer Attempt
FN: NODE_TO_OPR
```

Action

Change the function in table OAFUNDEF to a value other than operator.

OM register

None

OAIN605

This log is generated when data communications receives a message with an OAP protocol version that is not optimal.

Figure 220 Example log report for OAIN605

```
OAIN605 JAN24 07:46:17 8701 INFO INVALID PROTOCOL VERSION
  MESSAGE PROTOCOL RELEASE:     1
  MESSAGE PROTOCOL INCREMENT:   2
  SENDERS PROTOCOL RELEASE:     1
  SENDERS PROTOCOL INCREMENT:   2
  SWITCH PROTOCOL RELEASE:      5
  SWITCH PROTOCOL INCREMENT:    0
  NETWORK ADDRESS:  47 55 1 2 : 7001
  MESSAGE:
  050000003300180002010201C02F6201282300014
  01F0F00050640000009411O1AB0F2205F63301501
  FFFFF2FDE0160BCDF230155F0F022FF00FF00FF00
```

Action

The session pool listed on the identified SN (NETWORK ADDRESS field) is running an OAP protocol version that cannot be supported by the version running on the DMS. The session pool must be disabled and upgraded to a compatible OAP protocol version.

OM register

This log is associated with the following OM groups and registers:

- OM group OADATCOM, register ORCVRTFL
- OM group OANODEDC, register ONRCRTFL
- OM group OASNPLDC, register OSRCRTFL

OAIN606

This log is generated when data communications software fails to route an incoming message to the OSSAIN software that would process it. Following are the possible reasons:

- invalid message class identifier
- invalid message length
- invalid operation offset
- invalid source node identifier
- invalid session pool identifier
- invalid session identifier
- invalid session pool state

- invalid node state
- invalid message size
- msg received at unexpected destination address
- msg received with unexpected source address
- session pool not supported by node
- unable to forward msg to destination
- destination’s queue at max depth

Figure 221 Example log report for OAIN606

```
OAIN606 JAN24 07:46:17 8701 INFO UNDELIVERABLE MESSAGE
  MESSAGE PROTOCOL RELEASE:     1
  MESSAGE PROTOCOL INCREMENT:   2
  SENDERS PROTOCOL RELEASE:     1
  SENDERS PROTOCOL INCREMENT:   2
  NODE NAME: SN_1                         NODE ID: 1
  SESSION POOL NAME: SP_15                SESSION POOL ID: 15
  SESSION ID:  0
  MESSAGE CLASS: CALLP                    MESSAGE CLASS ID: 0005
  NETWORK ADDRESS: 47 55 1 2 : 7001
  REASON: INVALID OPERATION OFFSET
  MESSAGE:
  05000000330018000201020 1C02F6201282300014
  01F0F000506400000941101AB0F2205F63301501
  FFFFF2FDE0160BCDF230155F0F022FF00FF00FF00
```

Action

The action depends on the reason (REASON field) for the undeliverable message, described in the following paragraphs.

Note: The action to be taken for OAIN606 logs may require personnel in SN software support and IP network support in order to examine SN software or messaging errors.

- Invalid message class identifier

  A message was corrupted or the sending software was using an invalid message class identifier. Message corruption is suspected if the log is generated for other nodes on the same LAN subnet or over the same WAN. If this is the only node causing this log to be generated, the software on the node should be investigated to verify that it is correctly transmitting OAP messages.

- Invalid message length

  A message was corrupted or the sending software incorrectly calculated the message length.

- Invalid operation offset

  The message was corrupted or the offset value provided by the SN was incorrectly calculated.

- Invalid source node identifier

  The message was corrupted or the SN provided an incorrect node identifier.

- Invalid network address

  The network address provided by the SN does not match the DMS switch datafill. This can occur if the message was corrupted, the SN provided incorrect network address information, or the SN network address was not properly datafilled on the DMS switch.

  Verify the DMS switch datafill and SN software to ensure that the SN's network address is correctly datafilled.

- Invalid session pool identifier

  The message was corrupted or the SN provided an unregistered session pool identifier. If message corruption is not suspected, verify that the session pool has been datafilled in the DMS switch.

- Invalid session identifier

  The message was corrupted or the session identifier provided by the node was incorrect. If message corruption is not suspected, the invalid session identifier reason may be produced by the SN specifying a session identifier that exceeds the current maximum session limit for the node. This indicates an SN software error.

- Invalid session pool state

  The session pool identified by the message is in an invalid state for sending or receiving messages. The session pool may not have successfully processed a maintenance busy message.

- Invalid node state

  The node identified by the message is in an invalid state for sending or receiving messages. The session pool may not have successfully processed a maintenance busy message.

- Invalid message size

  The message does not provide the required minimum amount of information to be processed by data communications. The message was corrupted or the SN sent a malformed message.

- Msg received at unexpected destination address

  The incoming message was received at a core IP address other than the one datafilled in the switch for the far-end node. The NETWORK ADDRESS in the log is the actual destination address at which the message was received.

The most likely explanation is that the node that sent the message is configured to have the wrong one of the core’s IP addresses. If the switch where the log was generated is datafilled to use an EIU for messaging with the far-end node, the IP address in table IPNETWRK at the switch should be configured in the far-end node as the switch address. If the switch is datafilled to use XA-Core Ethernet interface for messaging with the node, the CM IP address datafilled against the node in table OANODINV in the switch should be configured in the node as the switch IP address.

A less likely possibility is that the security of the service provider’s network may have been compromised in a way that causes messages to be mis-routed.

- Msg received with unexpected source address

  The source IP address, port, or both in the incoming message do not match switch datafill for the node whose node name and ID are shown in the log. The NETWORK ADDRESS field shows the actual source address.

  The most likely explanation is incorrect datafill for the node’s IP address at the switch at which the log is generated. Another possibility is that the security of the service provider’s network may have been compromised, and a rogue node may be sending OAP messages to the switch.

- Session pool not supported by node

  The session pool identified by the message is not datafilled for the SN specified by the message. Check the SN to verify that it is using the correct session pool identifier and node identifier.

- Unable to forward msg to destination

  An internal DMS messaging failure occurred or an unsolicited maintenance message was received from an SN or session pool. Verify that the indicated SN or session pool is properly responding to DMS originated maintenance messages. SWERRS should be checked for additional information on internal messaging failure.

- Destination’s queue at max depth

  The remote attempted to query the SN, but failed because the destination queue had reached its maximum depth and was full. The process could not queue either the log or alarm of the session pool, or the log or unsolicited message of the node. If this log persists, contact support.

OM register

This log is associated with the following registers in OM group OADATCOM:

- ORCVRTFL
- OMSGRCFL

This log is associated with the following registers in OM group OANODEDC:

- ONMSGRFL
- ONRCRTFL

This log is associated with the following registers in OM group OASNPLDC:

- OSMSGRFL
- OSRCRTFL

OAIN607

This log is generated when an error is encountered by OSSAIN data communications software while interfacing with the lower-layer data transport software. The underlying problem is unlikely to be in TOPS or OSSAIN code. It is more likely to be either a hardware problem or a software problem in the underlying software.

The reason text may be either of the following:

- XAETHR TRANSMIT FAILURE
- TLI TRANSMIT FAILURE

Figure 222 Example log report for OAIN607

```
OAIN607 JAN24 07:46:17 8701 INFO TRANSPORT INTERFACE ERROR
  NODE NAME: SN_1                         NODE ID: 1
  SESSION POOL NAME: SP_15                SESSION POOL ID: 15
  MESSAGE CLASS: CALLP                    MESSAGE CLASS ID: 0005
  NETWORK ADDRESS:  47 55 1 2 : 9000
  REASON: TLI TRANSMIT FAILURE
  MESSAGE:
  050000003300180002010201C02F6201282300014
  01F0F000506400000941101AB0F2205F63301501
  FFFFF2FDE0160BCDF230155F0F022FF00FF00FF00
```

Action

The action depends on the reason (REASON field).

A reason including the word XAETHR means the problem was encountered when OSSAIN was attempting to use the XA-Core Ethernet interface. Check XAC logs for any that might indicate the underlying cause of the OSSAIN problem. Investigate at MAP levels MAPCI;MTC;XAC;IO and MAPCI;MTC;XAC;ETHR. The peg counts in CI XAIPTOOL may be useful.

Note: TOPS does not generate an OAIN607 log if the problem is that no route exists for the source IP address. The underlying transport always raises an XAC alarm when there is no route for messages from a CM host IP address.

OSSAIN logs would only clutter up the log system and make it more difficult to troubleshoot the real problem.

A reason including TLI means the problem was encountered when OSSAIN was attempting to use an Ethernet Interface Unit (EIU). First, verify that all EIUs used for OSSAIN are in service. If so, check for any ITN (Inter Network) logs may accompany the OAIN607 log.

The following ITN logs may be useful in determining corrective actions:

- ITN 301—The ITN subsystem generates this report when an incoming IP packet cannot be delivered to its destination because the route to the destination is unknown.
- ITN 302—The ITN subsystem generates this report when an incoming IP packet cannot be delivered to its destination because the route to the destination is unavailable.
- ITN 305—The ITN subsystem generates this report when an incoming IP packet cannot be delivered to its destination due to IP screening.
- ITN 310—The ITN subsystem generates this report when the subsystem fails to transmit a message, originating on this node because of IP screening.
- ITN 312—The ITN subsystem generates this report when it fails to transmit a message originated on this node because the route to the destination is unknown.
- ITN 313—The ITN subsystem generates this report when failing to transmit a message, generated on this node, because the route to the destination is unavailable.

Note: For complete information on ITN logs, refer to *Log Report Reference Manual*.

Events that cause an OAIN607 log may also cause the lower software subsystems to generate a SWERR (software error). SWERRs should be investigated to determine the reason for the error.

OM register

This log is associated with the following OM groups and registers:

- OM group OADATCOM, register OMSGSNFL
- OM group OANODEDC, register ONMSGSFL
- OM group OASNPLDC, register OSMSGSFL

OAIN608

This log is generated when data communications message tracing has been enabled for the session pool or node specified by an incoming (IC) or outgoing (OG) call processing class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 223 Example log report for OAIN608

```
OAIN608 JAN24 07:46:17 8701 INFO OAP CALLP MSG DUMP
  OAP IC CALLP CLASS MESSAGE
  CLASSID:        5      OPTIONS:   0
  LENGTH         48      OFFSET:    26
  MSG REL:        1      MSG INC:   0
  SND REL:        1      SND INC:   0
  SRC NODEID:     0      SEQ NUM:   1
  DEST NODEID:   32
  FN ID:         10      POOLID:    0
  SESN ID:        0      CALLID:    FFFF FFFF
  MSG LEN:       27
  Invoke  Session Initiation Request
  OPERATION ID:   060F  INVOKE ID: 010D
  OP LEN:        24      ARG LEN:   12
  Network Service DB
  0F01 0200 0500
  Charge Status DB
  0A01 0200 FF00
  82A118000202020F068204C000F0102005000A010200FF00
```

Action

None; this log is for information only.

OM register

None

OAIN609

This log is generated when data communications message tracing has been enabled for the node specified by an incoming (IC) or outgoing (OG) node class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 224 Example log report for OAIN609

```
OAIN609 JAN24 07:46:17 8701 INFO OAP NODE MSG DUMP
  OAP OG NODE CLASS MESSAGE
  CLASSID:       1      OPTIONS:   0
  LENGTH        36      OFFSET:    16
  MSG REL:       1      MSG INC:   0
  SND REL:       1      SND INC:   0
  SRC NODEID:    0      SEQ NUM:   1
  DEST NODEID:  32
  MSG LEN:      21
  Invoke  Node RTS Request
  OPERATION ID:   0010  INVOKE ID: 0139
  OP LEN:       22      ARG LEN:   10
  Time DB
  0F01 0200 0500 0A01 0200
  82A118000202020F068204C000F0102005000A010200FF00
```

Action

None; this log is for information only.

OM register

None

OAIN610

This log is generated when data communications message tracing has been enabled for the session pool or node specified by an incoming (IC) or outgoing (OG) session pool class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 225 Example log report for OAIN610

```
OAIN610 JAN24 07:46:17 8701 INFO OAP SESN POOL MSG DUMP
  OAP OG SESSION POOL CLASS MESSAGE
  CLASSID:       4      OPTIONS:   0
  LENGTH        24      OFFSET:    18
  MSG REL:       1      MSG INC:   0
  SND REL:       1      SND INC:   0
  SRC NODEID:    0      SEQ NUM:   1
  DEST NODEID:  32
  POOLID:        0
  MSG LEN:      15
  Invoke  Session Initiation Request
  OPERATION ID:   060F  INVOKE ID: 010D
  OP LEN:       24      ARG LEN:   12
  Network Service DB
  0F01 0200 0500
  Charge Status DB
  0A01 0200 FF00
  82A118000202020F068204C000F0102005000A010200FF00
```

Action

None; this log is for information only.

OM register

None

OAIN611

This log is generated when data communications tracing has been enabled and a message of undetermined format is processed. The length of the message and the message data in hex format are displayed. The text area at the beginning of the log report indicates whether the message originated from the switch (OG, outgoing) or from an SN (IC, incoming).

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 226 Example log report for OAIN611

```
OAIN611 JAN24 07:46:17 8701 INFO OAP UNKNOWN MSG DUMP
  OAP IC MESSAGE
  MSG LEN:   15
  82A118000202020F068204C000F0102005000A010200FF00
```

Action

Check the log to determine whether the message originated from the DMS switch (outgoing—OG) or from an SN (incoming—IC). Troubleshoot the originating node.

OM register

None

OAIN612

A session in a session pool is represented internally to the DMS switch by both a queuing agent and a virtual terminal identifier (VID). This log is generated when an internal audit process determines that the state of the queuing agent does not match the state of the VID. For example, in an agent state mismatch, an agent (session) that is not serving a call could appear internally to be serving a call.

Note: VID also refers to the call being served by a specific session.

Figure 227 Example log report for OAIN612

```
OAIN612 JAN24 07:46:17 8701 INFO AGENT STATE MISMATCH
  STATE OF OSSAIN QUEUING AGENT DOES
  NOT MATCH ITS EXPECTED VALUE OF LINKEDTOCPTLB.
  AGENT NUMBER: 48
  SESSION POOL: 2
  SESSION ID:   5
  VID STATE:    0
```

Action

Users should contact Nortel when this log is generated, because internal switch resources are not being freed or initialized properly.

OM register

None

OAIN613

This log is generated when an internal audit process determines that the state of the VID does not match the state of the session pool. For example, in a VID state mismatch, a session in a session pool that is not in service could appear internally to be serving a call or to be available to serve a call.

This log also is generated when an available session in a session pool that is in service appears internally not to be available to serve a call.

Figure 228 Example log report for OAIN613

```
OAIN613 JAN24 07:46:17 8701 INFO VID STATE MISMATCH
  STATE OF OSSAIN VID DOES NOT
  MATCH ITS EXPECTED VALUE.
  SESSION POOL:   2
  SESSION ID:     5
  CURRENT STATE:  5
  EXPECTED STATE: 2
```

Action

None

OM register

None

OAIN614

This log is generated when the DMS switch changes the version of the OAP protocol being used for an OSNM node or session pool.

Figure 229 Example log report for OAIN614

```
OAIN614 JAN24 07:46:17 8701 INFO OAP PROTOCOL VERSION CHANGE
  NODE NAME: SN_1                     NODE ID: 1
  SESSION POOL NAME: SP_15            SESSION POOL ID: 15
  PREVIOUS PROTOCOL RELEASE:   1
  PREVIOUS PROTOCOL INCREMENT: 0
  NEW PROTOCOL RELEASE:        2
  NEW PROTOCOL INCREMENT:      0
```

Action

If this log is generated during RTS of the OSNM or session pool, user should repeat the RTS command.

OM register

This log is associated with the following OM groups and registers:

- OM group OAPMTYPS, register INREJCT
- OM group OAPMTYPN, register NIREJCTN

OAIN615

This log is no longer generated as of TOPS14. Formerly it was generated when the DMS switch denied an RLT request from an SN.

OAIN616

This log is generated when data communications message tracing has been enabled for the node specified by an incoming (IC) OAP node log inform message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 230 Example log report for OAIN616

```
OAIN616 JUN17 10:01:41 4501 INFO OAP NODE LOG MSG DUMP
OAP IC NODE LOG CLASS MESSAGE
CLASSID:          2      OPTIONS: 0
LENGTH:         114      OFFSET:  18
MSG REL:          3      MSG INC: 0
SND REL:          3      SND INC: 0
SRC NODEID:      23      SEQ NUM: 4
DEST NODEID:    768      SOL NUM: 0
MSG LEN:         60
Invoke    Node Log Inform
OPERATION ID:  010F   INVOKE ID:  0135
OP LEN:          98   ARG LEN:    86
Report Text DB
3C01 5200 16FF 6854 7369 6920 2073 6572 6F70 7472 6C20 6E69
2065 2E31 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000
82A16200020235010202OF01820456003C01520016FF6854736969202073
65726F7074726C206E6920652E3100000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000
```

Action

None; this log is for information only.

OM register

None

OAIN617

This log is generated when data communications message tracing has been enabled for the session pool specified by an incoming (IC) OAP session pool alarm inform message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 231 Example log report for OAIN617

```
OAIN617 JUN17 10:01:41 4501 INFO OAP SP ALARM MSG DUMP
OAP IC SESSION POOL ALARM CLASS MESSAG
CLASSID:          7      OPTIONS: 0
LENGTH:         122      OFFSET:  20
MSG REL:          3      MSG INC: 0
SND REL:          3      SND INC: 0
SRC NODEID:      23      SEQ NUM: 4
DEST NODEID:    768      SOL NUM: 0
POOLID:          23
MSG LEN:         64
Invoke    Session Pool Alarm Inform
OPERATION ID:  0307   INVOKE ID:  0136
OP LEN:         104   ARG LEN:    92
Alarm Information DB
3B01 0200 0100
Report Text DB
3C01 5200 16FF 6854 7369 6920 2073 6572 6F70 7472 6C20 6E69
2065 2E31 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000
82A168000202360102020703820450C003B01020001003C01520016FF6854
73696920207365726F7074726C206E6920652E310000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000
```

Action

None; this log is for information only.

OM register

None

OAIN618

This log is generated when data communications message tracing has been enabled for the session pool specified by an incoming (IC) OAP session pool log inform message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 232 Example log report for OAIN618

```
OAIN618 JUN17 10:01:41 4501 INFO OAP SP LOG MSG DUMP
OAP IC SESSION POOL LOG CLASS MESSAGE
CLASSID:          3      OPTIONS: 0
LENGTH:         116      OFFSET:  20
MSG REL:          3      MSG INC: 0
SND REL:          3      SND INC: 0
SRC NODEID:      23      SEQ NUM: 4
DEST NODEID:    768      SOL NUM: 0
POOLID:          23
MSG LEN:         61
Invoke    Session Pool Log Inform
OPERATION ID:  0306   INVOKE ID:  0134
OP LEN:          98   ARG LEN:    86
Report Text DB
3C01 5200 16FF 6854 7369 6920 2073 6572 6F70 7472 6C20 6E69
2065 2E31 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000
82A162000202340102020603820456003C01520016FF6854736969202073
65726F7074726C206E6920652E31000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000
```

Action

None; this log is for information only.

OM register

None

OAIN619

This log is generated when data communications message tracing has been enabled for the session pool or node specified by an incoming (IC) or outgoing (OG) OSAC call processing class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 233 Example log report for OAIN619

```
OAIN619 JAN24 07:46:17 8701 INFO OSAC CALLP MSG DUMP
OSAC IC CALLP CLASS MESSAGE
CLASSID:     25       OPTIONS: 0
LENGTH:      40       OFFSET:  24
MSG REL:      1       MSG INC: 0
SND REL:      1       SND INC: 0
SRC NODEID   60       SEQ NUM: 1
DEST NODEID  32       POOLID:  61
SESN ID:      5       CALLID:  0013 004C
MSG LEN:     23
Invoke    OSAC Session Begin Request
OPERATION ID:  0204   INVOKE ID:  0100
OP LEN:          18   ARG LEN:    6
OSN Function DB
01E0 0200 4B00
82A112000202000102020402820406000 1E002004B00
```

Action

None; this log is for information only.

OM register

None

OAIN620

This log is generated when data communications message tracing has been enabled for the node specified by an incoming (IC) or outgoing (OG) OSAC node class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 234 Example log report for OAIN620

```
OAIN620 JUN17 10:01:41 4501 INFO OSAC NODE MSG DUMP
OSAC IC NODE CLASS MESSAGE
CLASSID:        22      OPTIONS: 0
LENGTH:         64      OFFSET:  18
MSG REL:         1      MSG INC: 0
SND REL:         1      SND INC: 0
SRC NODEID:     60      SRC SOLNO: 112
DEST NODEID:    32      DEST SOLNO: 255
MSG LEN:        35
Invoke    OSAC Node Audit Request
OPERATION ID:  0208   INVOKE ID:  0170
OP LEN:          48   ARG LEN:    36
Service Node DB
05E0 1600 2000 0003 00FF F52F 0109 FFFF FFFF FFFF FFFF FFFF
FFFF
Switch Time DB
0CE0 0600 0561 1208 1913
82A13000020270010202080282042400005E016002000000300FFF52F0109
FFFFFFFFFFFFFFFFFFFFFFFF0CE00600056112081913
```

Action

None; this log is for information only.

OM register

None

OAIN621

This log is generated when data communications message tracing has been enabled for the node specified by an incoming (IC) or outgoing (OG) OSN node class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 235 Example log report for OAIN621

```
OAIN621 JUN17 10:01:41 4501 INFO OSN NODE MSG DUMP
OSN IC NODE CLASS MESSAGE
CLASSID:        24      OPTIONS: 0
LENGTH:         48      OFFSET:  20
MSG REL:         1      MSG INC: 0
SND REL:         1      SND INC: 0
SRC NODEID:     60      SRC SOLNO: 112
DEST NODEID:    32      DEST SOLNO: 255
MSG LEN:        27
Invoke    OSN Session Pool Datafill Check Request
OPERATION ID:  0214   INVOKE ID:  019E
OP LEN:          30   ARG LEN:    18
Session Pool DB
08E0 0E00 2E00 1800 3200 631B FF7F 0000 FF02
82A11E0002029E010202140282041200008E00E002E0018003200631BFF7F
0000FF02
```

Action

None; this log is for information only.

OM register

None

OAIN622

This log is generated when data communications message tracing has been enabled for the session pool specified by an incoming (IC) or outgoing (OG) OSAC session pool class message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 236 Example log report for OAIN622

```
OAIN622 JUN17 10:01:41 4501 INFO OSAC SESN POOL MSG DUMP
OSAC OG SESSION POOL CLASS MESSAGE
CLASSID:        23      OPTIONS: 0
LENGTH:         48      OFFSET:  20
MSG REL:         1      MSG INC: 0
SND REL:         1      SND INC: 0
SRC NODEID:     32      SRC SOLNO: 112
DEST NODEID:    60      DEST SOLNO: 255
POOLID:         61
MSG LEN:        27
Invoke    OSAC Session Pool Audit Request
OPERATION ID:  020C   INVOKE ID:  010A
OP LEN:          30   ARG LEN:    18
Session Pool DB
08E0 0E00 3D00 1800 3C00 FF7F FF7F 0400 FF01
82A11E0002020A0102020C028204120008E00E003D0018003C00FF7FFF7F
0400FF01
```

Action

None; this log is for information only.

OM register

None

OAIN623

This log is generated when data communications message tracing has been enabled for the node specified by an incoming (IC) OAP node alarm inform message.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 237 Example log report for OAIN623

```
OAIN623 JUN17 10:01:41 4501 INFO OAP NODE ALARM MSG DUMP
OAP IC NODE ALARM CLASS MESSAGE
CLASSID:         6      OPTIONS: 0
LENGTH:        120      OFFSET:  18
MSG REL:         3      MSG INC: 0
SND REL:         3      SND INC: 0
SRC NODEID:     23      SEQ NUM: 4
DEST NODEID:   768      SOL NUM: 0
MSG LEN:        63
Invoke    Node Alarm Inform
OPERATION ID:  010E   INVOKE ID:  0135
OP LEN:         104   ARG LEN:    92
Alarm Information DB
3B01 0200 0000
Report Text DB
3C01 5200 16FF 6854 7369 6920 2073 6572 6F70 7472 6C20 6E69
2065 2E31 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000
82A168000202370102020E0182045C003B01020000003C01520016FF6854
73696920207365726F7074726C206E6920652E3100000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000
```

Action

None; this log is for information only.

OM register

None

OAIN624

This log is generated when data communications message tracing has been enabled for the MIS node specified by an outgoing (OG) OAP MIS inform message. The report contains the date and time of the transmission, as well as information on the OAP class and operation headers and Information ID data block.

Note: For information on how to enable message tracing, refer to “OSSAINCI directory” on page 444.

Figure 238 Example log report for OAIN624

```
OAIN624 JUN17 10:01:41 4501 INFO OAP MIS MSG DUMP
OAP OG MIS CLASS MESSAGE
CLASSID:         8      OPTIONS: 0
LENGTH:         42      OFFSET:  18
MSG REL:         5      MSG INC: 0
SND REL:         5      SND INC: 0
SRC NODEID:     60
DEST NODEID:    38
POOLID:         61
MSG LEN:        24
Invoke    MIS OSSAIN Inform
OPERATION ID:  010D   INVOKE ID:  7FFF
OP LEN:          26   ARG LEN:    14
Information ID DB
08E0 0E00 3D00 1800 3C00 FF7F FF7F 0400 FF01
82A11E0002020A0102020C028204120008E00E003D0018003C00FF7FFF7F
0400FF01
```

Action

None; this log is for information only.

OM register

None

OAIN625

This log is generated when the DMS switch selects the voice link but cannot connect to a voice channel because all circuits are busy.

Figure 239 Example log report for OAIN625

```
OAIN625  JUN24 11:47:38 2133 INFO VOICE TRUNK BUSY
ALL CIRCUITS ARE BUSY
SN: SN_01
FUNCTION: AABS_SVC
CLLI: VL_01
```

Action

Ensure that enough voice links are available for the SN and function pair indicated in the log report. For SN functions that require a voice link connection on *every* call, it is recommended that the number of available voice links be equal to the maximum number of SN sessions that serve the function.

OM register

None

OAIN700

This log is generated when the DMS switch receives a request from an OSNM node to generate a log.

Figure 240 Example log report for OAIN700

```
OAIN700  JUN24 11:47:38 2133 INFO OSNM REPORT
    SN: NODE_3       SNID: OSNM_3
    CRITICAL RESOURCE FAILURE DUE TO SOFTWARE ERROR
```

Action

Refer to the message text in the log report.

OM register

This log is associated with OM group OAPNMTC, register NDLOG.

OAIN701

This log is generated when the DMS switch receives a request from a session pool to generate a log.

Figure 241 Example log report for OAIN701

```
OAIN701  JUN24 11:47:38 2133 INFO SESSION POOL REPORT
    SP: SP_15       SPID: 15
    NUMBER OF ACTIVE SESSIONS WILL DECREASE AT 1600
```

Action

Refer to the message text in the log report.

OM register

This log is associated with OM group OAPSPMTC, register SPLOG.

OAP logs

OAP logs are related to the Open Automated Protocol (OAP). The following table lists each OAP log in order and the page in this chapter where its description begins.

Table 142 Location of OAP log descriptions

OAP log	Page number
OAP600	page 490
OAP601	page 492
OAP602	page 493
OAP603	page 494

OAP600

The OAP subsystem generates this report when an invalid value is received in an OAP message field. If the Invoke ID of the operation in question cannot be determined, the operation is discarded (that is, not processed by OSSAIN call processing).

If the Invoke ID can be determined, a Reject operation is returned to the sender. The operation is still discarded, but it is possible that the sending node will resend or take an alternate action in this case.

Figure 242 Example log report for OAP600

```
OAP600  JUN21 12:07:05 2134 INFO OAP Protocol Violation
    CLASS:    5         INVOKE ID:    010A
    NODEID:   3         NODE NAME:    SN_1
    RELEASE:  1         INCREMENT:    0
    POOLID:   15        POOLNAME:     SP_15
    SESSION:  0         CALL ID:      #7700 #0000
    FUNCID:   12        FUNCNAME:     Func_12
    OP TYPE:  INVOKE
    OP ID:    060F      OP NAME:      Session Initiation
    FIELD ID: Arg Len   FIELD VAL:    {on next line}
    00
    RESULT:   Operation Rejected
    A182 0016 0202 0100 0202 060F 0482 0000
    010F 0006 1234 5678 9ABC
```

Note 1: The POOLID and POOLNAME fields are not displayed for node class messages.

Note 2: The SESSION, CALL ID, FUNCID, and FUNCNAME fields are displayed only for call processing class messages.

Action

When this log occurs, the service provided by the node likely is affected. The severity of the impact cannot be determined by the log report alone, because OAP messaging is simply a framework for general service and application development.

To determine the reason that invalid data is being sent by the node, perform the following checks:

- Verify that the correct software load is resident in the sending node.
- To validate the error, refer to *OSSAIN Open Automated Protocol Specification*, Q235-1. Different message fields have different validation criteria.
- Check other logs and OMs (including OM groups OAPMERRS, OAPMERRN, OAPMTYPN, and OAPMTYPS) to determine if this event is related to another problem.
- Check any related datafill and software (such as CM SOC option settings) to ensure that both the sending and receiving nodes are properly set.

OM register

When the Result field indicates Operation Rejected, this log is associated with the following OM groups and registers:

- OM group OAPMTYPS, register OGREJECT
- OM group OAPMTYPN, register NOREJCT

Depending on the values in the OP TYPE and FIELD ID fields, this log is associated with the following registers in OM group OAPMERRN:

- NINVKER
- NRESLTER
- NERRORER
- NREJCTE
- NINVOPHD
- NINVDFD

Depending on the values in the OP TYPE and FIELD ID fields, this log is associated with the following registers in OM group OAPMERRS:

- INVKERR
- RRESLTER
- RERRORER
- REJECTER
- INVLFN

- INVLOPHD
- INVDFLD

OAP601

The OAP subsystem generates this report when an unrecognized OAP operation is requested. This log indicates that sending nodes are attempting operations the switch does not recognize.

Figure 243 Example log report for OAP601

```
OAP601  JUN24 11:47:38 2133 INFO Unrecognized OAP Operation
    CLASS:    5         INVOKE ID:   0200
    NODEID:   3         NODE NAME:   SN_1
    RELEASE:  1         INCREMENT:   0
    POOLID:   15        POOLNAME:    SP_15
    SESSION:  20        CALL ID:     #7700 #0000
    FUNCID:   32        FUNCNAME:    FUNC_32
    OP ID:    1700
    82A1 000C 0202 0200 0202 0700 8204 0008
```

Note 1: The POOLID and POOLNAME fields are not displayed for node class messages.

Note 2: The SESSION, CALL ID, FUNCID, and FUNCNAME fields are displayed only for call processing class messages.

Action

To determine the reason that invalid data is being sent by the node, perform the following checks:

- Verify that the correct software load is resident in the sending node.
- To validate the error, refer to *OSSAIN Open Automated Protocol Specification*, Q235-1. Different message fields have different validation criteria.
- Check other logs and OMs (including OM groups OAPMERRS, OAPMERRN, OAPMTYPN, and OAPMTYPS) to determine if this event is related to another problem.
- Check any related datafill and software (such as CM SOC option settings) to ensure that both the sending and receiving nodes are properly set.

OM register

This log is associated with the following OM groups and registers:

- OM group OAPMTYPS, register OGREJECT
- OM group OAPMTYPN, register NOREJCT
- OM group OAPMERRS, register UNKWNOP
- OM group OAPMERRN, register NUNKNOP

OAP602

The OAP subsystem generates this report when an OAP operation request from a sending node either includes a data block that is not recognized as being associated with the requested operation/response, or excludes a data block that is required (mandatory) with the requested operation.

Figure 244 Example log report for OAP602

```
OAP602  JUN28 10:47:08 2137 INFO Data Block Error
    CLASS:    5         INVOKE ID:   010A
    NODEID:   3         NODE NAME:   SN_1
    RELEASE:  1         INCREMENT:   0
    POOLID:   15        POOLNAME:    SP_15
    SESSION:  0         CALL ID:     #7700 #0000
    FUNCID:   12        FUNCNAME:    Func_12
    OP TYPE:  INVOKE
    OP ID:    060F      OP NAME:     Session Initiation
    DB ID:    0110      RESULT:      Unexpected
    FIELD ID: 46        FIELD VAL:   {on next line}
    0110
    0000 0002 2000
```

Note 1: The POOLID and POOLNAME fields are not displayed for node class messages.

Note 2: The SESSION, CALL ID, FUNCID, and FUNCNAME fields are displayed only for call processing class messages.

Action

To determine the reason that invalid data is being sent by the node, perform the following checks:

- Verify that the correct software load is resident in the sending node.
- To validate the error, refer to *OSSAIN Open Automated Protocol Specification*, Q235-1. Different message fields have different validation criteria.
- Check other logs and OMs (including OM groups OAPMERRS, OAPMERRN, OAPMTYPN, and OAPMTYPS) to determine if this event is related to another problem.
- Check any related datafill and software (such as CM SOC option settings) to ensure that both the sending and receiving nodes are properly set.

OM register

This log is associated with the following OM groups and registers:

- OM group OAPMTYPS, register OGREJCT
- OM group OAPMTYPN, register NOREJCT
- OM group OAPMERRS, register UNKWNDB (or MISNGDB)
- OM group OAPMERRN, register NUNKNDB (or NMSNGDB)

OAP603

The OAP subsystem generates this report when an OAP Reject is received from another node. The log indicates that the sending node could not interpret a Return Result message it received from the switch.

Figure 245 Example log report for OAP603

```
OAP603  JUN24 11:47:38 2133 INFO OAP Reject Received
    CLASS:    5         INVOKE ID:   010A
    NODEID:   2         NODE NAME:   SN_1
    RELEASE:  1         INCREMENT:   0
    POOLID:   15        POOLNAME:    SP_15
    SESSION:  10        CALL ID:     #7700 #0000
    FUNCID:   12        FUNCNAME:    Func_12
    PROBLEM TYPE: ReturnResult Problem
    REJECT REASON: Mistyped Result
```

Note 1: The POOLID and POOLNAME fields are not displayed for node class messages.

Note 2: The SESSION, CALL ID, FUNCID, and FUNCNAME fields are displayed only for call processing class messages.

Refer to the following list of values for the Problem Type field and the Reject Reason field in the log report.

Table 143 Problem types and reject reasons

Problem type	Reject reason
0 - General Problem	0 - Unrecognized APDU 1 - Mistyped APDU 2 - BadlyStructured APDU
1 - Invoke Problem	0 - Duplicate Invocation 1 - Unrecognized Operation 2 - Mistyped Argument 3 - Resource Limitation 4 - Initiator Releasing 5 - Unrecognized LinkID 6 - Linked Response Unexpected 7 - Unexpected Child Operation
2 - ReturnResult Problem	0 - Unrecognized Invocation 1 - ResultResponse Expected 2 - Mistyped Result
3 - ReturnError Problem	0 - Unrecognized Invocation 1 - ErrorResponse Expected 2 - Unrecognized Error 3 - Unexpected Error 4 - Mistyped Parameter

Action

When this log occurs, the service provided by the node likely is affected. The severity of the impact cannot be determined by the log report alone, because OAP messaging is simply a framework for general service and application development.

To determine the reason that invalid data is being sent by the node, perform the following checks:

- Check the source node logs (or equivalent error reporting) for details of the operation Reject.
- Verify that the correct software load is resident in the sending node.

  Note: A Protocol Version reject is normal when a node is brought in service (as part of normal protocol level establishment) or when the DMS protocol version is changed. Other cases should be investigated.

- To validate the error, refer to *OSSAIN Open Automated Protocol Specification*, Q235-1. Different message fields have different validation criteria.
- Check other logs and OMs (including OM groups OAPMERRS, OAPMERRN, OAPMTYPN, and OAPMTYPS) to determine if this event is related to another problem.
- Check any related datafill and software (such as CM SOC option settings) to ensure that both the sending and receiving nodes are properly set.

OM register

This log is associated with the following OM groups and registers:

- OM group OAPMTYPS, register INREJCT
- OM group OAPMTYPN, register NIREJCT

OSAC logs

OSAC logs are related to nodes and session pools in a centralized OSSAIN (OSAC) network. The following table lists each OSAC log in order and the page in this chapter where its description begins.

Table 144 Location of OSAC log descriptions

OSAC log	Page number
OSAC200	page 497
OSAC201	page 497
OSAC202	page 498
OSAC203	page 498
OSAC204	page 498
OSAC300	page 499
OSAC600	page 499
OSAC601	page 500

OSAC200

This log is generated at the OSAC host switch when it receives an OSAC session request that has an invalid function identifier (ID). The function ID can be invalid because it is not datafilled in table OAFUNDEF, or because it is not an OSSAIN function.

Figure 246 Example log report for OSAC200

```
OSAC200 JAN24 07:46:17 8701 INFO BAD OAFUNDEF DATAFILL
Requesting Node: Remote_1
Requesting Nodeid: 2
FUNCID:  23
TEXT: Function ID not datafilled in Table OAFUNDEF
```

Action

In the OSAC host switch, datafill the function ID as an OSSAIN function in table OAFUNDEF. Ensure that the OSAC host switch provides the CAM for the function ID.

OM register

None

OSAC201

This log is generated at the OSAC remote switch when it receives an invalid session pool ID or an invalid session ID from the OSAC host switch. The session pool ID can be invalid because it is not datafilled in table OASESNPL, or because it is not datafilled as a subscriber origination session pool. The session ID can be invalid because it exceeds the maximum session limit in table OASESNPL.

Figure 247 Example log report for OSAC201

```
OSAC201 JAN24 07:46:17 8701 INFO BAD OASESNPL DATAFILL
SESNPL:  14
SESN:  20
FUNCID: 3
TEXT: Session Pool ID not datafilled in Table OASESNPL
```

Action

Ensure that the datafill for the session pool is parallel between the OSAC remote switch and the OSAC host switch.

OM register

None

OSAC202

This log is generated at the OSAC remote switch when it receives an error response from the OSAC host switch that indicates a function datafill mismatch.

Figure 248 Example log report for OSAC202

```
OSAC202 JAN24 07:46:17 8701 INFO BAD DATAFILL
NODE:  TEST_SN
NODEID:  20
SESNPL: TEST_SP 4
FUNCID: 31
TEXT: Function datafill mismatch with Host
```

Action

Ensure that the datafill for the function is parallel between the OSAC remote switch and the OSAC host switch.

OM register

None

OSAC203

This log is generated at the OSAC host switch when it receives a voice connect request from the OSAC remote switch that has missing datafill in table OSCVLGRP.

Figure 249 Example log report for OSAC203

```
OSAC203 JAN24 07:46:17 8701 INFO BAD OSCVLGRP DATAFILL
DATAFILL TABLE OSCVLGRP WITH
NODEID: AABS_SN
```

Action

Check the datafill for table OSCVLGRP at the OSAC host switch.

OM register

None

OSAC204

This log is generated at the OSAC host switch when it receives a voice connect request from the OSAC remote switch that has missing datafill in table OAVLMAP.

Figure 250 Example log report for OSAC204

```
OSAC204 JAN24 07:46:17 8701 INFO BAD OAVLMAP DATAFILL
DATAFILL TABLE OAVLMAP WITH
NODEID: REMOTE_21
TRUNK MEMBER: 4
```

Action

Check the datafill for table OAVLMAP at the OSAC host switch.

OM register

None

OSAC300

This log is generated at the OSAC host switch or the OSAC remote switch when it receives an unexpected message.

Figure 251 Example log report for OSAC300

```
OSAC300 JAN24 07:46:17 8701 INFO UNEXPECTED MESSAGE
CALLID: 0023 0011
SN: AABS_SN             SNID: OSNM 6
FN: AABS                SESNPL: 6 AABS_SP
OSAC_SWITCH: Remote_2   OSAC_SP: 20 REMOTE_2
SNVL: CKT OSSAINVL 22   OSACVL: CKT OSACVL 22
TEXT: UNKNOWN MESSAGE
```

OSAC600

This log is generated at the OSAC host switch if the switch needs to take a call down for any reason.

Figure 252 Example log report for OSAC600

```
OSAC600 JAN24 07:46:17 8701 INFO CALL TAKE DOWN
CALLID: 0023 0011
SN: AABS_SN             SNID: OSNM 6
FN: AABS                SESNPL: 6 AABS_SP
OSAC_Remote: Remote_2   OSAC_SP: 20 REMOTE_2
SNVL: CKT OSSAINVL 22   OSACVL: CKT OSACVL 22
TEXT: TERMINAL FORCE RELEASED
```

Action

Ensure that the SN is not holding the connection for the call. Check logs in the OSAC remote switch that are associated with the CALLID.

OM register

This log is associated with OM group OASVNDCP, register OSCCLERR.

OSAC601

This log is generated at the OSAC host switch and the OSAC remote switch if a check for parallel datafill fails. Parallel datafill is checked during RTS, TST, and audit of OSAC nodes and session pools, and during RTS, TST, and audit of OSN nodes and session pools.

Figure 253 Example log report for OSAC601

```
OSAC601 JAN24 07:46:17 8701 INFO DATAFILL MISMATCH
SN_NAME:AABS_SN             SNID: 6
SP_NAME:AABSSP              SPID: 12
OSAC_NODE_NAME: Remote_2    OSAC_SNID: 20
OSAC_SP_NAME:               OSAC_SPID:
TABLE: OASESNPL
```

Action

Check the datafill at both the OSAC host switch and the OSAC remote switch for the node and session pool specified in the log. Ensure that datafill for the table specified (OASESNPL or OANODNAM) matches on both switches.

Refer to the following matrix for the type of parallel datafill check done for OSAC and OSN nodes and session pools.

Table 145 Parallel datafill checking

Type of parallel datafill check	Node name	Node ID	Pool name	Pool ID	OSAC node name	OSAC node ID	OSAC pool name	OSAC pool ID
OSN node	X	X			X	X		
OSN session pool	X	X	X	X	X	X		
OSAC node					X	X		
OSAC session pool					X	X	X	X

OM register

None

EXT logs

EXT logs are related to maintenance of external nodes. In the OSSAIN network, EXT logs are associated with alarms requested by an OSNM node or an OSNM session pool. OSSAIN modifies information provided by EXT106, EXT107, and EXT108 log reports.

The following table lists each EXT log in order and the page in this chapter where its description begins.

Table 146 Location of EXT log descriptions

EXT log	Page number
EXT106	page 501
EXT107	page 502
EXT108	page 502

Note: The DMS switch also generates an OAIN700 log (for OSNM nodes) or an OAIN701 log (for OSNM session pools) when it receives a request to generate an EXT log report. The OAIN log specifies the reason for the alarm. Refer to page 489, for more information on OAIN700 and OAIN701.

EXT106

This log is generated when the DMS switch receives a request from an OSNM node or an OSNM session pool to generate a minor alarm. The log report displays the ID of the SN or session pool and whether the alarm is on or off.

Figure 254 Example log report for EXT106

```
EXT106 JAN24 07:46:17 8701 INFO SP_MINOR ON
  SPID: 12
```

Action

Refer to the SN service documentation for the OSNM or session pool identified in the log.

OM register

This log is associated with OM group OAPNMTC, register NDALARM (for a node alarm request) or OM group OAPSPMTC, register SPALARM (for a session pool alarm request).

EXT107

This log is generated when the DMS switch receives a request from an OSNM node or an OSNM session pool to generate a major alarm. The log report displays the ID of the SN or session pool and whether the alarm is on or off.

Figure 255 Example log report for EXT107

```
EXT107 JAN24 07:46:17 8701 INFO SN_MAJOR OFF
  SNID: OSNM 6
```

Action

Refer to the SN service documentation for the SN or session pool identified in the log.

OM register

This log is associated with OM group OAPNMTC, register NDALARM (for a node alarm request) or OM group OAPSPMTC, register SPALARM (for a session pool alarm request).

EXT108

This log is generated when the DMS switch receives a request from an OSNM node or an OSNM session pool to generate a critical alarm. The log report displays the ID of the SN or session pool and whether the alarm is on or off.

Figure 256 Example log report for EXT108

```
EXT108 JAN24 07:46:17 8701 INFO SN_CRITICAL ON
  SNID: OSN 2
```

Action

Refer to the SN service documentation for the SN or session pool identified in the log.

OM register

This log is associated with OM group OAPNMTC, register NDALARM (for a node alarm request) or OM group OAPSPMTC, register SPALARM (for a session pool alarm request).

AUD logs

AUD logs are related to audit subsystem. The following table lists each AUD log in order and the page in this chapter where its description begins.

Table 147 Location of AUD log descriptions

AUD log	Page number
AUD607	page 503
AUD609	page 504
AUD629	page 504

AUD607

This log is generated when an OSSAIN call traps or dies and the call has an attached OSSAINRU extension block.

Figure 257 Example log report for AUD607

```
AUD607 JAN24 07:46:15 8700 INFO EXT DUMP   0302 0011
   0321 6342 8681 010C 0200 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000 8000 AAAA AAAA AAAA AAAA
   00AA AAAA AAAA AAAA AAAA 80AA 0050 5991 0420 1647
   4530 270F 0000 0000 0000 0000 3201 0000 3201 0000
   0104 0000 0000 0000 0000 0000 0000 2AAA 0000 8000
   0843 0000 CE5F FAE4 0000 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000 8000 AAAA AAAA AAAA AAAA
   00AA AAAA AAAA AAAA AAAA 80AA 0050 5991 0420 1647
   0E30 E71F 0000 0000 0000 0000 3201 0000 3201 0000
   0467 0000 0000 0000 0000 0000 0000 2BBC 0000 1800
```

Action

Collect all log reports occurring at approximately the same time that AUD607 is generated. In particular, AUD395 and AUD398 logs typically accompany the AUD607 log. Also capture any traps or SWERRs associated with the logs.

OM register

None

AUD609

This log is generated when a TOPS call traps or dies and the call has an attached TOPSRU2 extension block.

Figure 258 Example log report for AUD609

```
AUD609 JAN24 07:46:15 8700 INFO EXT DUMP  0302 0011
   0321 6342 8681 010C 0200 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000 8000 AAAA AAAA AAAA AAAA
   00AA AAAA AAAA AAAA AAAA 80AA 0050 5991 0420 1647
   4530 270F 0000 0000 0000 0000 3201 0000 3201 0000
   0104 0000 0000 0000 0000 0000 0000 2AAA 0000 8000
   0843 0000 CE5F FAE4 0000 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000
```

Action

Collect all log reports occurring at approximately the same time that AUD609 is generated. In particular, AUD395 and AUD398 logs typically accompany the AUD609 log. Also capture any traps or SWERRs associated with the logs.

OM register

None

AUD629

This log is generated at the OSAC host switch when an OSAC call traps or dies and the call has an attached OSACRU extension block.

Figure 259 Example log report for AUD629

```
AUD629 JAN24 07:46:15 8700 INFO EXT DUMP  0302 0011
   0321 6342 8681 010C 0200 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000 8000 AAAA AAAA AAAA AAAA
   00AA AAAA AAAA AAAA AAAA 80AA 0050 5991 0420 1647
   4530 270F 0000 0000 0000 0000 3201 0000 3201 0000
   0104 0000 0000 0000 0000 0000 0000 2AAA 0000 8000
   0843 0000 CE5F FAE4 0000 0000 0000 0000 0000 0000
   0000 0000 0000 0000 0000
```

Action

Collect all log reports occurring at approximately the same time that AUD629 is generated. Also capture any traps or SWERRs associated with the logs.

OM register

None

PM logs

PM logs are related to peripheral modules (PM). For OSSAIN, PM logs apply to nodes that are datafilled in table OANODINV, including OSN, OSNM, and OSAC nodes.

The following table lists each PM log in order and the page in this chapter where its description begins.

Table 148 Location of PM log descriptions

PM log	Page number
PM100	page 505
PM102	page 506
PM103	page 506
PM104	page 507
PM105	page 507
PM106	page 507
PM128	page 508

PM100

This log is generated when a node fails the test command.

Figure 260 Example log report for PM100

```
PM100 JUN27 10:41:02 5678 FAIL DIAG OSNM 0
```

Action

Perform the following actions:

- Ensure all DMS EIUs are in service. Or, if the node is datafilled to use XA-Core Ethernet interface, ensure that the XA-Core Ethernet interface cards are in service.
- Ensure all cabling is intact.
- Check for SN faults.

Refer to the SN documentation to run further diagnostics to correct the problem.

OM register

This log is associated with OM group OAINNODE, register NTSTFAIL.

PM102

This log is generated when a node goes system busy. For MIS nodes, this log is generated when the Ethernet connection is lost.

Figure 261 Example log report for PM102

```
PM102 JUN27 10:41:02 5678 SYSB OSNM 1
SYSB from INSV
```

Action

Perform the following actions:

- Ensure all DMS EIUs are in service. Or, if the node is datafilled to use XA-Core Ethernet interface, ensure that the XA-Core Ethernet interface cards are in service.
- Ensure all cabling is intact.
- Check for SN faults.

One OAIN507 log is generated for every in-service session pool datafilled on the SYSB node. MANB and RTS the node.

OM register

This log is associated with the following registers in OM group OAINNODE:

- NRTSFAIL
- NDSYSB
- NAUDFAIL

PM103

This log is generated when a node is placed in an offline state or datafilled in table OANODINV.

Figure 262 Example log report for PM103

```
PM103 JUN27 10:41:21 5678 OFFL OSN 0
OFFL from MBSY
```

Action

None

OM register

None

PM104

This log is generated when a node is deleted from table OANODINV.

Figure 263 Example log report for PM104

```
PM104 JUN27 10:41:21 5678 UNEQ OSNM 0
UNEQ from OFFL
```

Action

None

OM register

None

PM105

This log is generated when a node is manually busied.

Figure 264 Example log report for PM105

```
PM105 JUN27 10:41:21 5678 MANB OSNM 0
MANB from INSV
```

Action

None

OM register

This log is associated with OM group OAINNODE, register NDMANB.

PM106

This log is generated when a node comes into service.

Figure 265 Example log report for PM106

```
PM106 JUN27 10:41:21 5678 RTS OSNM 0
INSV from MANB
```

Action

None

OM register

This log is associated with OM group OAINNODE, register NDINSV.

PM128

This log is generated when a node goes from INSV to ISTB because a session pool supported by the node is not in the offline state or in service state.

Figure 266 Example log report for PM128

```
PM128 JUN27 10:41:21 5678 TBL ISTB OSNM 0
ISTB from INSV
```

Action

Check the corresponding OAIN logs as follows:

- If an OAIN502 log is generated, the session pool is SYSB. Follow recovery procedures for that session pool.
- If an OAIN505 log is generated, the session pool is MANB and no action is required.

OM register

This log is associated with OM group OAINNODE, register NDISTB.

DAS logs

DAS logs are related to the directory assistance system (DAS). The following table lists each DAS log in order and the page in this chapter where its description begins.

Table 149 Location of DAS log descriptions

DAS log	Page number
DAS105	page 508
DAS106	page 509

DAS105

This log is generated when the switch fails to obtain a required system resource for transferring a call from the DAS to OSSAIN.

Figure 267 Example log report for DAS105

```
DAS105 JUN21 12:07:05 2134 TBL TOPS DIRECTORY ASSISTANCE
      PROTOCOL = CCI                   INFO = 00
      ORIG CKT = TBELLIC1              DASCALLID = 0014
      REASON = OSSAIN Not SOCed On
      CONTEXT TRANSFER RESOURCE PROBLEM
```

Note: The INFO field in the log report is not supported in the first release of OSSAIN.

Action

The action depends on which system resource is unavailable, as follows:

- insufficient OSSAIN recording units

  Check the value of office parameter OSSAIN_NUM_RU in table OFCENG. It may need to be increased. For more information, refer to Chapter 6: “OSSAIN engineering.”

- OSSAIN not SOCed on

  Verify that the SOC option indicated in the log has the ON state assigned. For more information, refer to Chapter 8: “OSSAIN software optionality control.”

- invalid queuing system

  Ensure that the QMS queuing system is being used. For more information on QMS, refer to the *Translations Guide*.

OM register

This log is associated with OM group DAMISC, register XFERFAIL.

DAS106

This log is generated when the DMS switch receives an invalid control list identifier (ID) from the DAS in a Transfer with Context message. If the control list ID is not datafilled in table OACTLDEF, the switch returns a Call Status message to the DAS that indicates failure.

Figure 268 Example log report for DAS106

```
DAS106 JUN21 12:07:05 2134 INFO TOPS DIRECTORY ASSISTANCE
      PROTOCOL = CCI                  INFO = 00
      ORIG CKT = TBELLIC1             DASCALLID = 0015
      REASON = CONTROL LIST NOT DATAFILLED
      CONTROL LIST DATAFILL PROBLEM
```

Note: The INFO field in the log report is not supported in the first release of OSSAIN.

Action

Determine the value of the control list ID in the Transfer with Context message to ensure that it agrees with the value of the control list ID in table OACTLDEF (OACTLNUM field).

Note: ID values that are outside the range 0 to 4095 are treated as protocol errors and instead produce a TCCI101 log.

OM register

This log is associated with OM group DAMISC, register XFERFAIL.

TCCI logs

TCCI logs are related to the TOPS CCI protocol (TCCI). The following table lists each TCCI log in order and the page in this chapter where its description begins.

Table 150 Location of TCCI log descriptions

TCCI log	Page number
TCCI100	page 510
TCCI101	page 511
TCCI102	page 511

TCCI100

This log is generated for one of the following reasons:

- DA tracing is turned on.
- The switch attempts to send an improperly formatted message to the DAS.
- The switch receives an improperly formatted message from the DAS.

Figure 269 Example log report for TCCI100

```
TCCI100 MAR03 10:38:29 5700 INFO TOPS CCI PROTOCOL ERROR
ERROR   = TRACING ACTIVATED
STATE   = 14
DATABASE= STUB    0
HEX MSG =010FFFFF03010024021659022143 14F1FFFFFFFF0FFFFFFFFF
FFFF00FF020000000040400E00FFFFFFFFFFFF
FORMATTED MSG:
 MSG TYPE  = CALL BEGIN
 SWITCHID  = 15
 DAS AREA  = FFFF
 DETAIL    = 3
 DAS CALLID= 1
 POS/ARU ID= 548
 CLG/REQ DN= 619-520-1234
 CLD DN    =     411-FFFF
 ANN NUMBER= 255
 ORIG INFO = FFF0
```

Action

Users should contact DAS system support personnel when this log is generated.

OM register

None

TCCI101

This log is generated for one of the following reasons:

- DA tracing is turned on and the switch receives a Transfer with Context message from the DAS.
- The switch receives an improperly formatted Transfer with Context message from the DAS.

Figure 270 Example log report for TCCI101

```
TCCI101 MAR03 10:38:31 5910 INFO TOPS CCI PROTOCOL ERROR
ERROR   = TRACING ACTIVATED
STATE   = 14
DATABASE= STUB    0
HEX MSG = 150F0000FF0100010022000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F2021FFFF
FORMATTED MSG:
 MSG TYPE      = XFER WITH CNTX
 SWITCHID      = 15
 DAS AREA      = 0000
 DETAIL        = 255
 DAS CALLID    = 1
 CONTROL LISTID= 1
 CONTEXT BLOCK = 0022020104030605080707A090C0B0E0D100F12111413
161518171A191C1B1E1D201F4A21
```

Action

Users should contact DAS system support personnel when this log is generated.

OM register

None

TCCI102

This log is generated for one of the following reasons:

- DA tracing is turned on and the switch sends a Context Block message to the DAS.
- The switch attempts to send an improperly formatted Context Block message to the DAS.

Figure 271 Example log report for TCCI102

```
TCCI102 MAR03 10:38:29 5800 INFO TOPS CCI PROTOCOL ERROR
ERROR   = TRACING ACTIVATED
STATE   = 14
DATABASE= STUB    0
HEX MSG = 160FFFFFFF0100220001020304050607080 90A0B0C0D0E0F1011
12131415161718191A1B1C1D1E1F2021FFFFFFFF
FORMATTED MSG:
 MSG TYPE     = CONTEXT BLOCK
 SWITCHID     = 15
 DAS AREA     = FFFF
 DETAIL       = 255
 DAS CALLID   = 1
 CONTEXT BLOCK= 00220201040306050807 0A090C0B0E0D100F12111413
161518171A191C1B1E1D201F0021
```

Action

Users should contact DAS system support personnel when this log is generated.

OM register

None

QMIS logs

QMIS logs are related to the QMS MIS application. The OSSAIN QMS MIS application generates a QMIS101 log.

QMIS101

This log is generated when the Ethernet connection is unable to send a QMIS MIS buffer because of insufficient resources. The log report contains the date and time that the transmission failed, as well as the name and ID number of the MIS node datafilled in table QMSMIS.

Figure 272 Example log report for QMIS101

```
QMIS101 NOV03 15:46:37 2400 INFO QMS_MIS_ETHERNET_SEND_FAIL
   APPLN = OSSAIN  NODE = MISNODE  NODEID = 24
```

Action

Users should investigate the data connectivity between the DMS switch and the MIS node.

OM register

None

TOPS logs

TOPS logs are related to TOPS processing. The following table lists each TOPS log in order and the page in this chapter where its description begins.

Table 151 Location of TOPS log descriptions

TOPS log	Page number
TOPS129	page 513
TOPS132	page 514
TOPS 610	page 515

TOPS129

This log is generated when the previous switch cannot perform RLT after the TOPS switch initiates the RLT request.

Figure 273 Example log report for TOPS129

```
TOPS129 JAN24 07:46:17 8701 INFO RLT REQUEST FAILED
INCOMING TRK = CKT ISUPIC 2
OUTGOING TRK = CKT ISUPOG 3
REASON: 17
```

Action

Ensure that datafill in table ISUPTRK is correct in the TOPS switch. Also check network datafill. This log may indicate a problem at the previous switch.

OM register

This log is associated with the following registers in OM group TOPSRLT2:

- TRNSFAIL
- BRDGFAIL

TOPS132

This log is generated when OSSAIN alternate routing routes the call to treatment or cannot route the call at all. The log is generated regardless of the treatment applied and regardless of whether the call would have routed to treatment without OSSAIN alternate routing.

Figure 274 Example log report for TOPS132

```
TOPS132 FEB28 07:46:17 8701 INFO ROUTE ERROR
CALLID: 0302 0011
TRUNK: T1TI00
CONNECTION: CALLED
DN: 9193629345
RI: 2046
RN: 9193634824
SN: NODE_1
SESPL: 2
```

Action

Ensure that the alternate route sent by the SN in the RN or RI is set up correctly in the switch routing tables. Another option is for the operating company to disable OSSAIN alternate routing through SOC OSAN0102.

Note: A TOPS132 log is generated every time a TRK138 log is generated when an RI or RN is applied to a connection, even if the RI or RN did not cause the log. Based solely on a TOPS132 log, users should not assume that the alternate route actually caused a treatment. Refer to *Log Report Reference Manual* for a description of the TRK138 log.

OM register

None

TOPS610

This log is generated when a special LRN call is received over a trunk other than a North American ISUP trunk. CO type SLRN is unsupported on any non-NA ISUP trunk and the call will be routed to treatment. Log 610 is generated before the call is sent to treatment.

Figure 275 Example log report for TOPS610

```
TOPS610 JUL23 07:41:02 0900 INFO Bad Rte For Special LRN Call
INCOMING TRK: CKT ETSIISUPIC5 1
CALLING DN: 619-320-4550
SPECIAL LRN: 619-472-1098
CT4Q:
```

Action

Verify translations for the call in the previous switch. Calls routing to the specified LRN should arrive over NA ISUP trunks.

OM register

None

Chapter 12: OSSAIN operational measurements

This chapter provides information on operational measurements (OM) for OSSAIN. For each OM group there is a brief description, a list of registers, an OMSHOW example, and a list of any associated OM groups.

Note: For complete information on all OMs for the DMS switch, refer to *Operational Measurements Reference Manual*. Some of the TOPS OM groups that are especially relevant to OSSAIN, but are not documented here because they are not OSSAIN-specific, are

- TDGTHRU (TOPS Datagram Throughput). Pegs the numbers of UDP messages of various sizes sent and received over various interfaces for various applications, including OSSAIN.
- QMSACT (Queue Management System Activity). In OSAC host and standalone OSSAIN switches, pegs counts related to call and agent queue management.

The following table lists each OM group in alphabetical order and the page in this chapter where its description begins.

Table 152 Alphabetical reference for OSSAIN OM descriptions

OM group name	Page number
AABSHAND	page 519
CTRYDIR	page 520
DAMISC	page 521
EXT	page 522
OADATCOM	page 522
OAFLTRIG	page 524
OAINNODE	page 525
OAINQMS	page 526
OAINRTE	page 528
OANODEDC	page 529

Table 152 Alphabetical reference for OSSAIN OM descriptions

OM group name	Page number
OAPCALP1	page 531
OAPCALP2	page 533
OAPCALP3	page 535
OAPCALP4	page 537
OAPCALP5	page 539
OAPCALP6	page 541
OAPCALP7	page 543
OAPCALP8	page 545
OAPCALP9	page 547
OAPCP10	page 549
OAPMERRN	page 551
OAPMERRS	page 552
OAPMTYPN	page 554
OAPMTYPS	page 556
OAPNMIS	page 558
OAPNMTC	page 558
OAPSPMTC	page 561
OASNPLDC	page 563
OASNPOOL	page 565
OASVNDCP	page 566
OSACCP1	page 566
OSACCP2	page 568
OSACND	page 570
OSACSP	page 572
OSNND	page 574
OSNSP	page 575
QMSMIS	page 576
TOPPACT3	page 578
TOPPACT4	page 579
TOPPDID5	page 579

Table 152 Alphabetical reference for OSSAIN OM descriptions

OM group name	Page number
TOPSRLT2	page 581

AABSHAND

OM group AABSHAND (Automated Alternate Billing Service Handoff) provides peg counts for attempts by operators and SNs to hand off calls for billing acceptance. Operators can hand off calls to OSSAIN SNs only. (Before support for the AABS system was discontinued, they could also hand off calls to AABS.) An OSSAIN SN can attempt to hand off a call to AABS, but it will route either to MCCS/ACCS or to a TOPS operator.

Note: Because operator handoffs to an OSSAIN SN always succeed, there are no OM registers that count successes or failures.

Twelve registers in OM group AABSHAND apply to OSSAIN. The following table describes these registers.

Table 153 OM group AABSHAND

Register	Description
SHATZMIN	SN handoff attempt for 0 minus call
SHATANIF	SN handoff attempt for ANI failure call
SHATONI	SN handoff attempt for ONI call
SHATMISC	SN handoff attempt for call other than 0 minus, ANI failure, or ONI
SHANCOSC	SN handoff success for collect call
SHANTHSC	SN handoff success for bill to third number call
SHANCOFL	SN handoff failure for collect call
SHANTHFL	SN handoff failure for bill to third number call
OHNDZMIN	Operator handoff to SN for 0 minus call
OHNDANIF	Operator handoff to SN for ANI failure call
OHNDONI	Operator handoff to SN for ONI call
OHNDMISC	Operator handoff to SN for call other than 0 minus, ANI failure, or ONI

The following figure shows an example for OM group AABSHAND.

Figure 276 MAP display example for OM group AABSHAND

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:        18;

     HATTZMIN    HATTANIF     HATTONI    HATTMISC
     HANDCOSC    HANDTHSC    HANDCOFL    HANDTHFL
     SHATZMIN    SHATANIF     SHATONI    SHATMISC
     SHANCOSC    SHANTHSC    SHANCOFL    SHANTHFL
     OHNDZMIN    OHNDANIF     OHNDONI    OHNDMISC

            0           0           0           0
            0           0           0           0
           30           3           1           0
           25           5           2           1
            4           1           0           0
```

Associated OM groups

None

CTRYDIR

OM group CTRYDIR (Country Direct) provides peg counts for TOPS and OSSAIN country direct calls.

Four registers in OM group CTRYDIR apply to OSSAIN. The following table describes these registers.

Table 154 OM group CTRYDIR

Register	Description
CDIRQD	Country direct queued. This register is pegged each time a TOPS country direct call can be handled by an operator or by an SN. This includes calls that are queued as well as those that are handled immediately.
CDIRQD2	Country direct queued extension register
CDIRSN	Country direct SN. This register is pegged each time a TOPS country direct call is routed to an SN.
CDIRSN2	Country direct SN extension register

The following figure shows an example for OM group CTRYDIR.

Figure 277 MAP display example for OM group CTRYDIR

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       CDIRCO     CDIRCO2      CDIRQD     CDIRQD2
       CDIRSV     CDIRSV2      CDIRHA     CDIRHA2
       CDIRFL     CDIRFL2      CDCSAL     CDCSAL2
       CDCSBL     CDCSBL2     CDCSSUB    CDCSSUB2
       CDIRSN     CDIRSN2

0          10           0           9           0
           11           0           8           0
           12           0           7           0
           13           0           6           0
            5           0
```

Associated OM groups

None

DAMISC

OM group DAMISC (Directory Assistance Miscellaneous) provides peg counts for miscellaneous events in TOPS DA service, such as auto-intercept failures and communication failures with the DAS.

Two registers in OM group DAMISC apply to OSSAIN. The following table describes these registers.

Table 155 OM group DAMISC

Register	Description
XFERCNTX	Transfer with context. This register is pegged each time the switch successfully transfers a DA call to OSSAIN.
XFERFAIL	Transfer with context fail. This register is pegged each time the switch fails to transfer a DA call to OSSAIN.

The following figure shows an example for OM group DAMISC.

Figure 278 MAP display example for OM group DAMISC

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

     AUTINTFL    OHIMPOSE    DAMCCSUC    DAACTSUC
     INTFLTTO    DAAABSUC    XFERCNTX    XFERFAIL

0           0           0           0           0
            0           0           0           0
```

Associated OM groups

None

EXT

OM group EXT (Extension Block) counts the number of times an extension blocks is used. OSSAIN does not change the registers for OM group EXT; however OSSAIN and OSAC extension blocks add the following two tuples:

- OSSAINRU
- OSACRU

The following figure shows an example for OM group EXT.

Figure 279 MAP display example for OM group EXT

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      EXTSEIZ     EXTOVFL       EXTHI    EXTSEIZ2
       EXTHI2

139 OSSAINRU
          100           0           0           0
            0

161 OSACRU
          100           0           0           0
            0
```

Associated OM groups

None

OADATCOM

OM group OADATCOM (OSSAIN Data Communications) provides peg counts for OSSAIN data communications messaging events. It counts the total number of messages sent from the compute module (CM) to other nodes and the total number of messages received by the CM from other nodes. The messages are divided into success counts and failure counts.

The following table describes each register.

Table 156 OM group OADATCOM

Register	Description
OMSGSND	OSSAIN message send requested. This register is pegged each time the data communications software is requested to send a message. It includes requests from both call and maintenance processes.
OMSGSND2	OSSAIN message send requested extension register

Table 156 OM group OADATCOM

Register	Description
OMSGRCV	OSSAIN message received. This register is pegged each time an incoming message from an external node is received. It includes messages from both call and maintenance processes.
OMSGRCV2	OSSAIN message received extension register
OMSGSNSC	OSSAIN message send success. This register is pegged when the data communications software successfully processes an outgoing message. ***Note:*** OSSAIN call processing and maintenance messages do not have guaranteed messaging, whereas OSSAIN QMS MIS uses TCP for guaranteed messaging. Pegging this register does not indicate that the message actually arrived at the destination node.
OMSGSNS2	OSSAIN message send success extension register
OMSGRCSC	OSSAIN message received success. This register is pegged when the data communications software successfully processes an incoming message.
OMSGRCS2	OSSAIN message received success extension register
OMSGSNFL	OSSAIN message send failure. This register is pegged each time data communications encounters an error when sending an outgoing message. The error can be caused by a transport layer failure. (For a list of other possible reasons for failure, refer to Note 1.)
OMSGRCFL	OSSAIN message received failure. This register is pegged each time data communications encounters an error when forwarding an external node originated message to the destination DMS process. The error can be caused by a failure in the DMS internal messaging system or data transport interface. (For a list of other possible reasons for failure, refer to Note 2.)
OSNDRTFL	OSSAIN message send route failure. This register is pegged each time the data communications software is unable to determine the destination of an outgoing message. (See Note 1.)

Table 156 OM group OADATCOM

Register	Description
ORCVRTFL	OSSAIN message received route failure. This register is pegged each time the data communications software is unable to determine the destination of an external node originated message. (See Note 2.)
Note 1: Possible reasons for failure include invalid node identifier, invalid session pool identifier, invalid session identifier, pool/node identifier mismatch, and corrupted message. ***Note 2:*** Possible reasons for failure include invalid protocol version, invalid class header identifier, invalid operation offset, invalid message length, invalid node identifier, invalid session pool identifier, invalid session identifier, invalid network address, invalid session pool state, invalid message size, pool/node identifier mismatch, and corrupted message.	

The following figure shows an example for OM group OADATCOM.

Note: Messages that correspond to OSSAIN call processing and maintenance use the 0 UDP tuple. Messages that correspond to OSSAIN QMS MIS use the 1 TCP tuple.

Figure 280 MAP display example for OM group OADATCOM

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:         7; FASTSAMPLES:       62;

      OMSGSND    OMSGSND2     OMSGRCV    OMSGRCV2
    OMSGSNCSC    ONSGSNS2    OMSGRCSC    OMSGRCS2
     OMSGSNFL    OMSGRCFL    OSNDRTFL    ORCVRTFL

0 UDP
        36033           3       36001           3
        36031           3       35099           3
            2           0           2           0

1 TCP
         2517           0           0           0
         2485           0           0           0
           32           0           0           0
```

Associated OM groups

OADATCOM is associated with the following OM groups:

- OANODEDC
- OASNPLDC

OAFLTRIG

OM group OAFLTRIG (OSSAIN Float Triggers) provides peg counts for actions related to OSSAIN float trigger processing that cause call control to transfer to a function or a control list.

The following table describes each register.

Table 157 OM group OAFLTRIG

Register	Description
OAFLTFUN	OSSAIN float to function. This register is pegged each time a call in the floated state triggers, causing call control to pass successfully to a function.
OAFLTCTL	OSSAIN float to control list. This register is pegged each time a call in the floated state triggers, causing call control to pass successfully to a control list.
OATRIGFL	OSSAIN trigger failure. This register is pegged each time a call in the floated state triggers, but fails to pass control to a TOPS automated system. Passing call control to a TOPS automated system by way of trigger processing is not allowed in OSSAIN. Failure can occur under one of the following conditions: - passing control to an OSSAIN function that is defined as a TOPS automated system - passing control to an OSSAIN control list whose associated function is defined as a TOPS automated system

The following figure shows an example for OM group OAFLTRIG.

Figure 281 MAP display example for OM group OAFLTRIG

```
CLASS: ACTIVE
START:1999/06/07 13:30:00 WED; STOP: 1999/06/07 13:33:14 WED;
SLOWSAMPLES:          9; FASTSAMPLES:          97;

      OAFLTFUN    OAFLTCTL    OATRIGFL

           319          21           0
```

Associated OM groups

OAFLTRIG is associated with the OAINQMS OM group.

OAINNODE

OM group OAINNODE (OSSAIN Node) provides peg counts for state changes to all nodes, including OSN, OSNM, and OSAC nodes. Each OM tuple corresponds to a node defined in table OANODINV.

The following table describes each register.

Table 158 OM group OAINNODE

Register	Description
NDINSV	Node in service. This register is pegged when a node is brought into service.

Table 158 OM group OAINNODE

Register	Description
NDISTB	Node in service trouble. This register is pegged when a node goes in service trouble because a session pool supported by the node went out of service.
NDSYSB	Node system busy. This register is pegged when a node goes system busy because of a request from a remote node, or from a failure of an RTS or audit.
NDMANB	Node manual busy. This register is pegged when a node is manually busied from a MAP.
NRTSFAIL	Node return to service fail. This register is pegged when a node goes system busy from an RTS failure.
NTSTFAIL	Node test fail. This register is pegged when a node goes system busy from a manual test failure.
NAUDFAIL	Node audit fail. This register is pegged when a node goes system busy from an audit failure.

The following figure shows an example for OM group OAINNODE.

Figure 282 MAP display example for OM group OAINNODE

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:         9; FASTSAMPLES:       97;

       NDINSV      NDISTB      NDSYSB      NDMANB
    NDRTSFAIL    NTSTFAIL    NAUDFAIL

3 NODE_3
            1           1           0           1
            0           0           0
```

Associated OM groups

None

OAINQMS

OM group OAINQMS (OSSAIN QMS) provides peg counts for OSSAIN calls on an individual queue basis. It counts calls that request an OSSAIN session from the QMS CAM and actions taken by the CAM in response to the request.

The following table describes each register.

Table 159 OM group OAINQMS

Register	Description
SESRQSTD	Session requested. This register is pegged when a session is requested by an OSSAIN call from the QMS CAM.

Table 159 OM group OAINQMS

Register	Description
SESRQST2	Session requested extension register
QUEUEDC	Queued call. This register is pegged when an OSSAIN call is queued for a session by the QMS CAM.
QUEUEDC2	Queued call extension register
GOTSESIM	Got session immediately. This register is pegged when a session is obtained immediately from the QMS CAM on request.
GOTSESI2	Got session immediately extension register
DEFLCTCQ	Deflected call queue. This register is pegged when a call destined for a call queue is deflected by the QMS CAM. The deflection is because no agent is available to serve the call and the projected wait time for the call exceeds CDTIME datafilled for the queue in table QMSCQDEF.
OVFLMXCQ	Overflowed call queue (MAXSIZE exceeded). This register is pegged when a call destined for a call queue is overflowed by the QMS CAM. The overflow is because no agent is available to serve the call and the number of calls in the call queue exceeds MAXSIZE datafilled for the queue in table QMSCQDEF.
OVFLMXAP	Overflowed call queue (no call queue elements). This register is pegged when a call destined for a call queue is overflowed by the QMS CAM. The overflow is because no agent is available to serve the call and the call queuing elements for the application have been exhausted (as specified by datafill in table QAPLNDEF).
DENIEDCQ	Denied call queuing. This register is pegged when a call destined for a call queue is deflected by the QMS CAM. The deflection is because no agent is available to serve the call and the QMS CAM is unable to queue the call for a reason not covered in the other registers.
ABANDONC	Abandoned call in queue. This register is pegged when an OSSAIN call is abandoned by a subscriber while the call is in queue for a session.

The following figure shows an example for OM group OAINQMS.

Figure 283 MAP display example for OM group OAINQMS

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:         9; FASTSAMPLES:       97;

INFO (OQMS_QUEUEINDEX_REGISTERINFO)

     SESRQSTD    SESRQST2     QUEUEDC    QUEUEDC2
     GOTSESIM    GOTSESI2    DEFLCTCQ    OVFLMXCQ
     OVFLMXAP    DENIEDCQ    ABANDONC

CQ0
           59           0          21           0
           38           0           0           0
            0           0           1
```

Associated OM groups

OAINQMS is associated with the OAFLTRIG OM group.

OAINRTE

OM group OAINRTE (OSSAIN Route) provides peg counts for obtaining sessions from session pools used for host-remote sessions or trigger event informs. Each OM tuple corresponds to a session pool in table OASESNPL that is defined in one of the following two ways:

- an OSAC session pool (ORIGTYPE subfield set to OSACORIG or OSACTERM)
- a trigger event session pool (TRIGEVNT subfield set to Y)

The following table describes each register.

Table 160 OM group OAINRTE

Register	Description
OSCSESQ	OSAC session request. This register is pegged each time a host-remote session is requested.
OSCSESQ2	OSAC session request extension register
OSCGOTS	OSAC got session. This register is pegged each time a host-remote session is obtained.
OSCGOTS2	OSAC got session extension register
OSCOVFL	OSAC session overflow. This register is pegged each time a host-remote session is requested but is unavailable.
TRGSESQ	Trigger event inform session request. This register is pegged each time a session for a trigger event inform is requested.
TRGSESQ2	Trigger event inform session request extension register
TRGGOTS	Trigger event inform got session. This register is pegged each time a session for a trigger event inform is obtained.

Table 160 OM group OAINRTE

Register	Description
TRGGOTS2	Trigger event inform got session extension register
TRGOVFL	Trigger event overflow. This register is pegged each time a session for a trigger event inform is requested but is unavailable.

The following figure shows an example for OM group OAINRTE.

Figure 284 MAP display example for OM group OAINRTE

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:          9; FASTSAMPLES:       97;

      OSCSESQ    OSCSESQ2     OSCGOTS    OSCGOTS2
      OSCOVFL     TRGSESQ    TRGSESQ2     TRGGOTS
     TRGGOTS2     TRGOVFL

3 SESNPL_3
          120           0         120           0
           16          84           0          84
            0           0
```

Associated OM groups

None

OANODEDC

OM group OANODEDC (OSSAIN Node Data Communications) provides peg counts for OSSAIN data communications messaging events on an individual node basis. It counts the total number of messages sent from the CM to each external node and the total number of messages received by the CM from each external node. Messages from the CM are divided into success counts and failure counts.

The following table describes each register.

Table 161 OM group OANODEDC

Register	Description
ONMSGSND	OSSAIN node message send requested. This register is pegged for a specific node each time the data communications software is requested to send a message. It includes requests from both call and maintenance processes.
ONMSGSN2	OSSAIN node message send requested extension register
ONMSGRCV	OSSAIN node message received. This register is pegged for a specific node each time an incoming message from that node is received. It includes messages from both call and maintenance processes.

Table 161 OM group OANODEDC

Register	Description
ONMSGRC2	OSSAIN node message received extension register
ONMSGSSC	OSSAIN node message send success. This register is pegged for a specific node when the CM's data communications software successfully processes an outgoing message destined for that node. ***Note:*** OSSAIN uses unguaranteed messaging. Pegging this register does not indicate that the message actually arrived at the destination node.
ONMSGSS2	OSSAIN node message send success extension register
ONMSGRSC	OSSAIN node message received success. This register is pegged for a specific node when the CM's data communications software successfully processes an incoming message.
ONMSGRS2	OSSAIN node message received success extension register
ONMSGSFL	OSSAIN node message send failure. This register is pegged for a specific node each time data communications encounters an error when sending an outgoing message to that node. The error can be caused by a transport layer failure or a corrupted message.
ONMSGRFL	OSSAIN node message received failure. This register is pegged for a specific node each time data communications encounters an error when forwarding a message originating from that node to the destination DMS process. The error can be caused by a failure in the DMS internal messaging system or data transport interface. (For a list of other possible reasons for failure, refer to Note.)
ONSNRTFL	OSSAIN node message send route failure. This register is pegged for a specific node each time the data communications software is unable to determine the destination of an outgoing message. The reason for failure is a corrupted message.
ONRCRTFL	OSSAIN node message received route failure. This register is pegged for a specific node each time the data communications software is unable to determine the destination of a message originating from that node. (See Note.)
Note: Possible reasons for failure include invalid protocol version, invalid network address, invalid node state, invalid message size, and corrupted message.	

The following figure shows an example for OM group OANODEDC.

Figure 285 MAP display example for OM group OANODEDC

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:         7; FASTSAMPLES:       62;

     ONMSGSND    ONMSGSN2    ONMSGRCV    ONMSGRC2
     ONMSGSSC    ONMSGSS2    ONMSGRSC    ONMSGRS2
     ONMSGSFL    ONMSGRFL    ONSNRTFL    ONRCRTFL

0 NODE_0
        17620           0       17620           0
        17620           0       17620           0
            0           0           0           0
```

Associated OM groups

OANODEDC is associated with the following OM groups:

- OADATCOM
- OASNPLDC

OAPCALP1

OM group OAPCALP1 (OAP Call Processing) provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note 1: Eight other OM groups (OAPCALP2, OAPCALP3, OAPCALP4, OAPCALP5, OAPCALP6, OAPCALP7, OAPCALP8, and OAPCALP9) are continuations of OAPCALP1; their descriptions follow.

Note 2: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 162 OM group OAPCALP1

Register	Description
APNDAMA	Append AMA module request
APNDAMA2	Append AMA module request extension register
APDAMAS	Append AMA module success response
APDAMAS2	Append AMA module success response extension register
APDAMAE	Append AMA error response
APDAMAE2	Append AMA error response extension register
GENAMA	Generate AMA record

Table 162 OM group OAPCALP1

Register	Description
GENAMA2	Generate AMA record extension register
GENAMAS	Generate AMA record success response
GENAMAS2	Generate AMA record success response extension register
GENAMAE	Generate AMA record error response
GENAMAE2	Generate AMA record error response extension register
BLNGNUM	Billing number request
BLNGNUM2	Billing number request extension register
BILNUMS	Billing number success response
BILNUMS2	Billing number success response extension register
BILNUME	Billing number error response
BILNUME2	Billing number error response extension register
CLSCHRG	Class charge request
CLSCHRG2	Class charge request extension register
CLSCHGS	Class charge success response
CLSCHGS2	Class charge success response extension register
CLSCHGE	Class charge error response
CLSCHGE2	Class charge error response extension register
CALLDET	Call details request
CALLDET2	Call details request extension register
CALDETS	Call details success response
CALDETS2	Call details success response extension register
CALDETE	Call details error response
CALDETE2	Call details error response extension register
CALLEND	Call end inform
CALLEND2	Call end inform extension register

The following figure shows an example for OM group OAPCALP1.

Figure 286 MAP display example for OM group OAPCALP1

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:        18;

      APNDAMA    APNDAMA2    APDAMAS    APDAMAS2
      APDAMAE    APDAMAE2     GENAMA     GENAMA2
      GENAMAS    GENAMAS2    GENAMAE    GENAMAE2
      BLNGNUM    BLNGNUM2    BILNUMS    BILNUMS2
      BILNUME    BILNUME2    CLSCHRG    CLSCHRG2
      CLSCHGS    CLSCHGS2    CLSCHGE    CLSCHGE2
      CALLDET    CALLDET2    CALDETS    CALDETS2
      CALDETE    CALDETE2    CALLEND    CALLEND2

0 SESNPL_0
           12          0          11           0
            1          0          20           0
           20          0           0           0
           23          0           0           0
            0          0          13           0
           13          0           0           0
           33          0          30           0
            3          0         102           0
```

Associated OM groups

OAPCALP1 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP2

OM group OAPCALP2 is a continuation of OAPCALP1 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 163 OM group OAPCALP2

Register	Description
CALLFLT	Call float request
CALLFLT2	Call float request extension register

Table 163 OM group OAPCALP2

Register	Description
CALFLTS	Call float success response
CALFLTS2	Call float success response extension register
CALFLTE	Call float error response
CALFLTE2	Call float error response extension register
DIRNUM	Directory number request
DIRNUM2	Directory number request extension register
DIRNUMS	Directory number success response
DIRNUMS2	Directory number success response extension register
DIRNUME	Directory number error response
DIRNUME2	Directory number error response extension register
ENDCALL	End call request
ENDCALL2	End call request extension register
ENDCALS	End call success response
ENDCALS2	End call success response extension register
ENDCALE	End call error response
ENDCALE2	End call error response extension register
CONDN	Connect DN request
CONDN2	Connect DN request extension register
CONDNS	Connect DN success response
CONDNS2	Connect DN success response extension register
CONDNE	Connect DN error response
CONDNE2	Connect DN error response extension register
RELSDN	Release DN request
RELSDN2	Release DN request extension register
RELSDNS	Release DN success response
RELSDNS2	Release DN success response extension register
RELSDNE	Release DN error response
RELSDNE2	Release DN error response extension register
CONSTAT	Connection status inform
CONSTAT2	Connection status inform extension register

The following figure shows an example for OM group OAPCALP2.

Figure 287 MAP display example for OM group OAPCALP2

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      CALLFLT    CALLFLT2     CALFLTS    CALFLTS2
     CALLFLTE    CALFLTE2      DIRNUM     DIRNUM2
      DIRNUMS    DIRNUMS2     DIRNUME    DIRNUME2
      ENDCALL    ENDCALL2     ENDCALS    ENDCALS2
      ENDCALE    ENDCALE2       CONDN      CONDN2
       CONDNS     CONDNS2      CONDNE     CONDNE2
       RELSDN     RELSDN2     RELSDNS    RELSDNS2
      RELSDNE    RELSDNE2     CONSTAT    CONSTAT2

0 SESNPL_0
           42           0          40           0
            2           0          12           0
           12           0           0           0
           10           0          10           0
            0           0           1           0
           10           0           0           0
            3           0           3           0
            0           0           3           0
```

Associated OM groups

OAPCALP2 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP3

OM group OAPCALP3 is a continuation of OAPCALP2 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 164 OM group OAPCALP3

Register	Description
SESNBEG	Session begin inform
SESNBEG2	Session begin inform extension register
TRIGEVT	Trigger event inform

Table 164 OM group OAPCALP3

Register	Description
TRIGEVT2	Trigger event inform extension register
SESNINI	Session initiation request
SESNINI2	Session initiation request extension register
SESNINS	Session initiation success response
SESNINS2	Session initiation success response extension register
SESNINE	Session initiation error response
SESNINE2	Session initiation error response extension register
SPCHPTH	Speech path request
SPCHPTH2	Speech path request extension register
SPCHPTS	Speech path success response
SPCHPTS2	Speech path success response extension register
SPCHPTE	Speech path error response
SPCHPTE2	Speech path error response extension register
TXTOPR	Text to operator request
TXTOPR2	Text to operator request extension register
TXTOPRS	Text to operator success response
TXTOPRS2	Text to operator success response extension register
TXTOPRE	Text to operator error response
TXTOPRE2	Text to operator error response extension register
XFRCTRL	Transfer to control list request
XFRCTRL2	Transfer to control list request extension register
XFRCTRS	Transfer to control list success response
XFRCTRS2	Transfer to control list success response extension register
XFRCTRE	Transfer to control list error response
XFRCTRE2	Transfer to control list error response extension register

The following figure shows an example for OM group OAPCALP3.

Figure 288 MAP display example for OM group OAPCALP3

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      SESNBEG    SESNBEG2     TRIGEVT    TRIGEVT2
      SESNINI    SESNINI2     SESNINS    SESNINS2
      SESNINE    SESNINE2     SPCHPTH    SPCHPTH2
      SPCHPTS    SPCHPTS2     SPCHPTE    SPCHPTE2
       TXTOPR     TXTOPR2     TXTOPRS    TXTOPRS2
      TXTOPRE    TXTOPRE2     XFRCTRL    XFRCTRL2
      XFRCTRS    XFRCTRS2     XFRCTRE    XFRCTRE2

0 SESNPL_0
           53           0          10           0
           43           0          40           0
            0           0          17           0
           17           0           0           0
            7           0           7           0
            0           0          15           0
           15           0           0           0
```

Associated OM groups

OAPCALP3 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP4

OM group OAPCALP4 is a continuation of OAPCALP3 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 165 OM group OAPCALP4

Register	Description
VCECON	Voice connect request
VCECON2	Voice connect request extension register
VCECONS	Voice connect success response
VCECONS2	Voice connect success response extension register
VCECONE	Voice connect error response

Table 165 OM group OAPCALP4

Register	Description
VCECONE2	Voice connect error response extension register
VCERLS	Voice release request
VCDRLS2	Voice release request extension register
VCERLSS	Voice release success response
VCERLSS2	Voice release success response extension register
VCERLSE	Voice release error response
VCERLSE2	Voice release error response extension register
RELRCVR	Release receiver request
RELRCVR2	Release receiver request extension register
RELRCVS	Release receiver success response
RELRCVS2	Release receiver success response extension register
RELRCVE	Release receiver error response
RELRCVE2	Release receiver error response extension register
DTMFDIG	DTMF digit detected inform
DTMFDIG2	DTMF digit detected inform extension register
CARASGN	Carrier assignment request
CARASGN2	Carrier assignment request extension register
CARASNS	Carrier assignment success response
CARASNS2	Carrier assignment success response extension register
CARASNE	Carrier assignment error response
CARASNE2	Carrier assignment success response extension register
RTETRMT	Route to treatment request
RTETRMT2	Route to treatment request extension register
RTETMTS	Route to treatment success response
RTETMTS2	Route to treatment success response extension register
RTETMTE	Route to treatment error response
RTETMTE2	Route to treatment error response extension register

The following figure shows an example for OM group OAPCALP4.

Figure 289 MAP display example for OM group OAPCALP4

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       VCECON     VCECON2     VCECONS    VCECONS2
      VCECONE    VCECONE2      VCERLS     VCERLS2
      VCERLSS    VCERLSS2     VCERLSE    VCERLSE2
      RELRCVR    RELRCVR2     RELRCVS    RELRCVS2
      RELRCVE    RELRCVE2     DTMFDIG    DTMFDIG2
      CARASGN    CARASGN2     CARASNS    CARASNS2
      CARASNE    CARASNE2

0 SESNPL_0
          107           0         105           0
            2           0         105           0
          105           0           0           0
           14           0          14           0
            0           0          12           0
            5           0           5           0
            0           0
```

Associated OM groups

OAPCALP4 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP5

OM group OAPCALP5 is a continuation of OACALP4 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 166 OM group OAPCALP5

Register	Description
XFRCAR	Transfer to carrier request
XFRCAR2	Transfer to carrier request extension register
XFRCARS	Transfer to carrier success response
XFRCARS2	Transfer to carrier success response extension register

Table 166 OM group OAPCALP5

Register	Description
XFRCARE	Transfer to carrier error response
XFRCARE2	Transfer to carrier error response extension register
RESUME	Resume request
RESUME2	Resume request extension register
RESUMES	Resume success response
RESUMES2	Resume success response extension register
RESUMEE	Resume error response
RESUMEE2	Resume error response extension register
CONVTM	Conversation timing request
CONVTM2	Conversation timing request extension register
CONVTMS	Conversation timing success response
CONVTMS2	Conversation timing success response extension register
CONVTME	Conversation timing error response
CONVTME2	Conversation timing error response extension register
LNPREQ	Local number portability (LNP) request
LNPREQ2	LNP request extension register
LNPREQS	LNP request success response
LNPREQS2	LNP request success response extension register
LNPREQE	LNP request error response
LNPREQE2	LNP request error response extension register
SPDREQ	Service provider identification (SPID) assignment request
SPDREQ2	SPID assignment request extension register
SPDREQS	SPID assignment success response
SPDREQS2	SPID assignment success response extension register
SPDREQE	SPID assignment error response
SPDREQE2	SPID assignment error response extension register

The following figure shows an example for OM group OAPCALP5.

Figure 290 MAP display example for OM group OAPCALP5

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

        XFRCAR     XFRCAR2     XFRCARS    XFRCARS2
       XFRCARE    XFRCARE2      RESUME     RESUME2
       RESUMES    RESUMES2     RESUMEE    RESUMEE2
        CONVTM     CONVTM2     CONVTMS    CONVTMS2
       CONVTME    CONVTME2      LNPREQ     LNPREQ2
       LNPREQS    LNPREQS2     LNPREQE    LNPREQE2
        SPDREQ     SPDREQ2     SPDREQS    SPDREQS2
       SPDREQE    SPDREQE2

0 SESNPL_0
            12           0          11           1
             1           0           1           0
             0           0           1           0
             0           0           0           0
             0           0           0           0
             0           0           0           0
             5           0           5           0
             0           0
```

Associated OM groups

OAPCALP5 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP6

OM group OAPCALP6 is a continuation of OAPCALP5 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 167 OM group OAPCALP6

Register	Description
PASSTAT	Passive node status inform
PASSTAT2	Passive node status inform extension register
RELNOD	Release node request

Table 167 OM group OAPCALP6

Register	Description
RELNOD2	Release node request extension register
RELNODS	Release node success response
RELNODS2	Release node success response extension register
RELNODE	Release node error response
RELNODE2	Release node error response extension register
NODEREL	Node release inform
NODEREL2	Node release inform extension register
PASCNTL	Pass control request
PASCNTL2	Pass control request extension register
PASCNTS	Pass control success response
PASCNTS2	Pass control success response extension register
PASCNTE	Pass control error response
PASCNTE2	Pass control error response extension register
ACPCNTL	Accept control inform
ACPCNTL2	Accept control inform extension register
PASTHRU	Pass through inform
PASTHRU2	Pass through inform extension register
PASSREQ	Passive function provider request
PASSREQ2	Passive function provider request extension register
PASREQS	Passive function provider success response
PASREQS2	Passive function provider success response extension register
PASREQE	Passive function provider error response
PASREQE2	Passive function provider error response extension register
SESRECL	Session recall request
SESRECL2	Session recall request extension register
SESRECLS	Session recall success response
SESRECS2	Session recall success response extension register
SESRECLE	Session recall error response
SESRECE2	Session recall error response extension register

The following figure shows an example for OM group OAPCALP6.

Figure 291 MAP display example for OM group OAPCALP6

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      PASSTAT    PASSTAT2      RELNOD     RELNOD2
      RELNODS    RELNODS2     RELNODE    RELNODE2
      NODEREL    NODEREL2     PASCNTL    PASCNTL2
      PASCNTS    PASCNTS2     PASCNTE    PASCNTE2
      ACPCNTL    ACPCNTL2     PASTHRU    PASTHRU2
      PASSREQ    PASSREQ2     PASREQS    PASREQS2
      PASREQE    PASREQE2     SESRECL    SESRECL2
     SESRECLS    SESRECS2    SESRECLE    SESRECE2

0 SESNPL_0
           12           0          11           0
            1           0          10           0
            3           0           5           0
            9           0           8           0
            7           0           5           0
            4           0          10           0
           11           0          10           5
            5          20          10          10
```

Associated OM groups

OAPCALP6 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP7

OM group OAPCALP7 is a continuation of OAPCALP6 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 168 OM group OAPCALP7

Register	Description
CNFCRET	Conference create request
CNFCRET2	Conference create request extension register
CNFCRETS	Conference create success response
CNFCRES2	Conference create success response extension register

Table 168 OM group OAPCALP7

Register	Description
CNFCRETE	Conference create error response
CNFCREE2	Conference create error response extension register
CNFREMV	Conference remove request
CNFREMV2	Conference remove request extension register
CNFREMVS	Conference remove success response
CNFREMS2	Conference remove success response extension register
CNFREMVE	Conference remove error response
CNFREME2	Conference remove error response extension register
CNFADD	Conference add request
CNFADD2	Conference add request extension register
CNFADDS	Conference add success response
CNFADDS2	Conference add success response extension register
CNFADDE	Conference add error response
CNFADDE2	Conference add error response extension register
CNFREL	Conference release request
CNFREL2	Conference release request extension register
CNFRELS	Conference release success response
CNFRELS2	Conference release success response extension register
CNFRELE	Conference release error response
CNFRELE2	Conference release error response extension register
CNFDETL	Conference details request
CNFDETL2	Conference details request extension register
CNFDETLS	Conference details success response
CNFDETS2	Conference details success response extension register
CNFDETLE	Conference details error response
CNFDETE2	Conference details error response extension register

The following figure shows an example for OM group OAPCALP7.

Figure 292 MAP display example for OM group OAPCALP7

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      CNFCRET    CNFCRET2     CNFCRES    CNFCRES2
     CNFCRETE    CNFCREE2     CNFREMV    CNFREMV2
     CNFREMVS    CNFREMS2    CNFREMVE    CNFREME2
       CNFADD     CNFADD2     CNFADDS    CNFADDS2
      CNFADDE    CNFADDE2      CNFREL     CNFREL2
      CNFRELS    CNFRELS2     CNFRELE    CNFRELE2
      CNFDETL    CNFDETL2    CNFDETLS    CNFDETS2
     CNFDETLE    CNFDETE2

0 SESNPL_0
           12          0          11           0
            1          0          20           0
           20          0           0           0
           23          0           0           0
            0          0          13           0
           13          0           0           0
           33          0          30           0
            3          0
```

Associated OM groups

OAPCALP7 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP8

OM group OAPCALP8 is a continuation of OAPCALP7 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 169 OM group OAPCALP8

Register	Description
TMRREQ	Notification timer request
TMRREQ2	Notification timer request extension register
TMRREQS	Notification timer success response

Table 169 OM group OAPCALP8

Register	Description
TRMREQS2	Notification timer success response extension register
TMRREQE	Notification timer error response
TMRREQE2	Notification timer error response extension register
TMREXPR	Notification timer expiration inform
TMREXPR2	Notification timer expiration inform extension register
MRGINF	Call merge inform
MRGINF2	Call merge inform extension register
CALLMRG	Call merge request
CALLMRG2	Call merge request extension register
CALLMRGS	Call merge success response
CALMRGS2	Call merge success response extension register
CALLMRGE	Call merge error response
CALMRGE2	Call merge error response extension register
CGPNRQ	ISUP calling party number update request
CGPNRQ2	ISUP calling party number update request extension register
CGPNRQS	ISUP calling party number update success response
CGPNRQS2	ISUP calling party number update success response extension register
CGPNRQE	ISUP calling party number update error response
CGPNRQE2	ISUP calling party number update error response extension register
SVCCHG	Service change request
SVCCHG2	Service change request extension register
SVCCHGS	Service change success response
SVCCHGS2	Service change success response extension register
SVCCHGE	Service change error response
SVCCHGE2	Service change error response extension register
VCERLSI	Voice release inform
VCERLSI2	Voice release inform extension register

The following figure shows an example for OM group OAPCALP8.

Figure 293 MAP display example for OM group OAPCALP8

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:        18;

       TMRREQ     TMRREQ2     TRMREQS    TMRREQS2
      TRMREQE    TMRREQE2     TMREXPR    TMREXPR2
       MRGINF     MRGINF2     CALLMRG    CALLMRG2
     CALLMRGS    CALMRGS2    CALLMRGE   CALMERGE2
       CGPNRQ     CGPNRQ2     CGPNRQS    CGPNRQS2
      CGPNRQE    CGPNRQE2      SVCCHG     SVCCHG2
      SVCCHGS    SVCCHGS2     SVCCHGE    SVCCHGE2
      VCERLSI    VCERLSI2

0 SESNPL_0
           53           0          10           0
            3           0          20           0
            0           0          17           0
            1           0           5           0
            2           0           1           0
            3           0           2           0
            2           0           1           0
            3           0
```

Associated OM groups

OAPCALP8 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCALP9

OM group OAPCALP9 is a continuation of OAPCALP8 and provides peg counts to track OAP operations and responses. There is a register for each call processing operation and response message defined in the OAP protocol. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 170 OM group OAPCALP9

Register	Description
ESTCHG	Estimate of charges request
ESTCHG2	Estimate of charges request extension register

Table 170 OM group OAPCALP9

Register	Description
ESTCHGS	Estimate of charges success response
ESTCHGS2	Estimate of charges success response extension register
ESTCHGE	Estimate of charges error response
ESTCHGE2	Estimate of charges error response extension register
PASTHRQ	Pass through request
PASTHRQ2	Pass through request extension register
PASTHRS	Pass through success response
PASTHRS2	Pass through success response extension register
PASTHRE	Pass through error response
PASTHRE2	Pass through error response extension register
CNTTMT	Connect to treatment request
CNTTMT2	Connect to treatment request extension register
CNTTMTS	Connect to treatment success response
CNTTMTS2	Connect to treatment success response extension register
CNTTMTE	Connect to treatment error response
CNTTMTE2	Connect to treatment error response extension register
SACTINF	Session active inform
SACTINF2	Session active inform extension register

The following figure shows an example for OM group OAPCALP9.

Figure 294 MAP display example for OM group OAPCALP9

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       ESTCHG     ESTCHG2     ESTCHGS    ESTCHGS2
      ESTCHGE    ESTCHGE2     PASTHRQ    PASTHRQ2
      PASTHRS    PASTHRS2     PASTHRE    PASTHRE2
       CNTTMT     CNTTMT2     CNTTMTS    CNTTMTS2
      CNTTMTE    CNTTMTE2     SACTINF    SACTINF2

0 SESNPL_0
            0           0           0           0
            0           0          10           0
           10           0           0           0
           53           0           0           0
            3           0           0           0
```

Associated OM groups

OAPCALP9 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPCP10

OM group OAPCP10 was created in LET0012 with registers to count the usage, successes, and errors in the new OAP operations introduced in that release. SN07 adds registers for the new SMS operation.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 171 OM group OAPCP10

Register	Description
RETANRQ	Return answer request
RETANRQ2	Return answer request extension register
RETANRS	Return answer request success
RETANRS2	Return answer request success extension register
RETANRE	Return answer request error
RETANRE2	Return answer request error extension register
CBNSQRQ	Calling party BNS query result
CBNSQRQ2	Calling party BNS query result extension register
CBNSQRS	Calling party BNS query success
CBNSQRS2	Calling party BNS query success extension register
CBNSQRE	Calling party BNS query request error
CBNSQRE2	Calling party BNS query request error extension register
RNUPDRQ	LRN update request
RNUPDRQ2	LRN update request extension register
RNUPDRS	LRN update success
RNUPDRS2	LRN update success extension register
RNUPDRE	LRN update request error
RNUPDRE2	LRN update request error extension register
SMSRQ	SMS request

Table 171 OM group OAPCP10

Register	Description
SMSRQ2	SMS request extension register
SMSRS	SMS success
SMSRS2	SMS success extension register
SMSRE	SMS request error
SMSRE2	SMS request error extension register

The following figure shows an example for OM group OAPCP10.

Figure 295 MAP display example for OM group OAPCP10

```
CLASS: ACTIVE
START:1999/07/14 16:30:00 WED; STOP: 1999/07/14 16:33:00 WED;
SLOWSAMPLES:      18; FASTSAMPLES:        172;

RETANRQ          RETANRQ2     RETANRS    RETANRS2
RETANRE          RETANRE2     CBNSQRQ    CBNSQRQ2
CBNSQRS          CBNSQRS2     CBNSQRE    CBNSQRE2
RNUPDRQ          RNUPDRQ2     RNUPDRS    RNUPDRS2
RNUPDRE          RNUPDRE2       SMSRQ      SMSRQ2
SMSRS              SMSRS2       SMSRE      SMSRE2


0 SESNPL_0
           42           0          41           0
            1           0           2           0
            1           0           0           0
            8           0           8           0
            0           0           6           0

            5           0           1           0
```

Associated OM groups

OAPCP10 is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OAPMERRN

OM group OAPMERRN (OAP Message Error Node) provides peg counts for each type of error that an OAP message can have. Each OM tuple corresponds to a node defined in table OANODINV.

The following table describes each register.

Table 172 OM group OAPMERRN

Register	Description
NINVOPHD	Node invalid operation header ID. This register is pegged each time the switch receives a message with an invalid operation header ID.
NINVOPHD2	Node invalid operation header ID extension register
NINVKER	Node invoke error. This register is pegged each time the switch receives an invalid value in the Invoke Operation Header.
NINVKER2	Node invoke error extension register
NRESLTER	Node return result error. This register is pegged each time the switch receives an invalid value in the Return Result Operation Header.
NRESLTE2	Node return result error extension register
NERRORER	Node return error error. This register is pegged each time the switch receives an invalid value in the Return Error Operation Header.
NERRORE2	Node return error error extension register
NREJCTE	Node reject error. This register is pegged each time the switch receives an invalid value in the Reject Operation Header.
NREJCTE2	Node reject error extension register
NUNKNOP	Node unknown operation ID. This register is pegged each time the switch receives a message with an unknown operation ID.
NUNKNOP2	Node unknown operation ID extension register
NUNKNDB	Node unknown data block. This register is pegged each time the switch receives an operation or response with an unknown data block from an SN.
NUNKNDB2	Node Unknown Data Block extension register

Table 172 OM group OAPMERRN

Register	Description
NMSNGDB	Node missing data block. This register is pegged each time the switch receives an operation or response with a missing mandatory data block.
NMSNGDB2	Node missing data block extension register
NINVDFD	Node invalid field value. This register is pegged each time the switch receives a message that has a field with an invalid value.
NINVDFD2	Node invalid field value extension register

The following figure shows an example for OM group OAPMERRN.

Figure 296 MAP display example for OM group OAPMERRN

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

     NINVOPHD    NINVOPH2     NINVKER    NINVKER2
     NRESLTER    NRESLTE2    NERRORER    NERRORE2
      NREJCTE    NREJCTE2     NUNKNOP    NUNKNOP2
      NUNKNDB    NUNKNDB2     NMSNGDB    NMSNGDB2
      NINVDFD    NINVDFD2

0 NODE_0
            0           0           0           0
            0           0           0           0
            5           0           6           0
            3           0           1           0
            6           0
```

Associated OM groups

None

OAPMERRS

OM group OAPMERRS (OAP Message Error Session Pool) provides peg counts for each type of error that an OAP message can have. Each OM tuple corresponds to a node defined in table OANODINV. The following table describes each register.

Table 173 OM group OAPMERRS

Register	Description
INVLFN	Invalid function ID. This register is pegged each time the switch receives a message with an invalid function ID.
INVLFN2	Invalid function ID extension register

Table 173 OM group OAPMERRS

Register	Description
INVLCLL	Invalid call ID. This register is pegged each time the switch receives a message with an invalid call ID.
INVLCLL2	Invalid call ID extension register
SEQERRH	Out of sequence - high. This register is pegged each time the switch receives a message that is out of sequence and the sequence number is higher than what the switch is expecting.
SEQERRH2	Out of sequence - high extension register
SEQERRL	Out of sequence - low. This register is pegged each time the switch receives a message that is out of sequence and the sequence number is lower than what the switch is expecting.
SEQERRL2	Out of sequence - low extension register
INVLOPHD	Invalid operation header ID. This register is pegged each time the switch receives a message with an invalid operation header ID.
INVLOPHD2	Invalid operation header ID extension register
INVKERR	Invoke error. This register is pegged each time the switch receives an invalid value in the Invoke Operation Header.
INVKERR2	Invoke error extension register
RRESLTER	Return result error. This register is pegged each time the switch receives an invalid value in the Return Result Operation Header.
RRESLTE2	Return result error extension register
RERRORER	Return error error. This register is pegged each time the switch receives an invalid value in the Return Error Operation Header.
RERRORE2	Return error error extension register
REJECTER	Reject error. This register is pegged each time the switch receives an invalid value in the Reject Operation Header.
REJECTER2	Reject error extension register
UNKWNOP	Unknown operation ID. This register is pegged each time the switch receives a message with an unknown operation ID.
UNKWNOP2	Unknown operation ID extension register
UNKWNDB	Unknown data block. This register is pegged each time the switch receives an operation or response with an unknown data block from an SN.
UNKWNDB2	Unknown data block extension register

Table 173 OM group OAPMERRS

Register	Description
MISNGDB	Missing data block. This register is pegged each time the switch receives an operation or response with a missing mandatory data block.
MISNGDB2	Missing data block extension register
INVDFLD	Invalid field value. This register is pegged each time the switch receives a message that has a field with an invalid value.
INVDFLD2	Invalid field value extension register

The following figure shows an example for OM group OAPMERRS.

Figure 297 MAP display example for OM group OAPMERRS

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       INVLFN     INVLFN2     INVLCLL    INVLCLL2
      SEQERRH    SEQERRH2     SEQERRL    SEQERRL2
     INVLOPHD    INVLOPH2     INVKERR    INVKERR2
     RRESLTER    RRESLTE2    RERRORER    RERRORE2
     REJECTER    REJECTE2     UNKWNOP    UNKWNOP2
      UNKWNDB    UNKWNDB2     MISNGDB    MISNGDB2
      INVDFLD    INVDFLD2

0 SESNPL_0
            5           0           3           0
            7           0           0           0
            0           0           0           0
            0           0           5           0
            6           0           3           0
            1           0           6           0
            3           0
```

Associated OM groups

None

OAPMTYPN

OM group OAPMTYPN (OAP Message Type Node) provides peg counts for each incoming and outgoing OAP message type. Each OM tuple corresponds to a node defined in table OANODINV. The following table describes each register.

Table 174 OM group OAPMTYPN

Register	Description
NIINVOK	Node incoming invoke. This register is pegged each time the switch receives a request operation on a node level basis.

Table 174 OM group OAPMTYPN

Register	Description
NIINVOK2	Node incoming invoke extension register
NOINVOK	Node outgoing invoke. This register is pegged each time the switch sends a request operation on a node level basis.
NOINVOK	Node outgoing invoke extension register
NIRESLT	Node incoming result. This register is pegged each time the switch receives a success response on a node level basis.
NIRESLT2	Node incoming result extension register
NORESLT	Node outgoing result. This register is pegged each time the switch sends a success response on a node level basis.
NORESLT2	Node outgoing result extension register
NIERROR	Node incoming error. This register is pegged each time the switch receives an error response on a node level basis.
NIERROR2	Node incoming error extension register
NOERROR	Node outgoing error. This register is pegged each time the switch sends an error response on a node level basis.
NOERROR2	Node outgoing error extension register
NIREJCT	Node incoming reject. This register is pegged each time the switch receives a protocol violation on a node level basis.
NIREJCT2	Node incoming reject extension register
NOREJCT	Node outgoing reject. This register is pegged each time the switch sends a protocol violation on a node level basis.
NOREJCT2	Node outgoing reject extension register

The following figure shows an example for OM group OAPMTYPN.

Figure 298 MAP display example for OM group OAPMTYPN

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      NIINVOK    NIINVOK2     NOINVOK    NOINVOK2
      NIRESLT    NIRESLT2     NORESLT    NORESLT2
      NIERROR    NIERROR2     NOERROR    NOERROR2
      NIREJCT    NIREJCT2     NOREJCT    NOREJCT2

0 NODE_0
          110           0           0           0
            0           0         102           0
            0           0           5           0
            0           0           3           0
```

Associated OM groups

OAPMTYPN is associated with the following OM groups:

- OAPCALP1
- OAPCALP2
- OAPCALP3
- OAPCALP4
- OAPCALP5
- OAPCALP6
- OAPCALP7
- OAPCALP8
- OAPCALP9
- OAPCP10
- OAPMTYPS
- OAPNMTC

OAPMTYPS

OM group OAPMTYPS (OAP Message Type Session Pool) provides peg counts for each incoming and outgoing OAP message type. Each register is pegged on an individual session pool basis. The following table describes each register.

Table 175 OM group OAPMTYPS

Register	Description
ININVOK	Incoming invoke. This register is pegged each time the switch receives an incoming request operation from a session pool.
ININVOK2	Incoming invoke extension register
OGINVOK	Outgoing invoke. This register is pegged each time the switch sends a request operation to a session pool.
OGINVOKS	Outgoing invoke extension register
INRESLT	Incoming result. This register is pegged each time the switch receives a success response from a session pool.
INRESLT2	Incoming result extension register
OGRESLT	Outgoing result. This register is pegged each time the switch sends a success response for an operation request to a session pool.
OGRESLT2	Outgoing result extension register
INERROR	Incoming error. This register is pegged each time the switch receives an error response from a session pool.

Table 175 OM group OAPMTYPS

Register	Description
INERROR2	Incoming error extension register
OGERROR	Outgoing error. This register is pegged each time the switch sends an error response for an operation request to a session pool.
OGERROR2	Outgoing error extension register
INREJCT	Incoming reject. This register is pegged each time the switch receives a protocol violation from a session pool.
INREJCT2	Incoming reject extension register
OGREJCT	Outgoing reject. This register is pegged each time the switch sends a protocol violation to a session pool.
OGREJCT2	Outgoing reject extension register

The following figure shows an example for OM group OAPMTYPS.

Figure 299 MAP display example for OM group OAPMTYPS

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      ININVOK    ININVOK2     OGINVOK    OGINVOK2
      INRESLT    INRESLT2     OGRESLT    OGRESLT2
      INERROR    INERROR2     OGERROR    OGERROR2
      INREJCT    INREJCT2     OGREJCT    OGREJCT2

0 SESNPL_0
          110           0           0           0
            0           0         102           0
            0           0           5           0
            0           0           3           0
```

Associated OM groups

OAPMTYPS is associated with the following OM groups:

- OAPCALP1
- OAPCALP2
- OAPCALP3
- OAPCALP4
- OAPCALP5
- OAPCALP6
- OAPCALP7
- OAPCALP8

- OAPCALP9
- OAPCP10
- OAPMTYPN
- OAPSPMTC

OAPNMIS

OM group OAPNMIS (OAP MIS Node) provides peg counts for each OAP MIS node class message sent to an MIS node. Each OM tuple corresponds to an MIS node defined in table OANODNAM and in table OANODINV with the PM type set to OSNM. The MIS node also must have the OSSAIN QMS MIS application set in table QMSMIS.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register.

Table 176 OM group OAPNMIS

Register	Description
MISOAIN	MIS OSSAIN data. This register is pegged each time the switch sends an OSSAIN QMS MIS data buffer to an MIS node.
MISOAIN2	MIS OSSAIN data extension register

The following figure shows an example for OM group OAPNMIS.

Figure 300 MAP display example for OM group OAPNMIS

```
CLASS: ACTIVE
START:1999/06/07 13:30:00 WED; STOP: 1999/06/07 13:33:14 WED;
SLOWSAMPLES:         9; FASTSAMPLES:         97;

      MISOAIN    MISOAIN2

24 MISNODE
          110           0
```

Associated OM groups

OAPNMIS is associated with the OAPNMTC OM group.

OAPNMTC

OM group OAPNMTC (OAP Node Maintenance) provides peg counts for each incoming and outgoing node maintenance operation and response used. Each OM tuple corresponds to a node defined in table OANODINV that has a PM type set to OSNM or OSN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 177 OM group OAPNMTC

Register	Description
NODEAUD	Node audit request
NODEAUD2	Node audit request extension register
NDEAUDS	Node audit success response
NDEAUDS2	Node audit success response extension register
NDEAUDE	Node audit error response
NDEAUDE2	Node audit error response extension register
NODEBSY	Node busy request
NODEBSY2	Node busy request extension register
NDEBSYS	Node busy success response
NDEBSYS2	Node busy success response extension register
NDEBSYE	Node busy error response
NDEBSYE2	Node busy error response extension register
NODETST	Node test request
NODETST2	Node test request extension register
NDETSTS	Node test success response
NDETSTS2	Node test success response extension register
NDETSTE	Node test error response
NDETSTE2	Node test error response extension register
NODERTS	Node RTS request
NODERTS2	Node RTS request extension register
NDERTSS	Node RTS success response
NDERTSS2	Node RTS success response extension register
NDERTSE	Node RTS error response
NDERTSE2	Node RTS error response extension register
NDLOG	Node log report request
NDALARM	Node alarm request
NODECON	Node Connectivity Test request
NODECON2	Node Connectivity Test request extension register

Table 177 OM group OAPNMTC

Register	Description
NDECONS	Node Connectivity Test success response
NDECONS2	Node Connectivity Test success response extension register
NDECONE	Node Connectivity Test error response
NDECONE2	Node Connectivity Test error response extension register

The following figure shows an example for OM group OAPNMTC.

Figure 301 MAP display example for OM group OAPNMTC

```
CLASS: ACTIVE
START:2004/04/23 18:30:00 FRI; STOP: 2004/04/23 18:33:57 FRI;
SLOWSAMPLES:         3 ; FASTSAMPLES:         24 ;

      NODEAUD    NODEAUD2     NDEAUDS    NDEAUDS2
      NDEAUDE    NDEAUDE2     NODEBSY    NODEBSY2
      NDEBSYS    NDEBSYS2     NDEBSYE    NDEBSYE2
      NODETST    NODETST2     NDETSTS    NDETSTS2
      NDETSTE    NDETSTE2     NODERTS    NODERTS2
      NDERTSS    NDERTSS2     NDERTSE    NDERTSE2
        NDLOG     NDALARM     NODECON    NODECON2
      NDECONS    NDECONS2     NDECONE    NDECONE2

0 NODE_0
           27           0          27           0
            0           0          32           0
           32           0           0           0
           19           0          19           0
            0           0          35           0
           35           0           0           0
            3           1           0           0
            0           0           0           0
```

Associated OM groups

OAPNMTC is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN
- OAPNMIS

OAPSPMTC

OM group OAPSPMTC (OAP Session Pool Maintenance) provides peg counts for each incoming and outgoing session pool maintenance operation and response used. Each OM tuple corresponds to a session pool defined in table OASESNPL that has an ORIGTYPE set to SUBSCRIBER or SN.

Note: For complete information on OAP operations and responses, please refer to *OSSAIN Open Automated Protocol Specification*, Q235-1.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 178 OM group OAPSPMTC

Register	Description
SPAUDIT	Session pool audit request
SPAUDIT2	Session pool audit request extension register
SPAUDTS	Session pool audit success response
SPAUDTS2	Session pool audit success response extension register
SPAUDTE	Session pool audit error response
SPAUDTE2	Session pool audit error response extension register
SPBUSY	Session pool busy request
SPBUSY2	Session pool busy request extension register
SPBUSYS	Session pool busy success response
SPBUSYS2	Session pool busy success response extension register
SPBUSYE	Session pool busy error response
SPBUSYE2	Session pool busy error response extension register
SPTEST	Session pool test request
SPTEST2	Session pool test request extension register
SPTESTS	Session pool test success response
SPTESTS2	Session pool test success response extension register
SPTESTE	Session pool test error response
SPTESTE2	Session pool test error response extension register

Table 178 OM group OAPSPMTC

Register	Description
SPRTS	Session pool RTS request
SPRTS2	Session pool RTS request extension register
SPRTSS	Session pool RTS success response
SPRTSS2	Session pool RTS success response extension register
SPRTSE	Session pool RTS error response
SPRTSE2	Session pool RTS error response extension register
SPCH	Session pool change (throttle) request
SPCHS	Session pool change (throttle) success response
SPCHE	Session pool change (throttle) error response
SPLOG	Session pool log request
SPALARM	Session pool alarm request
SPDRAIN	Session pool drain request
SPSTATE	Session pool state inform
SPSTATE2	Session pool state inform extension register

The following figure shows an example for OM group OAPSPMTC.

Figure 302 MAP display example for OM group OAPSPMTC

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      SPAUDIT    SPAUDIT2     SPAUDTS    SPAUDTS2
      SPAUDTE    SPAUDTE2      SPBUSY     SPBUSY2
      SPBUSYS    SPBUSYS2     SPBUSYE     SBUSYE2
       SPTEST     SPTEST2    SPTESTS    SPTESTS2
      SPTESTE    SPTESTE2       SPRTS      SPRTS2
       SPRTSS     SPRTSS2      SPRTSE     SPRTSE2
         SPCH       SPCHS       SPCHE       SPLOG
      SPALARM     SPDRAIN     SPSTATE    SPSTATE2

0 SESNPL_0
          127           0         217           0
            0           0         132           0
          132           0           0           0
           19           0          19           0
            0           0         135           0
          135           0           0           0
            3           2           1           0
            0           0          10           0
```

Associated OM groups

OAPSPMTC is associated with the following OM groups:

- OAPMTYPS
- OAPMTYPN

OASNPLDC

OM group OASNPLDC (OSSAIN Session Pool Data Communications) provides peg counts for OSSAIN data communications messaging events on an individual session pool basis. It counts the total number of messages sent from the CM to each session pool and the total number of messages received by the CM from each session pool. Messages from the CM are divided into success counts and failure counts.

The following table describes each register.

Table 179 OM group OASNPLDC

Register	Description
OSMSGSND	OSSAIN session pool message send requested. This register is pegged for a specific session pool each time the data communications software is requested to send a message. It includes requests from both call and maintenance processes.
OSMSGSN2	OSSAIN session pool message send requested extension register
OSMSGRCV	OSSAIN session pool message received. This register is pegged for a specific session pool each time an incoming message from that session pool is received. It includes messages from both call and maintenance processes.
OSMSGRC2	OSSAIN session pool message received extension register
OSMSGSSC	OSSAIN session pool message send success. This register is pegged for a specific session pool when the CM's data communications software successfully processes an outgoing message destined for that session pool. ***Note:*** OSSAIN uses unguaranteed messaging. Pegging this register does not indicate that the message actually arrived at the destination session pool.
OSMSGSS2	OSSAIN session pool message send success extension register
OSMSGRSC	OSSAIN session pool message received success. This register is pegged for a specific session pool when the CM's data communications software successfully processes an incoming message.
OSMSGRS2	OSSAIN session pool message received success extension register

Table 179 OM group OASNPLDC

Register	Description
OSMSGSFL	OSSAIN session pool message send failure. This register is pegged for a specific session pool each time data communications encounters an error when sending an outgoing message to that session pool. The error can be caused by a transport layer failure. (For a list of other possible reasons for failure, refer to Note 1.)
OSMSGRFL	OSSAIN session pool message received failure. This register is pegged for a specific session pool each time data communications encounters an error when forwarding a message originating from that session pool to the destination DMS process. The error can be caused by a failure in the DMS internal messaging system or data transport interface. (For a list of other possible reasons for failure, refer to Note 2.)
OSSNRTFL	OSSAIN session pool message send route failure. This register is pegged for a specific session pool each time the data communications software is unable to determine the destination of an outgoing message. (See Note 1.)
OSRCRTFL	OSSAIN session pool message received route failure. This register is pegged for a specific session pool each time the data communications software is unable to determine the destination of a message originating from that session pool. (See Note 2.)
Note 1: Possible reasons for failure include pool/node identifier mismatch and corrupted message. ***Note 2:*** Possible reasons for failure include invalid protocol version, invalid network address, invalid session pool state, pool/node identifier mismatch, and corrupted message.	

The following figure shows an example for OM group OASNPLDC.

Figure 303 MAP display example for OM group OASNPLDC

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:         7; FASTSAMPLES:       62;

     OSMSGSND    OSMSGSN2    OSMSGRCV    OSMSGRC2
     OSMSGSSC    OSMSGSS2    OSMSGRSC    OSMSGRS2
     OSMSGSFL    OSMSGRFL    OSSNRTFL    OSRCRTFL

0 SESNPL_0
        17620           0       17620           0
        17620           0       17620           0
            0           0           0           0
```

Associated OM groups

OASNPLDC is associated with the following OM groups:

- OADATCOM
- OANODEDC

OASNPOOL

OM group OASNPOOL (OSSAIN Session Pool) provides peg counts for OSSAIN session pool maintenance states. Each session pool has its own group of OM registers. The following table describes each register.

Table 180 OM group OASNPOOL

Register	Description
SPSYSB	Session pool system busy. This register is pegged each time the session pool goes system busy.
SPMANB	Session pool manual busy. This register is pegged each time the session pool goes manual busy.
SPCBSY	Session pool c-side busy. This register is pegged each time the session pool goes c-side busy.
RTSFAIL	Return to service failure. This register is pegged each time the session pool fails to return to service.
TSTFAIL	Test failure. This register is pegged each time the session pool fails the test command.
AUDFAIL	Audit failure. This register is pegged each time the session pools the automated audit.
SPINSV	Session pool in service. This register is pegged each time the session pool is brought into service.

The following figure shows an example for OM group OASNPOOL.

Figure 304 MAP display example for OM group OASNPOOL

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:          9; FASTSAMPLES:       89;

     INFO (OASNPOOL_INDEX_OMINFO)
       SPSYSB      SPMANB      SPCBSY     RTSFAIL
      TSTFAIL     AUDFAIL      SPINSV

0 SESNPL_0
            0           1           0           0
            0           0           1
```

Associated OM groups

OASNPOOL is associated with the OAINNODE OM group.

OASVNDCP

OM group OASVNDCP (OSSAIN Service Node Call Processing) provides peg counts for OSSAIN calls on an individual session pool basis. It counts all SN or OSAC call processing activities. Each OM tuple corresponds to a session pool defined in table OASESNPL.

The following table describes each register.

Table 181 OM group OASVNDCP

Register	Description
NDCALERR	Node call error. This register is pegged each time a node requests to end a call because of an unrecoverable error.
NDMSGICL	Node message invalid call. This register is pegged each time a node attempts to send a message for a call that is no longer active.
OSCCLERR	OSAC call error. This register is pegged each time the OSAC host ends a call because of an error at the host.
OSCMICL	OSAC message invalid call. This register is pegged each time the OSAC remote receives a message from the OSAC host for a call that is no longer active.
SBTIMOUT	Session begin timer out

The following figure shows an example for OM group OASVNDCP.

Figure 305 MAP display example for OM group OASVNDCP

```
CLASS: ACTIVE
START:1999/06/07 13:30:00 WED; STOP: 1999/06/07 13:33:14 WED;
SLOWSAMPLES:         2; FASTSAMPLES:         18;

     NDCALERR    NDMSGICL    OSCCLERR     OSCMICL
     SBTIMOUT

3 SESNPL_3
           96           4          93           2
            6
```

Associated OM groups

None

OSACCP1

OM group OSACCP1 (OSAC Call Processing) provides peg counts to track each OSAC call processing class operation and response. Each OM tuple corresponds to an OSAC session pool defined in table OASESNPL. The session pool must have the ORIGYTPE field set to OSACORIG or OSACTERM.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 182 OM group OSACCP1

Register	Description
GOTSESN	Got session inform
GOTSESN2	Got session inform extension register
ENDSESN	End session inform
ENDSESN2	End session inform extension register
RELSNI	Release session inform
RELSNI2	Release session inform extension register
RELSESN	Release session request
RELSESN2	Release session request extension register
RELSENS	Release session success response
RELSENS2	Release session success response extension register
RELSENE	Release session error response
RELSENE2	Release session error response extension register
SESBRQ	Session begin request
SESBRQ2	Session begin request extension register
SESBRQS	Session begin request success response
SESBRQS2	Session begin request success response extension register
SESBRQE	Session begin request error response
SESBRQE2	Session begin request error response extension register
SIVERRQ	Session initiation verification request
SIVERRQ2	Session initiation verification request extension register
SIVERS	Session initiation verification request success response
SIVERS2	Session initiation verification request success response extension register
SIVERE	Session initiation verification request error response
SIVERE2	Session initiation verification request error response extension register

The following figure shows an example for OM group OSACCP1.

Figure 306 MAP display example for OM group OSACCP1

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       GOTSESN    GOTSESN2     ENDSESN    ENDSESN2
        RELSNI     RELSNI2     RELSESN    RELSESN2
       RELSENS    RELSENS2     RELSENE    RELSENE2
        SESBRQ     SESBRQ2     SESBRQS    SESBRQS2
       SESBRQE    SESBRQE2     SIVERRQ    SIVERRQ2
        SIVERS     SIVERS2      SIVERE     SIVERE2

12 SESNPL_12
            10           0           3           0
            16           0          84           0
            84           0           0           0
            80           0          80           0
             0           0          12           0
            12           0           0           0
```

Associated OM groups

OSACCP1 is associated with the following OM groups:

- OSACCP2
- OSACND
- OSACSP
- OSNND
- OSNSP
- OAPMTYPS
- OAPMERRS

OSACCP2

OM group OSACCP2 is a continuation of OSACCP1 and provides peg counts to track each OSAC call processing class operation and response. Each OM tuple corresponds to an OSAC session pool defined in table OASESNPL. The session pool must have the ORIGYTPE field set to OSACORIG or OSACTERM.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 183 OM group OSACCP2

Register	Description
VCCONN	Voice connect request
VCCONN2	Voice connect request extension register
VCCONNS	Voice connect success response
VCCONNS2	Voice connect success response extension register
VCCONNE	Voice connect error response
VCCONNE2	Voice connect error response extension register
VCRELS	Voice release request
VCRELS2	Voice release request extension register
VCRELSS	Voice release success response
VCRELSS2	Voice release success response extension register
VCRELSE	Voice release error response
VCRELSE2	Voice release error response extension register
VCRELSI	Voice release inform
VCRELSI2	Voice release inform extension register
MISUPDT	MIS update inform
MISUPDT2	MIS update inform extension register

The following figure shows an example for OM group OSACCP2.

Note: The registers MISUPDT and MISUPDT2 are used in OSSAIN QMS MIS message processing.

Figure 307 MAP display example for OM group OSACCP2

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

        VCONN      VCONN2      VCONNS     VCONNS2
       VCONNE     VCONNE2      VCRELS     VCRELS2
      VCRELSS    VCRELSS2     VCRELSE    VCRELSE2
      VCRELSI    VCRELSI2     MISUPDT    MISUPDT2

12 SESNPL_12
           96           0          96           0
            6           0          12           0
           12           0           0           0
           10           0           0           0
```

Associated OM groups

OSACCP2 is associated with the following OM groups:

- OSACCP1
- OSACND
- OSACSP
- OSNND
- OSNSP
- OAPMTYPS
- OAPMERRS

OSACND

OM group OSACND (OSAC Node) provides peg counts to track each OSAC node class operation and response. Each OM tuple corresponds to an OSAC node defined in table OANODINV.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 184 OM group OSACND

Register	Description
NDAUD	Node audit request
NDAUD2	Node audit request extension register
NDAUDS	Node audit success response
NDAUDS2	Node audit success response extension register
NDAUDE	Node audit error response
NDAUDE2	Node audit error response extension register
NDBSY	Node busy request
NDBSY2	Node busy request extension register
NDBSYS	Node busy request success response
NDBSYS2	Node busy request success response extension register
NDBSYE	Node busy request error response
NDBSYE2	Node busy request error response extension register
NDRTS	Node RTS request
NDRTS2	Node RTS request extension register
NDRTSS	Node RTS success response
NDRTSS2	Node RTS success response extension register

Table 184 OM group OSACND

Register	Description
NDRTSE	Node RTS error response
NDRTSE2	Node RTS error response extension register
NDTST	Node test request
NDTST2	Node test request extension register
NDTSTS	Node test success response
NDTSTS2	Node test success response extension register
NDTSTE	Node test error response
NDTSTE2	Node test error response extension register

The following figure shows an example for OM group OSACND.

Figure 308 MAP display example for OM group OSACND

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

         NDAUD      NDAUD2      NDAUDS     NDAUDS2
        NDAUDE     NDAUDE2       NDBSY      NDBSY2
        NDBSYS     NDBSYS2      NDBSYE     NDBSYE2
         NDRTS      NDRTS2      NDRTSS     NDRTSS2
        NDRTSE     NDRTSE2       NDTST      NDTST2
        NDTSTS     NDTSTS2      NDTSTE     NDTSTE2

20 NODE_20
            27           0          27           0
             0           0          10           0
            10           0           0           0
            10           0          10           0
             6           0          13           0
            13           0           2           0
```

Associated OM groups

OSACND is associated with the following OM groups:

- OSACCP1
- OSACCP2
- OSACSP
- OSNND
- OSNSP
- OAPMTYPN
- OAPMERRN

OSACSP

OM group OSACSP (OSAC Session Pool) provides peg counts to track each OSAC session pool class operation and response. Each OM tuple corresponds to an OSAC session pool defined in table OASESNPL.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 185 OM group OSACSP

Register	Description
SPLAUD	Session pool audit request
SPLAUD2	Session pool audit request extension register
SPLAUDS	Session pool audit success response
SPLAUDS2	Session pool audit success response extension register
SPLAUDE	Session pool audit error response
SPLAUDE2	Session pool audit error response extension register
SPLBSY	Session pool busy request
SPLBSY2	Session pool busy request extension register
SPLBSYS	Session pool busy success response
SPLBSYS2	Session pool busy success response extension register
SPLBSYE	Session pool busy error response
SPLBSYE2	Session pool busy error response extension register
SPLDRN	Session pool drain request
SPLDRN2	Session pool drain request extension register
SPLRTS	Session pool RTS request
SPLRTS2	Session pool RTS request extension register
SPLRTSS	Session pool RTS success response
SPLRTSS2	Session pool RTS success response extension register
SPLRTSE	Session pool RTS error response
SPLRTSE2	Session pool RTS error response extension register
SPLTST	Session pool test request
SPLTST2	Session pool test request extension register
SPLTSTS	Session pool test success response
SPLTSTS2	Session pool test success response extension register

Table 185 OM group OSACSP

Register	Description
SPLTSTE	Session pool test error response
SPLTSTE2	Session pool test error response extension register

The following figure shows an example for OM group OSACSP.

Figure 309 MAP display example for OM group OSACSP

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       SPLAUD     SPLAUD2     SPLAUDS    SPLAUDS2
      SPLAUDE    SPLAUDE2      SPLBSY     SPLBSY2
      SPLBSYS    SPLBSYS2     SPLBSYE    SPLBSYE2
       SPLDRN     SPLDRN2      SPLRTS     SPLRTS2
      SPLRTSS    SPLRTSS2     SPLRTSE    SPLRTSE2
       SPLTST     SPLTST2     SPLTSTS    SPLTSTS2
      SPLTSTE    SPLTSTE2

3 SESNPL_3
           20           0          20           0
            0           0          10           0
           10           0           0           0
           27           0          10           0
           10           0           4           0
           13           0          13           0
            2           0
```

Associated OM groups

OSACSP is associated with the following OM groups:

- OSACCP1
- OSACCP2
- OSACND
- OSNND
- OSNSP
- OAPMTYPS
- OAPMERRS

OSNND

OM group OSNND (OSN Node) provides peg counts to track each OSN node class operation and response. Each OM tuple corresponds to an OSN node defined in table OANODINV.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 186 OM group OSNND

Register	Description
ONDDFL	OSN node datafill check request
ONDDFL2	OSN node datafill check request extension register
ONDDFLS	OSN node datafill check success response
ONDDFLS2	OSN node datafill check success response extension register
ONDDFLE	OSN node datafill check error response
ONDDFLE2	OSN node datafill check error extension register
ONDBYSI	OSN node busy inform
ONDBSYI2	OSN node busy inform extension register
ONDRTSI	OSN node RTS inform
ONDRTSI2	OSN node RTS busy extension register

The following figure shows an example for OM group OSNND.

Figure 310 MAP display example for OM group OSNND

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       ONDDFL     ONDDFL2     ONDDFLS    ONDDFLS2
      ONDDFLE    ONDDFLE2     ONDBSYI    ONSBSYI2
      ONDRTSI    ONDRTSI2

20 NODE_20
           27           0          27           0
            0           0          13           0
           13           0
```

Associated OM groups

OSNND is associated with the following OM groups:

- OSACCP1
- OSACCP2
- OSACND
- OSACSP
- OSNSP
- OAPMTYPN
- OAPMERRN

OSNSP

OM group OSNSP (OSN Session Pool) provides peg counts to track each OSN node class operation and response. Each OM tuple corresponds to an OSN session pool defined in table OASESNPL.

The following table describes each register. The register name corresponds to the operation or response sent or received by the switch.

Table 187 OM group OSNSP

Register	Description
OSPDFL	OSN session pool datafill check request
OSPDFL2	OSN session pool datafill check request extension register
OSPDFLS	OSN session pool datafill check success response
OSPDFLS2	OSN session pool datafill check success response extension register
OSPDFLE	OSN session pool datafill check error response
OSPDFLE2	OSN session pool datafill check error extension register
OSPBYSI	OSN session pool busy inform
OSPBSYI2	OSN session pool busy inform extension register
OSPRTSI	OSN session pool RTS inform
OSPRTSI2	OSN session pool RTS busy extension register

The following figure shows an example for OM group OSNSP.

Figure 311 MAP display example for OM group OSNSP

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

       OSPDFL     OSPDFL2     OSPDFLS    OSPDFLS2
      OSPDFLE    OSPDFLE2     OSPBSYI    ONSBSYI2
      OSPRTSI    OSPRTSI2

3 SESNPL_3
           27           0          27           0
            0           0          13           0
           13           0
```

Associated OM groups

OSNSP is associated with the following OM groups:

- OSACCP1
- OSACCP2
- OSACND
- OSACSP
- OSNND
- OAPMTYPN
- OAPMERRN

QMSMIS

OM group QMSMIS (Queue Management System Management Information System) provides peg counts on session (or agent) event and call queue messages generated by the OSSAIN (or TOPS) QMS MIS application.

Seven registers in OM group QMSMIS apply to OSSAIN. The following table describes these registers.

Table 188 OM group QMSMIS

Register	Description
SESNMSG	Session event message. This register is pegged each time an OSSAIN session event message is generated by the OSSAIN QMS MIS.
SESNMSG2	Session event message extension register
QUEMSG	Queue event message. This register is pegged each time a call queue event message is generated by the TOPS or OSSAIN QMS MIS.
QUEMSG	Queue event message extension register

Table 188 OM group QMSMIS

Register	Description
BUFFSX	Buffer sent. This register is pegged each time a QMSMIS buffer is sent to the TOPS or OSSAIN MIS application.
BUFFSX2	Buffer sent extension register
BUFFAIL	Buffer fail. This register is pegged each time the switch fails to send a QMSMIS buffer to the TOPS or OSSAIN MIS application.

The following figure shows an example for OM group QMSMIS.

Note: The TOPS QMS MIS application uses the 0 TOPS tuple. The OSSAIN QMS MIS application uses the 1 OSSAIN tuple.

Figure 312 MAP display example for OM group QMSMIS

```
CLASS: ACTIVE
START:1999/11/02 09:30:00 TUE; STOP: 1999/11/02 09:37:02 TUE;
SLOWSAMPLES:      5; FASTSAMPLES:         42;

      INFO (QMS_MIS_APPLN_INDEX_REGISTERINFO)
       POSMSG     POSMSG2     SESNMSG    SESNMSG2
       QUEMSG     QUEMSG2      BUFFSX     BUFFSX2
      BUFFAIL    BUFIP1SX    BUFIP1S2    BUFIP2SX
     BUFIP2S2    BUFIP3SX    BUFIP3S2    BUFIP4SX
     BUFIP4S2    BUFIP1TL    BUFIP1T2    BUFIP2TL
     BUFIP2T2    BUFIP3TL    BUFIP3T2    BUFIP4TL
     BUFIP4T2
0 TOPS
           15           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0
1 OSSAIN
            0           0          10           0
           22           0           5           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0
```

Associated OM groups

None

TOPPACT3

OM group TOPPACT3 (TOPS Open Position Protocol Action Identifier) is a continuation of OM group TOPPACT1. It provides peg counts for each type of OPP action identifier (actID) sent from an OPP-compatible position to the TOPS DMS switch. ActIDs are sent from the operator position to the switch to update both call information and other information. A register is pegged only when a valid actID is sent.

Four registers in OM group TOPPACT3 apply to OSSAIN. The following table describes these registers.

Table 189 OM group TOPPACT3

Register	Description
DATAA	Data actID. This register is pegged each time the operator position sends the data actID to the DMS switch.
DATAA2	Data actID extension register
OTRIGRA	OSSAIN trigger actID. This register is pegged each time the operator position sends the OSSAIN trigger actID to the DMS switch.
OTRIGRA2	OSSAIN trigger actID extension register

The following figure shows an example for OM group TOPPACT3.

Figure 313 MAP display example for OM group TOPPACT3

```
CLASS: ACTIVE
START:1999/06/14 00:30:00 WED; STOP: 1999/06/14 00:44:51 WED;
SLOWSAMPLES:       2; FASTSAMPLES:      18;

         BCDBA      BCDBA2     FIXDURA    FIXDURA2
       FORASTA    FORASTA2     TKTNUMA    TKTNUMA2
         DATAA      DATAA2     OTRIGRA    OTRIGRA2
       ATTOPRA    ATTOPRA2

0            0           0           0           0
             0           0           0           0
            10           0           5           0
             0           0
```

Associated OM groups

TOPPACT3 is associated with the following OM groups:

- TOPPACT1
- TOPPACT2

TOPPACT4

OM group TOPPACT4 (TOPS Open Position Protocol Action Identifier) is a continuation of OM group TOPPACT1. It provides peg counts for each type of OPP action identifier (actID) sent from an OPP-compatible position to the TOPS DMS switch. ActIDs are sent from the operator position to the switch to update both call information and other information. A register is pegged only when a valid actID is sent.

Two registers in OM group TOPPACT4 apply to OSSAIN. The following table describes these registers.

Table 190 OM group TOPPACT4

Register	Description
CT4QA	Call type for queuing actID. This register is pegged each time the operator position sends the queuing actID to the DMS switch.
CT4QA2	Call type for queuing actID extension register.

The following figure shows an example for OM group TOPPACT4.

Figure 314 MAP display example for OM group TOPPACT4

```
CLASS:   ACTIVE
START:1999/10/26 09:00:00 TUE; STOP: 1999/10/26 09:15:03 TUE;
SLOWSAMPLES:         9 ; FASTSAMPLES:         90 ;
            CT4QA     CT4QA2
```

Associated OM groups

None

TOPPDID5

OM group TOPPDID5 (TOPS Open Position Protocol Data Identifier) is a continuation of OM group TOPPDID1. It provides peg counts for each type of OPP data identifier (DID) sent from the TOPS DMS switch. DIDs are sent from the switch to the operator position to update both call information and other information. A register is pegged only when a valid DID is sent.

Six registers in OM group TOPPDID5 apply to OSSAIN. The following table describes these registers.

Table 191 OM group TOPPDID5

Register	Description
DATAD	Data DID. This register is pegged each time the DMS switch sends the DID to the operator position.

Table 191 OM group TOPPDID5

Register	Description
DATAD2	Data DID extension register
OINFOD	OSSAIN information DID. This register is pegged each time the DMS switch sends the OSSAIN information DID to the operator position.
OINFOD2	OSSAIN information DID extension register
OTRIGRD	OSSAIN trigger DID. This register is pegged each time the DMS switch sends the OSSAIN trigger DID to the operator position.
OTRIGRD2	OSSAIN trigger DID extension register

The following figure shows an example for OM group TOPPDID5.

Figure 315 MAP display example for OM group TOPPDID5

```
CLASS: ACTIVE
START:1999/05/19 16:30:00 WED; STOP: 1999/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      XFROPRD    XFROPRD2    OPRQPRD    OPRQPRD2
      ALTRTED    ALTRRED2     DBACTD     DBACTD2
      DBCLASD    DBCLASD2     DBNUMD     DBNUMD2
      FIXDURD    FIXDURD2    FORASTD    FORASTD2
      POSINFD    POSINFD2    TKTNUMD    TKTNUMD2
       DBQRYD     DBQRYD2      DATAD      DATAD2
       OINFOD     OINFOD2    OTRIGRD    OTRIGRD2
      CTRYNMD    CTRYNMD2

0           0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0           0           0
            0           0          10           0
            1           0           2           0
            0           0
```

Associated OM groups

TOPPDID5 is associated with the following OM groups:

- TOPPDID1
- TOPPDID2
- TOPPDID3
- TOPPDID4

TOPSRLT2

OM group TOPSRLT2 (TOPS Release Link Trunking Variant 2) provides peg counts to track attempts, successes, failures, and abandons in performing RLT (variant 2).

Each OM tuple corresponds to an ISUP trunk datafilled in table ISUPTRK. However, only ISUP trunks with variant 2 datafilled (RLT_FAR) peg registers in the TOPSRLT2 OM group. The following table describes each register.

Table 192 OM group TOPSRLT2

Register	Description
TRNSATT	Transfer attempt. This register is pegged each time an RLT transfer is attempted.
TRNSATT2	Transfer attempt extension register
TRNSSCS	Transfer success. This register is pegged each time an RLT transfer succeeds.
TRNSSCS2	Transfer success extension register
TRNSFAIL	Transfer fail. This register is pegged each time an RLT transfer fails.
TRNSABDN	Transfer abandon. This register is pegged each time an RLT transfer is abandoned because the originator went onhook.
BRDGATT	Bridging attempt. This register is pegged each time an RLT bridging is attempted.
BRDGATT2	Bridging attempt extension register
BRDGSCS	Bridging success. This register is pegged each time an RLT bridging succeeds.
BRDGSCS2	Bridging success extension register
BRDGFAIL	Bridging fail. This register is pegged each time an RLT bridging fails.
BRDGABDN	Bridging abandon. This register is pegged each time an RLT bridging is abandoned because either party went onhook.

The following figure shows an example for OM group TOPSRLT2.

Figure 316 MAP display example for OM group TOPSRLT2

```
CLASS: ACTIVE
START:1997/05/19 16:30:00 WED; STOP: 1997/05/19 16:33:00 WED;
SLOWSAMPLES:      2; FASTSAMPLES:         18;

      TRNSATT    TRNSATT2     TRNSSCS     TRNSCCS2
      TRNSFAIL   TRNSABDN     BRDGATT     BRDGATT2
      BRDGSCS    BRDGSCS2     BRDGFAIL    BRDGABDN

225 ISUP2WIT
      71         0            70          0
      1          0            23          0
      23         0            0           0
```

Associated OM groups

TOPSRLT2 is associated with the TOPSISUP OM group. However, only ISUP trunks with variant 1 datafilled (RLT_REL) peg registers in the TOPSISUP OM group.

Appendixes

The following appendixes are included in *OSSAIN User's Guide*:

Appendix A: "OSSAIN data communications," beginning on page 585.

Appendix B: "Advanced OSAC configuration," beginning on page 597.

Appendix C: "Provisioning broadcast voice links in a JNET office," beginning on page 605.

Appendix D: "OSSAIN datafill example for Billing and Access Services," beginning on page 623.

Appendix E: "DA Sequence Calling," beginning on page 647.

Appendix A: OSSAIN data communications

This appendix discusses data communications in the OSSAIN network. It includes the following:

- brief discussions of the OSSAIN data network and the software architecture and protocols used by OSSAIN
- tutorial introduction to IP addressing
- UDP port numbers used by OSSAIN
- high-level overview of the datafill required for OSSAIN data communications

The information in this appendix is largely tutorial and/or is also covered by other chapters in this book.

Additional documentation

While *OSSAIN User's Guide* provides detailed information on how OSSAIN functionalities work together to deliver services, it does not provide more than an overview of the OSSAIN Open Automated Protocol (OAP) and the Internet Protocol (IP). For more detailed information on these components, please refer to the documents listed in the following table.

Table 193 Additional documentation

Component	Document
OAP	*OSSAIN Open Automated Protocol Specification*, Q235-1 ***Note:*** OAP is a licensed interface. To license OAP and receive the specification document, please contact TOPS Marketing.
IP	Many books provide information about the IP protocol suite. One example is: *TCP/IP Network Administration*, by Craig Hunt (O'Reilly and Associates, Inc. 2002)

OSSAIN data network

When OSSAIN is used in standalone mode, a single switch uses an Ethernet LAN to exchange call control and maintenance messages with one or more service nodes (SN). The SNs connect to the LAN using standard Ethernet technology. The switch computing module (CM) uses either Ethernet interface units (EIU) or XA-Core Ethernet interface cards such as HIOP or HCMIC to access the LAN.

- EIUs are provisioned on either a Link Peripheral Processor (LPP) or Fiber Link Interface Shelf (FLIS).
- XA-Core Ethernet interface cards are provisioned in the core, and provide a direct data path between the XA-Core and the LAN.

In an OSAC network, a wide area network (WAN) is used to transport OSSAIN messages between the LANs to which the switches and SNs are connected. Industry-standard routers connect the LANs to the WAN. (Although the EIU is technically a router, it is only intended to be used for routing among hosts within a DMS SuperNode.) OSSAIN imposes no constraints on the WAN technology used, as long as it has sufficient capacity and reliability to handle the messaging.

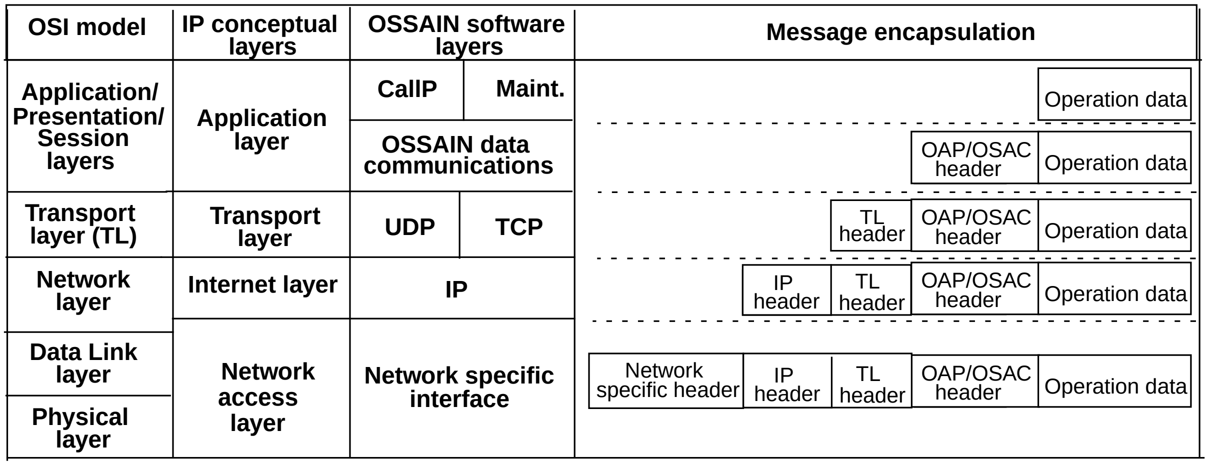
Data communications software architecture and protocols

This section briefly discusses the OSSAIN layered software architecture and the protocols used at each layer.

Data communications software architectures are standardly modeled using layers. Each layer uses a layer-specific protocol to communicate with its peer layer in the far-end node. Also, each layer uses a well defined interface to communicate with the layers above and below it within the same node. As an outbound application-layer message is passed down through the layers on a single host, additional headers are prepended to the message. As an inbound message is passed up through the layers, headers are stripped. Another way of saying this is that each lower layer encapsulates the message from the higher layer protocol.

Two well know models of data communications software layering are the ISO Open System Interconnection (OSI) model and the IP model. The following figure shows the correspondence between the conceptual layers of these two models and the OSSAIN software architecture. The figure also shows the message encapsulation that occurs at each layer. The figure is followed by a discussion of each layer.

Figure 317 OSSAIN messaging data encapsulation



Application layer—OAP and OSAC protocols

As shown in the figure, OSSAIN divides the standard application layer into two sub-layers. The higher sublayer is where the real application logic resides. In this layer, decisions are made about the call flow or maintenance flow, including the determination of what messages need to be sent to what far-end nodes, and when. “Operation data” in the figure refers to application-layer requests, responses, and information messages that control the flow of a call or that control maintenance states.

The OSSAIN data communications layer routes messages between OSSAIN call and maintenance processes, on one hand, and lower-layer switch transport software on the other hand. It performs tasks such as constructing an application-layer header for each outgoing message, verifying the sanity of the application-layer header for each incoming message, and pegging operational measurements. For incoming messages, it also determine which OSSAIN call or which maintenance process should receive the message.

Protocols used at the application layer to communicate with external nodes include OAP, which is used for messaging between switches and SNs, and OSAC, which is used for messaging between switches. Since the OSAC protocol is used only by DMS switches, it is not described in customer documentation. OAP is a licensable interface. For more information about OAP, see “OAP overview” on page 118.

Transport layer—UDP and TCP protocols

At and below the transport layer, OSSAIN makes use of software in the TL and BASE layers. This software is not specific to OSSAIN, and has no knowledge of the OSSAIN application.

The most essential function of the transport layer is to route incoming messages to the right application, once the message has reached the destination node. Beyond that, depending on the specific transport protocol used, the layer may provide additional functionality.

The two main transport protocols in the IP suite are UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). OSSAIN uses UDP for virtually all of its messaging. The only exception is the OSSAIN QMS MIS application, which uses TCP for messaging between the MIS node and the OSAC host or standalone OSSAIN switch. UDP is still used for maintenance of MIS nodes, and also for the QMS MIS messages that are exchanged between OSAC remotes and OSAC hosts.

Note: The OSSAIN QMS MIS application is scheduled for end of life in a future release. It is not recommended that operating companies make plans based on the assumption of its future availability.

The following subsections provide more information about the UCP and TCP protocols.

UDP

UDP is a datagram-oriented transport layer protocol that adds little to the underlying IP datagram delivery service. UDP permits packets to be sent with a minimum of protocol overhead. With UDP, message delivery is not guaranteed. There is no checking for missing, out-of-sequence, or duplicated packets. No acknowledgments are sent at the data transport level.

A checksum is sent with each datagram packet. If the checksum calculated by the receiving station does not agree with the checksum in the packet, the message is discarded.

OSSAIN uses UDP because of its simplicity and low overhead. To partially compensate for UDP's lack of guaranteed delivery and sequencing, the OAP protocol includes message sequencing capabilities. This allows SNs to determine if message loss has occurred.

TCP

TCP is a stream-oriented transport layer protocol that builds the underlying IP datagram delivery service. TCP adds reliability through sequencing, timeouts, and retransmissions. It provides acknowledgments and checks for missing, out-of-sequence, and duplicated packets.

TCP establishes connections in a host-remote manner. The host (MIS node) waits for a connection attempt from a client (DMS switch).

As with UDP, a checksum is sent with each datagram packet. If the checksum calculated by the receiving station does not agree with the checksum in the packet, the message is discarded.

Message sequencing is provided by both TCP and OAP. TCP ensures that the message is received and its parts are in the correct order. OAP detects when messages are unable to be sent due to lack of TCP resources.

The processing overhead for each TCP connection in the core is high. OSSAIN uses TCP only for messaging between OSSAIN QMS MIS servers and standalone or OSAC host switches.

Network / internet layer—IP protocol

The main function of the network or internet layer is to route messages to the correct destination nodes. In the IP protocol suite, routing is based on IP addresses. The following major section provides more information about IP addresses.

Network access layer

In the LAN, OSSAIN uses Ethernet at the network access layer. In an OSAC network, OSSAIN imposes no constraints on the WAN technology as long as it has the bandwidth and reliability needed to handle the messaging.

IP addressing

Each node in an IP network is known by its IP address. This section provides an introduction to IP addressing. It discusses IP address classes, subnets, and masks. The material is not specific to OSSAIN.

An IP address is a 32-bit number that uniquely identifies a network and a specific node on that network. IP addresses are usually written as four decimal numbers separated by periods. Each of the four numbers is in the range of 0 to 255, the decimal range for a single byte or octet. (8 bits equal one byte; a byte is also known as an octet.)

Following is an example of an IP address: 47.233.1.52

IP address classes

An IP address consists of a network part and a host part. There are three main IP address classes: class A, class B, and class C. Each class uses a different number of bits to specify the network part and the host part of the IP address.

The first few bits of an IP address designate the address class, as follows:

- If the first bit of an IP address, expressed as a binary number, is 0, then the address class is class A. The next 7 bits identify the network, and the last 24 bits identify the host.
- If the first two bits of an IP address, expressed as a binary number, are 10, then the address class is class B. The next 14 bits identify the network, and the last 16 bits identify the host.

- If the first three bits of an IP address, expressed as a binary number, are 110, then the address class is class C. The next 21 bits identify the network, and the last 8 bits identify the host.

Because address class bits of an IP address adjoin the network ID bits, the first byte of an IP address can be viewed as a full byte in a network address.

The following table provides a summary of how the IP address classes are divided into network parts and host parts.

Table 194 Internet address classes

	Class A	Class B	Class C
Network portion	octet 1	octets 1, 2	octets 1, 2, 3
Network range	1 to 126 (see Note 1)	128.1 to 191.254	192.1.1 to 223.254.254
Host portion	octets 2, 3, 4	octets 3, 4	octet 4
Host range	1.1.1 to 254.254.254	1.1 to 254.254	1 to 254

Note 1: Although127 expressed as a binary octet does begin with 0, it is not included in class A addresses. It is reserved for the host's loopback address, which is used for internal diagnostics and inter-process communication.

Note 2: 0 and 255 are reserved and should not be used within actual IP addresses.

Subnets and masks

A subnet is a network *within* the network designated by the network part of the IP address. The standard IP address structure can be extended by using some of the host address bits as additional network address bits. The additional network bits identify the subnets. This is referred to as subnetting. Subnetting allows for further refinement of an organization's network.

Subdividing the internet address space is accomplished by network and subnetwork masks. The following paragraphs define the masks.

- Network mask

  Network mask is a number of 1 bits that starts at the first upper bit of the standard network part of the address space and continues up to and including the last lower bit of the network space. Following are examples of network masks:

  — class A: 255.0.0.0

  — class B: 255.255.0.0

  — class C: 255.255.255.0

- Subnet mask

  Subnet mask is a number of 1 bits that starts at the first upper bit of the host part of the address space The number of bits in the subnet mask determines the number of possible subnets within the network. Following are examples of subnet masks:

- class A: 0.255.255.0 (16 bits)
- class A: 0.255.0.0 (8 bits)
- class B: 0.0.255.0 (8 bits)
- class C: 0.0.0.240 (4 bits)

- Netmask

  Netmask is a mask that includes the network mask *ORed* with the subnet mask. Following are examples of netmasks:

  - class A: 255.255.255.0 (16-bit subnet mask)
  - class A: 255.255.0.0 (8-bit subnet mask)
  - class B: 255.255.255.0 (8-bit subnet mask)
  - class C: 255.255.255.240 (4-bit subnet mask)

The network part and host part of an IP address are determined by applying a netmask to an IP address. If a bit in the mask is on, then the bit belongs to the network portion of the address. If the bit is off, then the bit belongs to the host portion of the address.

For example, when the class B netmask 255.255.255.0 is applied to IP address 128.66.12.1, the network part has three octets and the host part has one octet. The following IDs are determined:

- the network ID is 128.66
- the subnet ID is 12
- the host ID is 1

Note 1: For simplicity, most of the subnet masks used in the examples are an integral number of octets in size. In practice, the subnet mask does not have to be—and usually is not—an integral number of octets.

Note 2: It is not uncommon in the literature to see the terms *network mask* and *subnet mask* both used to refer to what has been described here as a netmask.

The following table illustrates network, subnet, and host ID ranges for some example class A, class B, and class C networks.

Table 195 Example netmasks

	Class A	Class B	Class C
Subnet mask size	16-bit	8-bit	a. 2-bit b. 3-bit c. 4-bit d. 5-bit
Netmask	255.255.255.0	255.255.255.0	a. 255.255.255.192 b. 255.255.255.224 c. 255.255.255.240 d. 255.255.255.248
Network ID range	1 to 127	128.1 to 191.254	192.1.1 to 223.254.254
Subnet ID range	1.1 to 254.254	1 to 254	a. highest 2 bits for 4 subnets b. highest 3 bits for 8 subnets c. highest 4 bits for 16 subnets d. highest 5 bits for 32 subnets
Host ID range	1 to 254	1 to 254	a. lowest 6 bits for 64 hosts b. lowest 5 bits for 32 hosts c. lowest 4 bits for 16 hosts d. lowest 3 bits for 8 hosts

Data port numbers

Once data is delivered to a specific node in an IP network, it must be passed to the correct application on that node. This is a function of the transport layer. For both UDP and TCP, it is done by using a 16-bit port number. The combination of IP address and port number uniquely identifies an application in an IP network.

Consider the following scenario to illustrate data port addressing. An SN has an IP address of 47.233.1.52 and supports Application A and Application B. For example, one application might be call processing and the other might be node maintenance. Application A uses port 9000 and Application B uses port 10,000. To fully address Application A when sending a message, the originator of the message would specify 47.233.1.52 port 9,000. Likewise, to fully address Application B, the originator would specify 47.233.1.52 port 10,000.

IP and data port assignments must be coordinated between the DMS switch and the SNs.

Port numbers used in the switch for OSSAIN are hard-coded to the following values.

- 5289 for the call processing port
- 5290 for the node maintenance port
- 5291 for the session pool maintenance port
- 5292 for the node log and session pool log port
- 5293 for the node alarm and session pool alarm port

The port numbers shown above are used by SNs when they send messages to a switch (standalone, OSAC host, or OSAC remote). They are also used in an OSAC network for inter-switch OSSAIN messaging.

Port numbers used in SNs are determined by the SN developer and datafilled in the switch. These are the ports that the switch uses when sending messages to the SN. SN developers can use ports in the range 1024 to 32,767.

Note 1: Ports 0 through 1023 are well known port addresses and are reserved for applications such as FTP (File Transfer Protocol), Telnet, and NFS (Network File System).

Note 2: Due to table control restrictions, the largest port number allowed to be datafilled is 32,767.

OSSAIN datafill for data communications

This section provides an overview of all the OSSAIN datafill that is needed for data communications. It includes subsections for the following:

- node and session pool inventory tables
- datafill specific to EIU
- datafill specific to XA-Core Ethernet interface

Node and session pool inventory tables

Table OANODINV (OSSAIN Node Inventory) includes a tuple for each SN and OSAC switch with which the switch can exchange messages. The datafill includes the following IP-specific information:

- whether the switch uses EIU or XA-Core Ethernet interface to exchange messages with the node
- if the switch uses XA-Core Ethernet interface, which of the switch's CM IP addresses (from table CMIPADDR) the switch uses for communication with that node
- the transport level protocol used for messaging with the node (always UDP)
- the IP address of the node

- the SN's UDP port used for node maintenance messages (OSNM nodes only)

  ***Note 1**:* Although an OSAC remote switch has no datafill for SN maintenance ports, it may need to send messages to those ports. At SN07 and higher, an OSAC remote can learn the SN's maintenance port number by a message exchange with the OSAC host.

  ***Note 2**:* Ports used in OSAC switches are not datafilled because they are hardcoded to the values listed in the previous section.

- whether the switch should make the node system busy if the switch is unable to process a threshold percentage of messages from the node

Table OASESNPL (OSSAIN Session Pool) includes a tuple for each session pool. Both SN session pools and OSAC session pools are datafilled here. For SN session pools, the datafill includes the UDP ports used on the SN for CallP messages and for session pool maintenance messages. (An exception is that session pool maintenance ports are not datafilled in OSAC remotes for centralized session pools that handle subscriber-originated calls. This is because an OSAC remote does not send session pool maintenance messages to OSNs.)

Note: Refer to Chapter 7: "OSSAIN data schema," for detailed information on datafilling tables OANODINV and OASESNPL.

Datafill specific to EIU

When OSSAIN uses the EIU to route IP traffic to the Ethernet, the following IP-related data tables also require datafill:

- LIUINV, in which all EIU hardware information is datafilled.
- IPNETWRK, which stores information about the IP network and the SuperNode subnet. Table IPNETWRK specifies the CM IP address, the netmask size, and the default EIU router to use when sending messages to destinations not on one of the local subnets. This table is also used to enable or disable EIU screening.
- IPHOST, in which a tuple must be added for the CM.
- IPROUTER, which stores the internet information for each EIU acting as an IP router between the DMS CM and the Ethernet LAN.
- IPTHRON, which specifies the IP throttling numbers. IP messages between the CM and EIU require throttling to control congestion in the Local Message Switch (LMS) of the LPP and Message Switch (MS) of the SuperNode.

Note: Refer to Chapter 6: "OSSAIN engineering," for detailed information on datafilling these tables.

Datafill specific to XA-Core Ethernet interface

When OSSAIN uses XA-Core Ethernet interface, the following IP-related data table requires datafill:

- CMIPADDR, which specifies the IP addresses and netmask associated with the XA-Core, as well as the gateway router IP address and the edge device IP addresses and netmasks.

Note: Refer to Chapter 6: “OSSAIN engineering,” for detailed information on datafilling this table.

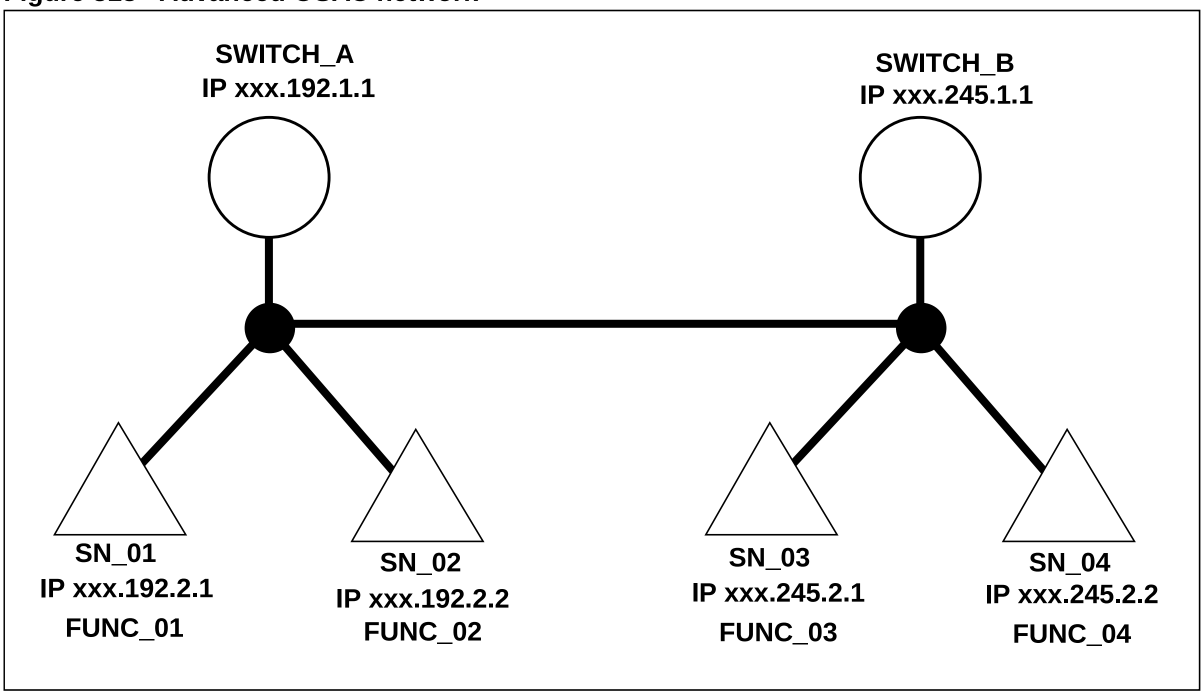
Appendix B: Advanced OSAC configuration

This appendix illustrates an example of an advanced OSAC network configuration and its datafill.

Advanced OSAC network example

The following figure shows two switches and four SNs. Each switch is provisioned with two centralized SNs. The switches are interconnected to use the functions (services) provided by the other switch's SNs. (The lines show data connections only).

Figure 318   Advanced OSAC network



In this configuration, a given switch can act as a standalone, an OSAC host, or an OSAC remote, as follows:

- Acting as a *standalone* switch—when the switch is using the functions of it own SNs. For example, at SWITCH_A, if a call routes to SN_01, then SWITCH_A processes the call in standalone mode.
- Acting as an *OSAC host* switch—when the *other* switch is using the functions of SNs provisioned on this switch. For example, at SWITCH_B, if a call routes to SN_01, then SWITCH_A processes the call in OSAC host mode.
- Acting as an *OSAC remote* switch—when this switch is using the functions of SNs provisioned on the other switch. For example, at SWITCH_B, if a call routes to SN_01, then SWITCH_B processes the call in OSAC remote mode.

Datafill examples

This section shows example datafill for SWITCH_A and SWITCH_B in each of the following tables:

- OANODNAM
- OANODINV
- OASESNPL
- OAFUNDEF
- OSCVLGRP
- OAVLMAP

Refer to Chapter 7: “OSSAIN data schema,” for details on the fields and values for each table.

OANODNAM

Table OANODNAM is datafilled with node identifiers and node names for each node in the network.

- SWITCH_A example

Figure 319 MAP display example for table OANODNAM—SWITCH_A

```
1           SWITCH_A
2           SWITCH_B
101         SN_01
102         SN_02
103         SN_03
104         SN_04
```

- SWITCH_B example

Figure 320 MAP display example for table OANODNAM—SWITCH_B

```
1           SWITCH_A
2           SWITCH_B
101         SN_01
102         SN_02
103         SN_03
104         SN_04
```

OANODINV

Table OANODINV is datafilled with details on the other switch and each SN in the network.

- SWITCH_A example

  The datafill on SWITCH_A includes information on SWITCH_B, the two centralized SNs provisioned on SWITCH_A (OSNM) and the two centralized SNs provisioned on SWITCH_B (OSN). Datafilling the two OSN nodes allows SWITCH_A to use the functions of SN_03 and SN_04.

Figure 321 MAP display example for table OANODINV—SWITCH_A

```
NODENAME  NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_A OSAC SELF  9          8         5
SWITCH_B OSAC OTHER 0 UDP IPV4 xxx 245 1 1 SWITCH 2 BB 11 CITYB REMOTEOSAC 9 8 5
SN_01   OSNM      0 UDP IPV4 xxx 192 2 1 7000 Y 2 240 60 SN 4 BB 3 CITYA FUNC1 8 6 60
SN_02   OSNM      1 UDP IPV4 xxx 192 2 2 7000 Y 2 240 30 SN 1 AA 1 CITYA FUNC2 8 6 60
SN_03    OSN      0 UDP IPV4 xxx 245 2 1 SWITCH_B Y 2 SN 7 LL 2 CITYB FUNC3 8 6 60
SN_04    OSN      1 UDP IPV4 xxx 245 2 2 SWITCH_B Y 2 SN 8 BB 4 CITYB FUNC4 8 6 60
```

- SWITCH_B example

  Likewise, the datafill on SWITCH_B includes information on SWITCH_A, the two centralized SNs provisioned on SWITCH_A (OSN) and the two centralized SNs provisioned on SWITCH_B (OSNM). Datafilling the two OSN nodes allows SWITCH_B to use the functions of SN_01 and SN_02.

Figure 322 MAP display example for table OANODINV—SWITCH_B

```
NODENAME NODEAREA   AUDTRIES AUDRTIMR AUDFREQ
--------------------------------------------------------------------------------
SWITCH_A OSAC OTHER 0 UDP IPV4 xxx 192 1 1 SWITCH 4 CC 12 CITYA REMOTEOSAC 9 8 5
SWITCH_B OSAC SELF  9        8        5
SN_01    OSN       0 UDP IPV4 xxx 192 2 1 SWITCH_A Y 2 SN 4 BB 3 CITYA FUNC1 8 6 60
SN_02    OSN       1 UDP IPV4 xxx 192 2 2 SWITCH_A Y 2 SN 1 AA 1 CITYA FUNC2 8 6 60
SN_03   OSNM      0 UDP IPV4 xxx 245 2 1 7000 Y 2 240 60 SN 7 LL 2 CITYB FUNC3 8 6 60
SN_04   OSNM      1 UDP IPV4 xxx 245 2 2 7000 Y 2 240 60 SN 8 BB 4 CITYB FUNC4 8 6 60
```

OASESNPL

Table OASESNPL is datafilled with details on the SN session pools and the host-remote session pool to the other switch in the network.

- SWITCH_A example

  The datafill on SWITCH_A includes the host-remote session pool to SWITCH_B (A_TO_B_SP) and the host-remote session pool from SWITCH_B (A_FROM_B_SP).

Figure 323 MAP display example for table OASESNPL—SWITCH_A

```
SESNPLID  SESNPLNM   MAXSESN  NODENAME  ORIGAREA
-------------------------------------------------------------------------------
1     A_TO_B_SP      20    SWITCH_B  OSACORIG
2     A_FROM_B_SP    20    SWITCH_B  OSACTERM
10    SN_01_SP      20   SN_01 SUBSCRIBER SH USEDEFLT USEDEFLT N 5 UDP 7001 7002
11    SN_02_SP      20   SN_02 SUBSCRIBER SH USEDEFLT USEDEFLT N 7 UDP 7001 7002
12    SN_03_SP       20    SN_03 SUBSCRIBER  R USEDEFLT USEDEFLT N UDP 7001
13    SN_04_SP       20    SN_04 SUBSCRIBER  R USEDEFLT USEDEFLT N UDP 7001
```

- SWITCH_B example

  Likewise, the datafill on SWITCH_B includes the host-remote session pool to SWITCH_A (B_TO_A_SP) and the host-remote session pool from SWITCH_A (B_FROM_A_SP).

Figure 324 MAP display example for table OASESNPL—SWITCH_B

```
SESNPLID  SESNPLNM   MAXSESN  NODENAME  ORIGAREA
-------------------------------------------------------------------------------
1     B_FROM_A_SP    20    SWITCH_A  OSACTERM
2     B_TO_A_SP      20    SWITCH_A  OSACORIG
10    SN_01_SP       20    SN_01 SUBSCRIBER R USEDEFLT USEDEFLT N UDP 7001
11    SN_02_SP       20    SN_02 SUBSCRIBER R USEDEFLT USEDEFLT N UDP 7001
12    SN_03_SP      20    SN_03 SUBSCRIBER  SH USEDEFLT USEDEFLT N 6 UDP 7001 7002
13    SN_04_SP      20    SN_04 SUBSCRIBER  SH USEDEFLT USEDEFLT N 8 UDP 7001 7002
```

OAFUNDEF

Table OAFUNDEF is datafilled with the functions provided by the SNs in the network.

- SWITCH_A example

  The datafill on SWITCH_A specifies the QMS CAM and the call queue at SWITCH_A for the OSNM nodes. For the OSN nodes, the QMS CAM is not at SWITCH_A and the host-remote session pool to SWITCH_B is specified.

Figure 325 MAP display example for table OAFUNDEF—SWITCH_A

```
FUNCID     FUNCNAME              FUNCAREA
-------------------------------------------------------------------------------------------
1          FUNCTION1             SN TASERV N N N N Y Y CQ1 Y Y
2          FUNCTION2             SN TASERV N N N N Y Y CQ2 N
3          FUNCTION3             SN TASERV N N N N N Y N A_TO_B_SP
4          FUNCTION4             SN TASERV N N N Y N A_TO_B_SP
```

- SWITCH_B example

  The datafill on SWITCH_B specifies the QMS CAM and the call queue at SWITCH_B for the OSNM nodes. For the OSN nodes, the QMS CAM is not at SWITCH_B and the host-remote session pool to SWITCH_A is specified.

Figure 326 MAP display example for table OAFUNDEF—SWITCH_B

```
FUNCID     FUNCNAME              FUNCAREA
-------------------------------------------------------------------------------------------
1          FUNCTION1             SN TASERV N N N N Y N B_TO_A_SP
2          FUNCTION2             SN TASERV N N N N Y N B_TO_A_SP
3          FUNCTION3             SN TASERV N N N N Y Y CQ3 Y N
4          FUNCTION4             SN TASERV N N N N Y Y CQ4 N
```

OSCVLGRP

Table OSCVLGRP is datafilled with the host-remote voice link group to use between SWITCH_A and SWITCH_B.

- SWITCH_A example

Figure 327 MAP display example for table OSCVLGRP—SWITCH_A

```
NODENAME          CLLI
----------------------------------------------------------------------
SWITCH_B          OSACVL_B
```

- SWITCH_B example

Figure 328 MAP display example for table OSCVLGRP—SWITCH_B

```
NODENAME          CLLI
----------------------------------------------------------------------
SWITCH_A          OSACVL_A
```

OAVLMAP

Table OAVLMAP is datafilled with the logical SN voice channels and the logical host-remote voice channel.

Note: Tables CLLI, TRKGRP, TRKSGRP, and TRKMEM also require datafill.

- SWITCH_A example

Figure 329 MAP display example for table OAVLMAP—SWITCH_A

```
NDANCH                 CLLI           EXTRKNUM   BCSTAREA
----------------------------------------------------------------------
SWITCH_B   1           OSACVL_B       1          N
SWITCH_B   2           OSACVL_B       2          N
SWITCH_B   3           OSACVL_B       3          N
SWITCH_B   4           OSACVL_B       4          N
SN_01      1           SNVL_01        1          Y   HKCHG 200  N
SN_01      2           SNVL_01        2          N
SN_01      3           SNVL_01        3          N
SN_01      4           SNVL_01        4          N
SN_02      1           SNVL_02        1          N
SN_02      2           SNVL_02        2          N
SN_02      3           SNVL_02        3          N
SN_02      4           SNVL_02        4          N
```

- SWITCH_B example

Figure 330 MAP display example for table OAVLMAP—SWITCH_B

```
NDANCH                 CLLI           EXTRKNUM   BCSTAREA
----------------------------------------------------------------------
SWITCH_A   1           OSACVL_A       1          N
SWITCH_A   2           OSACVL_A       2          N
SWITCH_A   3           OSACVL_A       3          N
SWITCH_A   4           OSACVL_A       4          N
SN_03      1           SNVL_03        1          N
SN_03      2           SNVL_03        2          N
SN_03      3           SNVL_03        3          N
SN_03      4           SNVL_03        4          N
SN_04      1           SNVL_04        1          N
SN_04      2           SNVL_04        2          N
SN_04      3           SNVL_04        3          N
SN_04      4           SNVL_04        4          N
```

Appendix C: Provisioning broadcast voice links in a JNET office

In a JNET (junctored network) office, a maximum of 255 simultaneous connections can be made to a broadcast voice link. Certain engineering guidelines must be followed so that no more than 256 (for NT0X48 networks) or 512 (for NT5X13 or NT8X11 networks) simultaneous connections (broadcast *and* non-broadcast) are attempted on a network subgroup (NSG).

This appendix discusses how to determine the total number of broadcast connections the JNET can support, and how to assign the broadcast connections. Information is provided on the following areas:

- JNET terminology
- time switching
- network blocking
- port deloading
- provisioning rules
- PM port and channel mapping
- provisioning examples

JNET terminology

The following table defines JNET terms.

Table 196 JNET terms

Term	Definition
Bus	A pathway used to get from one network module (NM) crosspoint card to another. There are 2048 unique bus ports and channels on an NM.
C-side port	A port going to the control side of a switching entity. In the context of a peripheral module (PM), the C-side port interfaces to the NM.
Crosspoint card	An NM card that contains the time switch for 512 channels (one quarter of one side of an NM).
DS-1 port	A P-side port on a PM consisting of 24 channels.

Table 196 JNET terms

Term	Definition
DS30 port	A port consisting of 32 channels between a PM and a junctor-type network. A DS30 port also is used in other places throughout the DMS network to connect two switching entities together.
From pathend	One of the two ends, or terminals, involved in a network connection. In voice link broadcasting, it refers to the voice link side of the connection.
Junctor	A port consisting of 32 channels (one DS30) used to connect two NMs together.
Network blocking	A condition whereby no bus channels exist from an incoming crosspoint card to an outgoing crosspoint card.
Network broadcast port	A DS30 port between a PM and an NM that has broadcast voice links assigned to at least one of its channels.
Network subgroup	A group of PM ports (either 8 or 16) on a JNET that are associated with a particular crosspoint card.
P-side port	A port going to the peripheral side of a switching entity. In the context of an NM, the P-side port interfaces to a PM. In the context of a PM, the P-side port interfaces to the terminal.
Port deloading	The act of leaving a network P-side port unassigned in datafill (in table LTCINV), so that there is at least one bus channel for every terminal on the NSG.
Port	An interface consisting of a group of channels.
Time slot	An assignment in time of a port and channel on the bus between two crosspoint cards or in a DS-1/DS30 stream.
Time switch	The circuits in an NM responsible for switching pulse code modulated (PCM) speech samples from a particular time slot (“from pathend” port and channel) to another time slot (“to pathend” port and channel). It consists of two cards—the incoming and outgoing crosspoint cards.
To pathend	One of the two ends, or terminals, involved in a network connection. In voice link broadcasting, it refers to the terminal that is being broadcasted to, which typically is a TOPS trunk.

Network module A-side and B-side

A single JNET connection between two pathends consists of the following concepts:

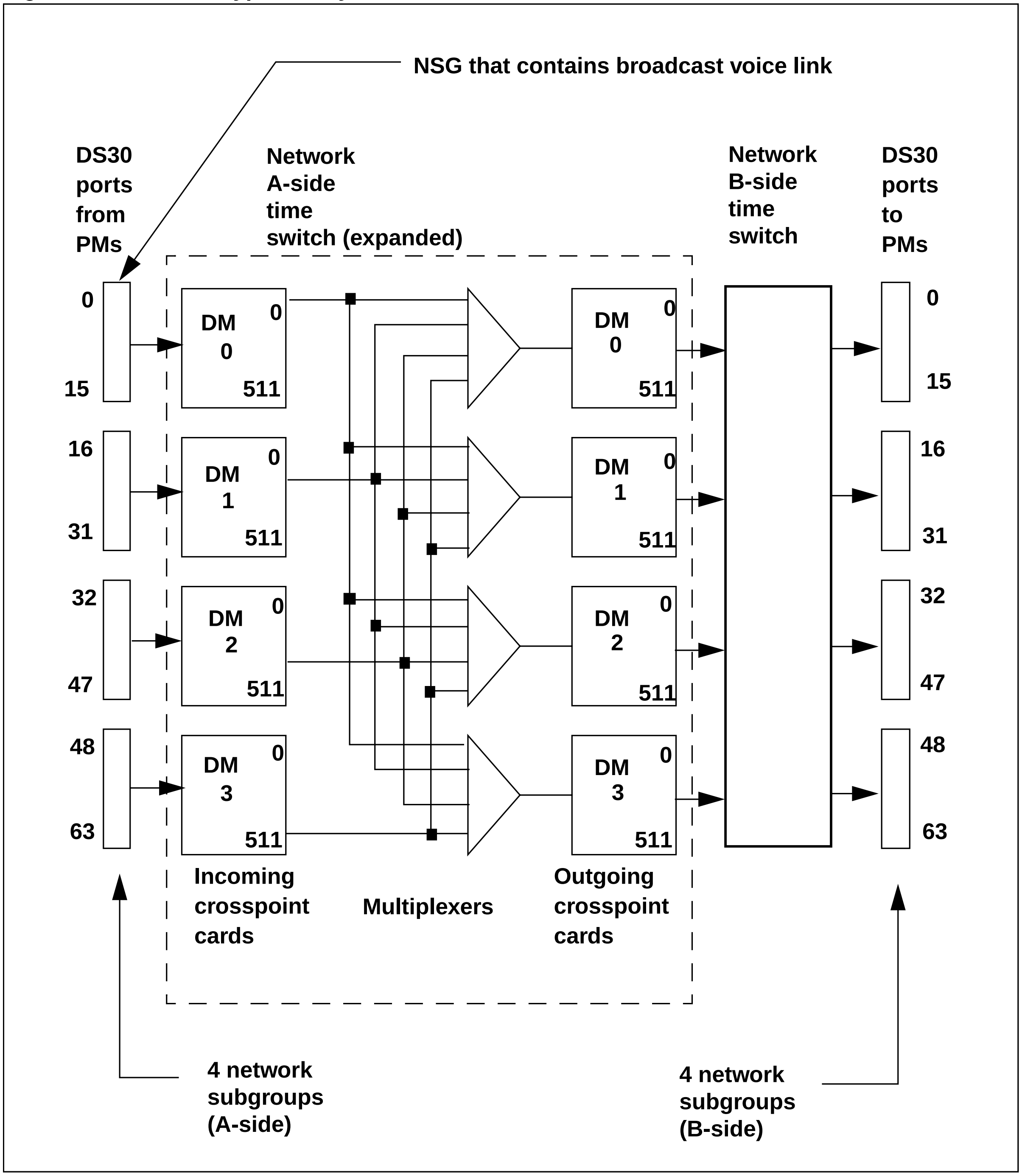
- an A-side NM, to which the from pathend terminal is connected
- a B-side NM, to which the to pathend terminal is connected
- a hardwired connection between the two NMs (through the junctors)

Each NM side also contains a time switch. On the A-side, the time switch maps the incoming port and channel of the from pathend to an outgoing port and channel (the junctor) to the B-side NM. On the B-side, another time switch maps this incoming junctor port and channel to the outgoing port and channel of the to pathend. When a terminal is broadcasting, it becomes the from pathend of multiple network connections.

Time switching

Each terminal (such as an OSSAIN voice link) on a PM is associated with 1 of 32 channels on a particular DS30 port in an NSG. A fully provisioned NSG has either 8 or 16 ports (depending on the network type), for a total of 256 or 512 PM channels. Figure 331 shows an example of the association between NSGs and PMs in an NT5X13-type of network.

Figure 331 NT5X13-type NM layout



The figure shows two NT5X13-type JNETs (an A-side and a B-side network). A total of 64 P-side ports are on each NM. Ports 0 to 15 make up the first NSG, ports 16 to 31 the second, ports 32 to 47 the third, and ports 48 to 63 the fourth.

Note: In a network connection, the A-side and B-side can be on the same or different NM.

In the figure, the broadcast voice link enters the NM as one channel on a DS30 port on the A-side NM. The path taken by this channel to the terminating end (a TOPS trunk on the B-side DS30 PM interface) is shaded.

Each NSG interfaces to a particular incoming crosspoint card. This card, along with an outgoing crosspoint card, two sections of data memory (DM) and four multiplexers, make up the time switch. Each DM holds 512 10-bit PCM speech samples, which is one sample from each of the 512 terminals on the NSG.

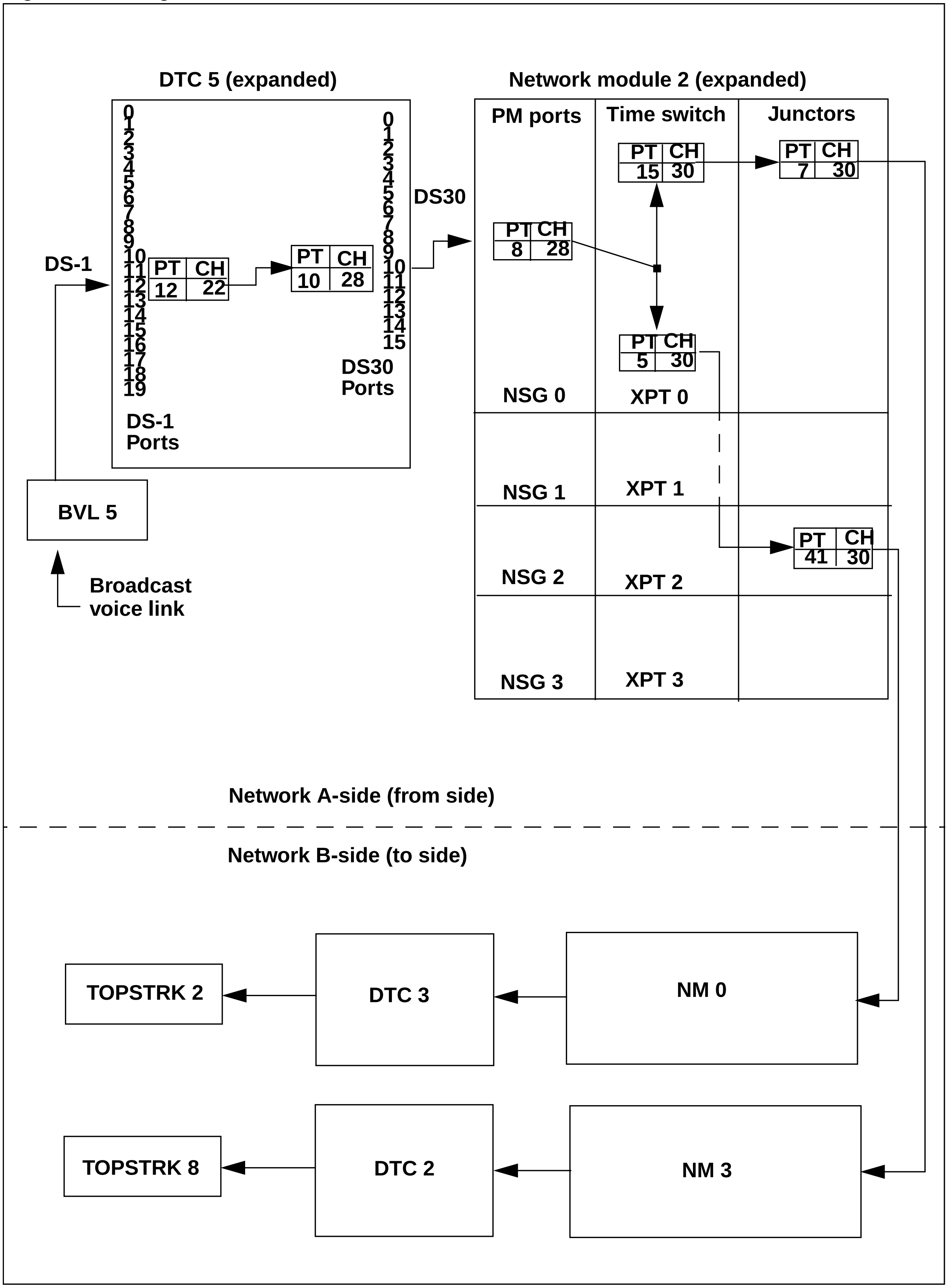
As speech samples arrive, they are written into the incoming DM. At the same time, samples are read from this DM to any of the four outgoing DMs. Data sent from the computing module (CM) when the connection is first made determines where a particular speech sample is written into the outgoing DM. This data is used to control the multiplexers.

The transfer of PCM speech samples from one location and time slot in memory to another accomplishes the time-change necessary to switch the 512 incoming channels to *any* of the 2048 outgoing channels. Each transfer of one speech sample from an incoming DM to an outgoing DM occurs over a bus between the two DMs, and is identified by a particular port and channel.

Channel mapping in a broadcast network connection

Figure 332 shows the channel mapping that occurs when a voice link and two TOPS trunks are involved in a broadcast network connection. (Only the A-side channel mapping is shown; the B-side mapping is similar).

Figure 332 Single broadcast voice link connected to two TOPS trunks



A single broadcast voice link is one channel of a DS30 port on the network A-side (from side). This port shares an NSG and 512 time switch bus channels, with up to 15 other ports coming from the same or different PMs. Likewise, the TOPS trunks each use a single channel on separate DS30 ports on the network B-side (to side).

To complete the connection to TOPSTRK 2, the call must obtain one of the 512 bus channels that transfers the voice links' speech samples out of crosspoint card 0 to another crosspoint on the A-side. The CM determines which card to transfer to (that is, which bus port and channel to use). In the figure, bus channels are shown as the port and channel assignments in the time switch. If a call cannot obtain one of the 512 channels, network blocking occurs.

Two bus channels are required in the A-side time switch, because a bus port and channel are also needed to connect the voice link to TOPSTRK 8 in the second call. These two channels are obtained from a pool of 512 available to all calls on that combination NSG and time switch. Because each broadcast connection requires one of these channels, fewer channels are available for other, *non*-broadcast calls. So when an NM has broadcasting terminals on it, port deloading of one or more network P-side ports may be necessary. The next sections describe network blocking and port deloading.

Network blocking

On the NT5X13 or NT8X11 type of network, 512 PM channels are associated with an NSG (16 ports at 32 channels each), along with 512 time switch bus channels. These channels are shared among all terminals connected to the 16 ports.

Only 30 channels of a DS30 port—channels 1 to 15 and 17 to 31—are used for PCM speech from a terminal on a PM. Channel 0 is reserved for messaging between the central control (CC) and the PM, and channel 16 is reserved for maintenance. So on an NSG, there are at most 480 terminals (16 ports * 30 channels = 480 terminals).

On a network *without* broadcast voice links, there is more than a one-to-one correspondence between PM terminals on an NSG and available crosspoint bus channels (480 terminals, 512 channels). A JNET is non-blocking if it has the same or fewer number of terminals than crosspoint bus channels.

Effect of broadcast voice links on NM capacity

Consider a single voice link with a MAXCONNS value (in table OAVLMAP) of 255. When this number of connections exist simultaneously, a single broadcast voice link uses 255 A-side outgoing crosspoint channels. In such a scenario, only 257 channels are available for other terminals associated with the NSG.

Therefore, one broadcast voice link can use the equivalent channel capacity available for *eight* NM P-side ports. Because there is no longer a channel for every terminal on the NSG, network blocking is possible. Once the 512 channels are used, no further connections can be made to any terminal associated with that crosspoint card and NSG.

Port deloading

To avoid contention for crosspoint channels, one or more NM P-side ports should be left unused, or *deloaded*. For more information, refer to "Determining the number of ports to deload."

Special provisioning cases

Port deloading is *not* necessary in the following special cases:

- NMs of the type NT5X13 or NT8X11 have 32 extra channels on an NSG (512 channels - 480 terminals = 32). So, up to 128 broadcast connections (4 NSGs * 32 channels = 128) can be made to one NM without port deloading.

  Note: Any number of trunks can be assigned as broadcast voice links with MAXCONNS values of 32, if they all terminate to *different* NSGs.

- NMs of the type NT0X48 have 16 extra channels on an NSG. So, up to 64 broadcast connections (4 NSGs * 16 channels = 64) can be made to one NM without port deloading.

Refer to Table 199, Table 200, and Table 201, beginning on page 615, to help map a voice link from the PM P-side to its C-side. Ensure that any broadcast voice links assigned without deloading terminate to different NSGs.

Determining the number of ports to deload

This section discusses JNETs that require port deloading to provision broadcast voice links. The following steps determine the number of NM P-side ports to deload on each NSG:

1. Identify a PM on which broadcast voice links will reside.
2. Select a single C-side port on the PM to be known as the network broadcast port (NBP). This C-side port connects to the NSG. It should be the *only* NBP on that NSG.
3. Based on the PM type, determine the 30 P-side ports and channels that map to the NBP. For a DTC or LTC, use the column from Table 199 that corresponds to the C-side port selected in step 2. Any of the 30 DS-1 ports in this column may be assigned as broadcast voice links.
4. Add up the MAXCONNS values (from table OAVLMAP) of each broadcast voice link. This is the total number of broadcast connections that can occur at any one time on this NBP. Ensure that it does not exceed 512 (or 256 for an NT0X48 network).

   Note: Details on table OAVLMAP are in Chapter 7: "OSSAIN data schema."

5 Based on the number of broadcast connections in step 4, use Table 197 (or Table 198 for NT0X48 networks) to determine the number of ports to deload (unassign) to provide non-blocking on the network.

Note: If more broadcast voice links are required, or if voice links are to be spread out over more PMs, repeat steps 1 through 5.

Table 197 maps the maximum number of broadcast connections to the number of P-side NM ports to deload on an NSG in an NT5X13 or NT8X11 type of network.

Table 197 Port deloading for NT5X13 and NT8X11 networks

Total MAXCONNS value on NSG	Maximum number of terminals supported by NSG	Number of ports to deload on NSG
0 - 32	480	0
33 - 62	450	1
63 - 92	420	2
93 - 122	390	3
123 - 152	360	4
153 - 182	330	5
183 - 212	300	6
213 - 242	270	7
243 - 272	240	8
273 -302	210	9
303 - 332	180	10
333 - 362	150	11
363 - 392	120	12
393 - 422	90	13
423 - 452	60	14
453 - 482	30	15
483 - 512	1 - 29 (see Note)	15

Note: In this situation, the number of terminals assigned to this single port depends on the number of broadcast connections required. For example, if 2 channels on this port are voice links that have a MAXCONNS value of 250 each (for a total of 500), then only 14 of the 30 terminals can be assigned to this port (2 broadcast trunks plus 12 non-broadcast terminals). The total number of connections (broadcast and non-broadcast) should not exceed 512.

Table 198 maps the maximum number of broadcast connections to the number of P-side NM ports to deload on an NSG in an NT0X48 type of network.

Table 198 Port deloading for NT0X48 networks

Total MAXCONNS value on NSG	Maximum number of terminals supported by NSG	Number of ports to deload on NSG
0 - 16	240	0
17 - 46	210	1
47 - 76	180	2
77 - 106	150	3
107 - 136	120	4
137 - 166	90	5
167 - 196	60	6
197 - 226	30	7
227 - 256	30 (see Note)	7

Note: In this situation, the number of terminals assigned to this single port depends on the number of broadcast connections required. For example, if 3 channels on this port are voice links that have a MAXCONNS value of 80 each (for a total of 240), then only 19 of the 30 terminals can be assigned to this port (3 broadcast trunks plus 16 non-broadcast terminals). The total number of connections (broadcast and non-broadcast) should not exceed 256.

Provisioning rules

The following provisioning rules affect broadcast voice links in JNET offices:

- Up to 32 (16 for NT0X48) broadcast connections per NSG are possible *without* port deloading.
- No more than one NBP should be on an NSG.
- Do not exceed 512 (256 for NT0X48) broadcast connections on a single NSG.
- All broadcast voice links on a PM should map to an NBP.
- A PM can be connected to more than one NBP as long as the NBPs are on different NSGs.
- An NSG with a port assigned to a digital recorded announcement machine (DRAM) that broadcasts announcements should not be used as the NBP. In effect, this NSG already has a NBP on it.
- A MAXCONNS value greater than 255 should not be assigned to a single broadcast voice link. (This restriction is enforced by DMS table control.)

- The provisioning rules do not apply to an OSAC remote switch, because broadcasting only takes place in the OSAC host switch.
- To limit the possibility of lost messages during a DS-1 failure, assign only one broadcast voice link on a DS-1. If more than one broadcast voice link is assigned on a DS-1, keep the total number of all broadcast connections on the DS-1 to less than 255.

PM port and channel mapping

This section discusses port and channel mapping for the following PMs:

- digital trunk controller/line trunk controller (DTC/LTC)
- digital carrier module (DCM)
- trunk module 8 (TM8)

DTC/LTC

A DTC or LTC can have 20 P-side ports that map to 16 C-side ports. For every DS-1 channel *n*, DS-1 channel (*n* + 4) maps to the same C-side port. Table 199 shows the resulting P-side DS-1 port and channel for a given C-side port. This is the port that is connected to the NM and makes up 1 of the 16 possible ports on an NSG.

When a DTC or LTC is used for broadcast voice links, select one of the 16 C-side ports as the NBP.

Note: DS30A ports on an LTC allocate C-side channels dynamically, whereas the DS-1 channels are nailed up to a particular C-side port and channel. DS30A channel allocation does not affect DS-1 mapping.

Table 199 Mapping of C-side ports (cols) and C-side channels (rows) to DS-1 ports and channels

C-side chnl	C-side port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	DS1port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	2	3	3	3	2	3	3	3	2	3	3	3	2	3	3	3
4	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	3	3	4	4	3	3	4	4	3	3	4	4	3	3	4	4

Table 199 Mapping of C-side ports (cols) and C-side channels (rows) to DS-1 ports and channels

C-side chnl	C-side port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	4	4	4	5	4	4	4	5	4	4	4	5	4	4	4	5
6	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	DS1 port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	6	7	7	7	6	7	7	7	6	7	7	7	6	7	7	7
9	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	7	7	8	8	7	7	8	8	7	7	8	8	7	7	8	8
10	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	8	8	8	9	8	8	8	9	8	8	8	9	8	8	8	9
11	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
12	DS1 port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
13	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	10	11	11	11	10	11	11	11	10	11	11	11	10	11	11	11
14	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	11	11	12	12	11	11	12	12	11	11	12	12	11	11	12	12
15	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	12	12	12	13	12	12	12	13	12	12	12	13	12	12	12	13
16		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
18	DS1 port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
19	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	14	15	15	15	14	15	15	15	14	15	15	15	14	15	15	15

Table 199 Mapping of C-side ports (cols) and C-side channels (rows) to DS-1 ports and channels

C-side chnl	C-side port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
20	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	15	15	16	16	15	15	16	16	15	15	16	16	15	15	16	16
21	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	16	16	16	17	16	16	16	17	16	16	16	17	16	16	16	17
22	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
23	DS1 port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
24	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	18	19	19	19	18	19	19	19	18	19	19	19	18	19	19	19
25	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	19	19	20	20	19	19	20	20	19	19	20	20	19	19	20	20
26	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	20	20	20	21	20	20	20	21	20	20	20	21	20	20	20	21
27	DS1 port	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
	DS1 chnl	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
28	DS1 port	0	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18
	DS1 chnl	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
29	DS1 port	4	0	1	2	9	5	6	7	14	10	11	12	19	15	16	17
	DS1 chnl	22	23	23	23	22	23	23	23	22	23	23	23	22	23	23	23
30	DS1 port	3	4	0	1	8	9	5	6	13	14	10	11	18	19	15	16
	DS1 chnl	23	23	24	24	23	23	24	24	23	23	24	24	23	23	24	24
31	DS1 port	2	3	4	0	7	8	9	5	12	13	14	10	17	18	19	15
	DS1 chnl	24	24	24	1	24	24	24	1	24	24	24	1	24	24	24	1

Table 200 is the reverse of Table 199; it shows the resulting C-side port for a given P-side DS-1 port and channel.

Table 200 Mapping of DS-1 ports (cols) and channels (rows) to a C-side port for DTC and LTC

DS1 chnl	DS1 port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
2	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
3	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
4	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14
5	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
6	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
7	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
8	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14
9	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
10	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
11	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
12	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14

Table 200 Mapping of DS-1 ports (cols) and channels (rows) to a C-side port for DTC and LTC

DS1 chnl	DS1 port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
13	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
14	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
15	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
16	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14
17	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
18	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
19	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
20	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14
21	C-side port	3	0	1	2	3	7	4	5	6	7	11	8	9	10	11	15	12	13	14	15
22	C-side port	0	1	2	3	0	4	5	6	7	4	8	9	10	11	8	12	13	14	15	12
23	C-side port	1	2	3	0	1	5	6	7	4	5	9	10	11	8	9	13	14	15	12	13
24	C-side port	2	3	0	1	2	6	7	4	5	6	10	11	8	9	10	14	15	12	13	14

DCM

A DCM can have five P-side DS-1 ports that map to four C-side ports. For every DS-1 channel *n*, DS-1 channel (*n* + 4) maps to the same C-side port. Table 201 shows the relationship between the C-side port assignment and DS-1 channel.

Table 201 DCM port and channel mapping

DCM C-side port	DCM DS-1 channel
0	1, 5, 9, 13, 17, 21
1	2, 6, 10, 14, 18, 22
2	3, 7, 11, 15, 19, 23
3	4, 8, 12, 16, 20, 24

The DS-1 port number is not important, because a channel always maps to the same C-side port regardless of which DS-1 port it is assigned to. For example, DS-1 channel 1 on each port maps to C-side port 0; DS-1 channel 12 on each port maps to C-side port 3, and so on.

When a DCM is used for broadcast voice links, select one of the four C-side ports as the NBP.

TM8

Because a TM8 has only a single C-side port, the mapping of a P-side port and channel to a C-side port and channel is straight forward. *All* ports assigned to the TM8 appear on the C-side port to the NM. When a TM8 is used for broadcast voice links, its single C-side port is the NBP.

Provisioning examples

This section provides examples of provisioning scenarios using the port deloading and mapping tables, Table 197 through Table 201. The first two examples assume that the C-side ports of the DTC have been assigned in table LTCINV (Line Trunk Controller Inventory).

Note: For details on table LTCINV, please refer to the *Customer Data Schema Reference Manual*.

Example 1

1 DTC 0 is to be assigned five broadcast voice links. Each link has a MAXCONNS value of 30.

2 DTC 0 C-side port 10 is selected as the NBP. This port is connected to an NSG that does not have any other NBPs (or DRAMs with broadcast announcements) assigned to it. The type of NM is NT5X13.

3 Using the column for C-side port 10 in Table 199, the following DS-1 ports and channels are selected as the five broadcast voice links:

— port 13 channel 1

— port 13 channel 5

— port 13 channel 9

— port 14 channel 12

— port 14 channel 16

4 The total of all MAXCONNS values for the ports is 150, which is acceptable because it does not exceed 512.

5 In Table 197, the 150 broadcast connections fall in the range of 123 to 152 (column 1). For this range, the number of P-side NM ports to be deloaded is four (column 3). So, four ports on the NSG with the NBP need to be unassigned in switch datafill.

Example 2

1 DTC 1 is to be assigned 10 broadcast voice links. Each link has a MAXCONNS value of 30. In this example, the 32 extra bus channels on an NSG can be used to assign broadcast voice links without port deloading.

Note: Without port deloading, DTC 1 must have 10 of its C-side ports assigned to different NSGs. If they are not, the broadcast links must be spread over two or more PMs. And because 10 NSGs are required, the office must be equipped with a minimum of three NMs (NT5X13 or NT8X11).

2 Select 10 P-side ports on DTC 1 for the broadcast voice links. Using Table 200, map the DS-1 port and channel to its C-side port and channel. Ensure that each C-side port and channel maps to a different NSG (same or different NM) on either an NT5X13 or NT8X11 network. If they do not, select another P-side port and channel until the 10 map to 10 different NSGs. Also ensure that these NSGs are not already assigned to broadcast ports (either DRAMs or other broadcast voice links).

Example 3

1 DCM 0 is to be assigned one broadcast voice link. The link has a MAXCONNS value of 255.

2 DCM 0 C-side port 3 is selected as the NBP. This port is connected to an NSG that does not have any other NBPs (or DRAMs with broadcast announcements) assigned to it. The type of NM is NT8X11.

3 Using the column for C-side port 3 in Table 201 (the P-side port number can be from 0 to 5), the DS-1 port and channel selected for the broadcast voice link is port 0 channel 4

4 The total of all MAXCONNS values is 255, which is acceptable because it does not exceed 512.

5 In Table 197, the 255 broadcast connections fall in the range of 243 to 272 (column 1). For this range, the number of P-side NM ports to be deloaded is eight (column 3). So, eight ports on the NSG with the NBP need to be unassigned in switch datafill.

Appendix D: OSSAIN datafill example for Billing and Access Services

Nortel’s Billing and Access Services is a software audio application that runs on the Intelligent Services Node (ISN) Network Application Vehicle (NAV) hardware platform. Billing and Access Services software supports the following automated calling services:

- 0- automation
- 0+ automation
- 1-800 access
- billing acceptance (BAC)

Appendix D: 'OSSAIN datafill example for Billing and Access Services' focuses on the TOPS OSSAIN *switch datafill* required to provision the Billing and Access Services application. It shows examples of both standalone OSSAIN and centralized OSSAIN (OSAC) network configurations and the corresponding switch datafill needed for Billing and Access Services.

Note 1: ISN, NAV, and Billing and Access Services are proprietary products of Nortel. For more information on these products, contact Nortel Network Information Services Marketing.

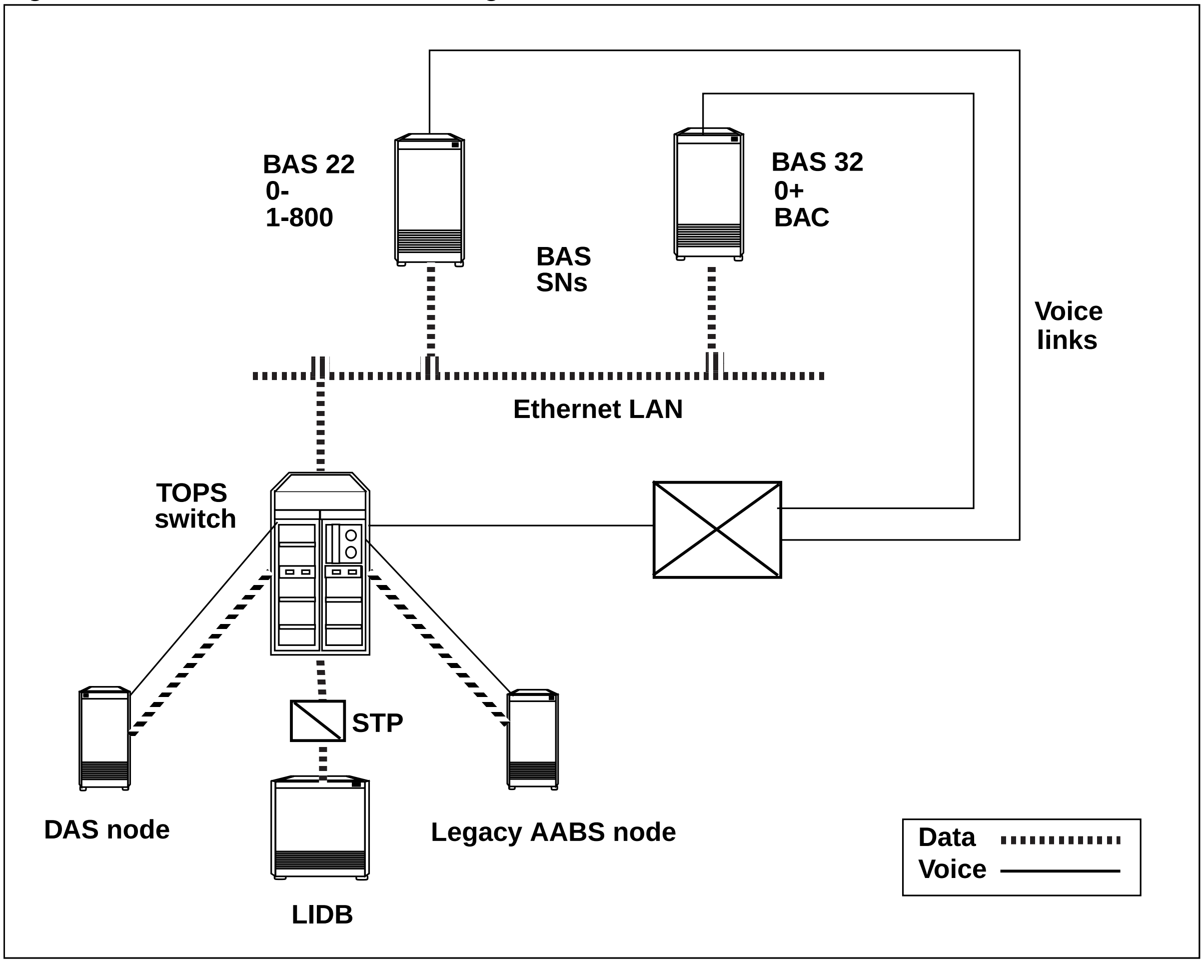
Note 2: The Billing and Access Services application also requires configuration datafill at the service node—the ISN NAV. This datafill must be coordinated with the switch datafill. This document *does not* describe how to provision Billing and Access Services on the NAV.

Example standalone OSSAIN configuration

The following figure shows an example of a standalone OSSAIN configuration with two service nodes (SN) that provide Billing and Access Services (abbreviated in datafill as “BAS”). In the figure, SN BAS 22 provides 0- automation and 1-800 services, while SN BAS 32 provides 0+ automation and BAC services. An Ethernet LAN provides data communication among the switch and SNs.

Other network components shown in the figure include a directory assistance system (DAS) for DA service, a line information database (LIDB) for LIDB queries, and an AABS node for legacy automated billing acceptance service.

Figure 333 Standalone OSSAIN configuration with BAS SNs



Note 1: For simplicity, operator positions are not shown.

Note 2: Refer to "Example OSAC configuration" on page 637 for an example of an OSAC network.

Assumptions

This appendix does not provide details on standard TOPS Queue Management System (QMS) processing. Readers must be familiar with the following QMS functions:

- How call type for queuing (CT4Q) refinements are applied to route the call to TOPS or OSSAIN
- How an individual operator is selected

For complete information on QMS, refer to the Advanced Queuing section of the *Translations Guide*.

Example datafill for Billing and Access Services

The examples in this section show how to datafill the Billing and Access Services application at the TOPS switch. The following areas of OSSAIN processing require datafill:

- QMS queuing
- nodes (service nodes and switches)
- session pools
- functions and control lists
- triggers
- voice links
- parameters

The following table lists the switch data tables in the order in which they are to be datafilled. Beginning on page 626 is a brief explanation and example of the Billing and Access Services datafill for each table.

Note 1: Refer to page 638 for details on OSAC-specific datafill.

Note 2: For details on all the fields in the OSSAIN tables, refer to Chapter 7: "OSSAIN data schema." For details on all the fields in the QMS tables, refer to the *Customer Data Schema Reference Manual*.

Table 202 Datafill sequence for setting up Billing and Access Services

Table name	Description
QAPLNDEF	Defines call queues and agents for OSSAIN QMS application
CT4QNAMS	Defines OSSAIN QMS call type for queuing (CT4Q) names and trigger profile index
TQCQINFO	Associates TOPS call queues with service names
QMSCQDEF	Defines OSSAIN QMS call queues (standalone/host switch only)
TQMSFCQA	Specifies the final call queue for each TOPS operator CT4Q name
OANODNAM	Specifies the identifiers and names of OSSAIN nodes (SNs and switches)
OANODINV	Provides node information for OSSAIN nodes (OSAC, OSNM, and OSN)
OQCQPROF	Specifies a call queue profile number for a list of OSSAIN QMS call queues (standalone/host switch only)
OASESNPL	Defines OSSAIN session pools
OAFUNDEF	Defines OSSAIN functions
OACTLDEF	Defines OSSAIN control lists
OADTFPRF	Specifies DTMF digit trigger profiles
OACNNPRF	Specifies connecting state trigger profiles

Table 202 Datafill sequence for setting up Billing and Access Services

Table name	Description
OADSCPRF	Specifies disconnecting state trigger profiles
OATPRFIX	Specifies an overall trigger profile index for the trigger profiles of the connecting, talking, and disconnecting states
OAINCTLA	Associates an OSSAIN CT4Q name with a control list name
TRKGRP	Specifies information on all the trunk groups in the switch
OAVLMAP	Associates the logical voice channel with the physical voice circuit datafilled in table TRKMEM
OSCVLGRP	Defines a voice link group for the voice connection between an OSAC host switch and an OSAC remote switch (host switch only)
OAFNDISP	Specifies actions for deflections, overflows, blocking, and origination failures
OAINPARM	Specifies various parameters for OSSAIN processing
OFCENG	Specifies various parameters, such as recording units, for the TOPS switch ***Note:*** Parameter OSSAIN_NUM_RU is datafilled in both the host switch and the remote switch. Parameter OSAC_NUM_RU is datafilled in the host switch only.
AABSOST	Specifies trunks used for Automated Alternate Billing Service (AABS)

QAPLNDEF

Datafill in table QAPLNDEF defines the call queues and agents for the OSSAIN QMS application. In the following example, the OSSAIN application has its own tuple.

Figure 334 MAP display example for table QAPLNDEF

```
APLNAME    CALLQS    CQELEMS   AGENTQS   NUMAGNTS   CTSEARCH
----------------------------------------------------------------------
TOPS       255       3000      127       150        0
OSSAIN     255       5000      127       8191       0
```

CT4QNAMS

Datafill in table CT4QNAMS defines the names of the CT4Qs. In the following example, these CT4Qs are defined:

- 724 BAS_0_PLUS_OSSAIN is for 0+ calls to BAS
- 725 BAS_0_MINUS_OSSAIN is for 0- calls to BAS
- 726 BAS_1_PLUS_OSSAIN is for 1+800 calls to BAS
- 727 BAS_NODE_TO_OPR is for BAS 0+TA transfer to operator calls
- 728 BAS_BAC_OSSAIN is for calls transferred to BAS for billing acceptance (BAC)

- 729 BAS_DA_OSSAIN_411 is for BAS 0+DA 411 calls
- 730 BAS_DA_NODE_TO_OPR is for BAS 0+DA transfer to operator calls (billing satisfied)
- 731 BAS_CDIR_OSSAIN is for country direct calls
- 732 BAS_DA_OSSAIN_FOR is for BAS 0+DA FOR_555 calls
- 733 BAS_DA_OSSAIN_HOM is for BAS 0+DA HOM_555 calls
- 734 BAS_DA_OSSAIN_555 is for BAS 0+DA DA_555 calls

Figure 335 MAP display example for table CT4QNAMS

```
CT4QNUM   CT4QNAME            NOAMA  ITRIGIDX SYSAREA
----------------------------------------------------------------------
724      BAS_0_PLUS_OSSAIN    N      Y   54   OSSAIN
725      BAS_0_MINUS_OSSAIN   N      Y   54   OSSAIN
726      BAS_1_PLUS_OSSAIN    N      Y   54   OSSAIN
727      BAS_NODE_TO_OPR      N      N        TOPSOPR Y EBAS2OPR N  N  N
728      BAS_AABS_OSSAIN      N      Y   54   OSSAIN
729      BAS_DA_OSSAIN_411    N      Y   54   OSSAIN
730      BAS_DA_NODE_TO_OPR   N      N        TOPSOPR Y EBAS_DA  N  N  N
731      BAS_CDIR_OSSAIN      N      N        OSSAIN
732      BAS_DA_OSSAIN_FOR    N      Y   54   OSSAIN
733      BAS_DA_OSSAIN_HOM    N      Y   54   OSSAIN
734      BAS_DA_OSSAIN_555    N      Y   54   OSSAIN
```

Datafill in table CT4QNAMS in field ITRIGIDX associates an OSSAIN CT4Q name with the call trigger profile index.

TQCQINFO

Datafill in table TQCQINFO identifies service information for the TOPS call queues. In the following example, these queues represent the final operator call queue assignments for the functions whose FUNCTYPE field in table OAFUNDEF is TOPSOPER.

Note: These queues also may be used for standard (non-OSSAIN) TOPS calls.

Figure 336 MAP display example for table TQCQINFO

```
CALLQ  QMSSERV   CWOFF CWON TREAT
------------------------------------
CQ228  TOPS_TA   10    20   EMR4
CQ229  TOPS_TA   10    20   EMR4
CQ232  TOPS_DA   10    20   EMR5
CQ233  TOPS_DA   10    20   EMR5
CQ236  TOPS_TA   10    20   EMR4
CQ237  TOPS_DA   10    20   EMR5
CQ245  TOPS_DA   10    20   EMR5
CQ246  TOPS_DA   10    20   EMR5
CQ249  TOPS_DA   10    20   EMR5
```

QMSCQDEF

Datafill in table QMSCQDEF defines characteristics of the call queues required by each application. In the following example, OSSAIN call queues are used when routing a call to a function whose FUNCTYPE field in table OAFUNDEF is SN. Likewise, TOPS call queues are used when routing a call to a function whose FUNCTYPE is TOPSOPER. (TOPS call queues also may be used for standard (non-OSSAIN) TOPS calls.

Figure 337 MAP display example for table QMSCQDEF

```
APPLNCQ      CQPRIO CQAGS CQMAXSIZ  DEFLAREA   PRAQAREA
----------------------------------------------------------
OSSAIN 32    0      10     300      Y 30 0     N
OSSAIN 33    0      10     300      Y 30 0     N
OSSAIN 34    0      10     300      Y 30 0     N
OSSAIN 36    0      10     300      Y 30 0     N
OSSAIN 38    0      10     300      Y 30 0     N
OSSAIN 60    0      10     300      Y 30 0     N
OSSAIN 61    0      10     300      Y 30 0     N
OSSAIN 62    0      10     300      Y 30 0     N
OSSAIN 63    0      10     300      Y 30 0     N
TOPS  228    0      10     300      N          N
TOPS  229    0      10     300      N          N
TOPS  232    0      10     300      N          N
TOPS  233    0      10     300      N          N
TOPS  236    0      10     300      N          N
TOPS  237    0      10     300      N          N
TOPS  245    0      10     300      N          N
TOPS  246    0      10     300      N          N
TOPS  249    0      10     300      N          N
```

TQMSFCQA

Datafill in table TQMSFCQA associates a final QMS CT4Q with a TOPS operator call queue. In the following example, only the CT4Qs from field OPRCT4Q in table OAFUNDEF are shown. These CT4Qs are used when routing a call to a function whose FUNCTYPE field is TOPSOPER.

Figure 338 MAP display example for table TQMSFCQA

```
CT4Q                CALLQ  RECALLQ  ASSTAREA
-----------------------------------------------
BAS_NODE_TO_OPR     CQ236  CQ236    CSE CQ250
BAS_DA_NODE_TO_OPR  CQ237  CQ237    CSE CQ251
BAS_DA_NEED_CASH_Q  CQ232  CQ232    CSE CQ251
```

OANODNAM

Datafill in table OANODNAM specifies the name of the nodes in the OSSAIN network. In the following example, two BAS SNs are defined: BAS_22 (0- and 1-800 services) and BAS_32 (0+ and BAC).

Figure 339 MAP display example for table OANODNAM

```
NODEID   NODENAME
-----------------------
11       BAS_22
15       BAS_32
```

OANODINV

Datafill in table OANODINV specifies node information for all nodes in the OSSAIN network. In the following example, the two BAS SNs have a node type of OSNM. The IP address for BAS_22 is 47.67.17.22; the IP address for BAS_32 is 47.67.17.32.

Figure 340 MAP display example for table OANODINV

```
NODENAME  NODEAREA
                         AUDTRIES AUDRTIMR AUDFREQ
----------------------------------------------------------------------
BAS_22   OSNM 11 UDP IPV4 47 67 17 22 8000 EIU Y 2 30 30 OAIN 2 C 11
      BAS_SN SHELF_1        4        6        20
BAS_32   OSNM 15 UDP IPV4 47 67 17 32 8000 EIU Y 2 30 30 OAIN 2 C 11
      BAS_SN  SHELF_5       4        6        20
```

Note 1: The descriptive name for the node type (OAIN) comes from datafill in table ENTYPES. The descriptive name for the node site (BAS_SN) comes from datafill in table ENSITES. The descriptive information (SHELF_5) appears at the maintenance MAP when the node is posted.

Note 2: The specific values of AUDTRIES, AUDTIMR, and AUDFREQ depend upon the applications in use. For more information, please refer to the *Billing and Access Services Administration Guide*, 203-3261-301, or the *Billing and Access Services Planning and Engineering Guide*, 203-3261-101

OQCQPROF

Datafill in table OQCQPROF associates lists of OSSAIN call queues with a profile number. This profile number is used by the session pool in table OASESNPL. In the following example, two call queue profile numbers (10 and 12) are datafilled.

Figure 341 MAP display example for table OQCQPROF

```
CQPROFNM       AGENTQ     PRIOAREA
--------------------------------------------------------------
10             AQ10       OFC CQ32 CQ33 CQ38 CQ60 $
12             AQ12       OFC CQ34 CQ36 CQ61 CQ62 CQ63 $
```

OASESNPL

Datafill in table OASESNPL defines the session pools on the BAS SNs. In the following example, two BAS SN session pools (BAS22_1 and BAS32_1) are datafilled, both of which are used for subscriber originations.

Figure 342 MAP display example for table OASESNPL

```
SESNPLID     SESNPLNM     MAXSESN     NODENAME
         ORIGAREA
--------------------------------------------------------------
30           BAS22_1      48          BAS_22
         SUBSCRIBER S USEDEFLT USEDEFLT N 10 UDP 8000 8000
34           BAS32_1      48          BAS_32
         SUBSCRIBER S USEDEFLT USEDEFLT N 12 UDP 8000 8000
```

OAFUNDEF

Datafill in table OAFUNDEF defines the OSSAIN functions. For SN functions, the QMS CAMHERE subfield is set to Y (at the standalone/host switch) and the call queue is identified. For operator functions, the operator CT4Q is datafilled. For automated system functions, the operator backup CT4Q is datafilled.

In the following example, 14 functions are defined:

- nine SN functions (SN)
- three operator functions (TOPSOPER)
- two legacy AABS functions (TOPSAUTO)

Figure 343 MAP display example for table OAFUNDEF

```
FUNCID  FUNCNAME          FUNCAREA
------------------------------------------------------------
50      BAS_0_MINUS_F     SN TASERV N N Y 7 N Y Y CQ32 N
51      BAS_1800_F        SN TASERV N N Y 7 N Y Y CQ33 N
52      BAS_BAC_F         SN TASERV N N Y 7 N Y Y CQ34 N
54      BAS_BAC_SEQ_F     SN TASERV N N Y 7 N Y Y CQ36 N
56      BAS_1800_SEQ_F    SN TASERV N N Y 7 N Y Y CQ38 N
57      BAS_ENG_AGENT_F   TOPSOPER BAS_NODE_TO_OPR N
58      LEGACY_AABS_F     TOPSAUTO AABS BAS_NODE_TO_OPR
60      BAS_0_MIN_SEQ_F   SN TASERV N N Y 7 N Y Y CQ60 N
61      BAS_0_PLUS_F      SN TASERV N N Y 7 N Y Y CQ61 N
62      BAS_0PLUS_SEQ_F   SN TASERV N N Y 7 N Y Y CQ62 N
63      BAS_0_PLUS_DA_F   SN DASERV N N Y 7 N Y Y CQ63 N
65      BAS_DA_AGENT_F    TOPSOPER BAS_DA_NODE_TO_OPR N
67      LEGACY_AABS_DA_F  TOPSAUTO AABS BAS_DA_NODE_TO_OPR
68      BAS_DA_CASH_F     TOPSOPER BAS_DA_NEED_CASH_Q N
```

Note: This example shows separate functions for triggering (sequence) calls. Datafilling separate functions may be useful for administrative purposes; however, sequence calls do not require a separate function. The function that processed the original call also can be used for a sequence call.

OACTLDEF

Datafill in table OACTLDEF associates an OSSAIN function with a control list. In the following example, the functions defined in table OAFUNDEF correspond to a control list name and number.

Figure 344 MAP display example for table OACTLDEF

```
OACTLNU   ACTLNAM           NETWRKID  OAFUNCTS
-------------------------------------------------------------
50        BAS_0_MINUS_C      50        (BAS_0_MINUS_F) $
51        BAS_1800_C         51        (BAS_1800_F) $
52        BAS_BAC_C          52        (BAS_BAC_F) $
57        BAS_ENG_AGENT_C    57        (BAS_ENG_AGENT_F) $
58        LEGACY_AABS_C      58        (LEGACY_AABS_F) $
61        BAS_0_PLUS_C       61        (BAS_0_PLUS_F) $
63        BAS_0_PLUS_DA_C    63        (BAS_0_PLUS_DA_F) $
65        BAS_DA_AGENT_C     65        (BAS_DA_AGENT_F) $
67        LEGACY_AABS_DA_C   67        (LEGACY_AABS_DA_F) $
68        BAS_DA_CASH_C      68        (BAS_DA_CASH_F) $
```

OADTFPRF

Datafill in table OADTFPRF specifies a list of DTMF triggers and their corresponding actions. The action must be either a control list (table OACTLDEF) or a function (table OAFUNDEF). In the following example, the octothorpe digit is defined in five different DTMF trigger profiles. The profile index is also used in tables OACNNPRF, OADSCPRF, and OATPRFIX.

Figure 345 MAP display example for table OADTFPRF

```
DTMFIDX  PROFILE                                    HOLDRCVR
-----------------------------------------------------------------
71       (OCTO Y Y N FUNCTION BAS_0PLUS_SEQ_F) $    Y
73       (OCTO Y Y N FUNCTION BAS_1800_SEQ_F)  $    Y
74       (OCTO Y Y N FUNCTION BAS_0_MIN_SEQ_F) $    Y
75       (OCTO Y Y N FUNCTION BAS_BAC_SEQ_F)   $    Y
76       (OCTO Y Y N FUNCTION BAS_0_PLUS_DA_F) $    Y
```

OACNNPRF

Datafill in table OACNNPRF specifies a list of connecting triggers and their corresponding actions. In the following example, each connecting profile uses a different DTMF profile (table OADTFPRF).

Figure 346 MAP display example for table OACNNPRF

```
CONNIDX    PROFILE      DTMFPRF CAUSEPRF
----------------------------------------------------------------------
81         $            Y 71    N
83         $            Y 73    N
84         $            Y 74    N
85         $            Y 75    N
86         $            Y 76    N
```

OADSCPRF

Datafill in table OADSCPRF specifies a list of disconnecting triggers and their corresponding actions. In the following example, each disconnect profile uses a different DTMF profile (table OADTFPRF).

Figure 347 MAP display example for table OADSCPRF

```
DISCIDX     PROFILE       DTMFPRF
---------------------------------------------------------------------
81          $             Y 71
83          $             Y 73
84          $             Y 74
85          $             Y 75
86          $             Y 76
```

OATPRFIX

Datafill in table OATPRFIX specifies an overall trigger profile index for a type of call. This index (IDX field) indirectly points to a trigger profile in table OADTFPRF (page 632). The index is used in table CT4QNAMS to specify a CT4Q name. The SN can override this index when the call is floated or transferred to an operator.

In the following example, these trigger profile indexes are defined:

- 50 is optionally assigned by the SN to 0- calls
- 51 is optionally assigned by the SN to 0+ calls
- 53 is optionally assigned by the SN to 1-800 calls
- 54 is for all Billing and Access Services call begins

  Note: Trigger profile 54 has no triggers defined, which has the same effect as setting the ITRIGIDX field in table CT2QNAMS to N for the final CT4Q.

- 55 is optionally assigned by the SN to BAC billing acceptance calls
- 56 is optionally assigned by the SN to 0+ DA calls

In the first tuple, the overall index of 50 points to a disconnect profile index of 84 (DPROFIDX field) in table OADSCPRF (page 632). Index 84 points to a DTMFPRF profile of 74 (DTMFPRF field) in table OADTFPRF (page 632). Index 74 has the following profile: OCTO Y Y N FUNCTION BAS_0_MIN_SEQ_F.

Figure 348 MAP display example for table OATPRFIX

```
IDX     CPROFIDX     TPROFIDX     DPROFIDX
---------------------------------------------
50      Y 84         N            Y 84
51      Y 81         N            Y 81
53      Y 83         N            Y 83
54      N            N            N
55      Y 85         N            Y 85
56      Y 86         N            Y 86
```

OAINCTLA

Datafill in table OAINCTLA associates an OSSAIN CT4Q name with a control list. In the following examples, seven OSSAIN control list names are datafilled.

Figure 349 MAP display example for table OAINCTLA

```
CT4Q                     OALISTNM
----------------------------------------------------
BAS_0_MINUS_OSSAIN  BAS_0_MINUS_C
BAS_1_PLUS_OSSAIN   BAS_1800_C
BAS_0_PLUS_OSSAIN   BAS_0_PLUS_C
BAS_DA_OSSAIN_FOR   BAS_0_PLUS_DA_C
BAS_DA_OSSAIN_HOM   BAS_0_PLUS_DA_C
BAS_DA_OSSAIN_555   BAS_0_PLUS_DA_C
BAS_DA_OSSAIN_411   BAS_0_PLUS_DA_C
```

TRKGRP

Datafill in table TRKGRP specifies the trunks used for the SN voice links. In the following example, the voice trunks for the BAS SNs are of the TOPSVL type, with a trunk selection algorithm of most idle (MIDL) and a circuit direction of two-way (2W).

Figure 350 MAP display example for table TRKGRP

```
GRPKEY           GRPINFO
------------------------------------------------------
BAS_SNVL22       TOPSVL  0   NPDGP NCRT    MIDL 2W
BAS_SNVL32       TOPSVL  0   NPDGP NCRT    MIDL 2W
```

OAVLMAP

Datafill in table OAVLMAP maps the logical voice channel to the physical voice circuit datafilled in table TRKMEM. In the following example, 96 BAS SN voice links are datafilled: 48 for BAS_SNVL22 and 48 for BAS_SNVL32.

Figure 351 MAP display example for table OAVLMAP

```
NDANCH             CLLI           EXTRKNUM    BCSTAREA
-------------------------------------------------------
BAS_22    1000    BAS_SNVL22     1           N
   |
   |
BAS_22    1047    BAS_SNVL22     48          N
BAS_32    1000    BAS_SNVL32     1           N
   |
   |
BAS_32    1047    BAS_SNVL32     48          N
```

OAFNDISP

Datafill in table OAFNDISP specifies actions for deflections, overflows, blocking, and origination failures. In the following example, the nine SN functions have disposition routing datafilled.

Note: The FUNCNAME field applies only to SN function types defined in table OAFUNDEF.

Figure 352 MAP display example for table OAFNDISP

```
FUNCNAME        IDFLACTN                     IOVFACTN
          IBLKACTN                     IOFLACTN
                TDFLACTN                     TORFACTN
          TBLKACTN                     TOFLACTN
-----------------------------------------------------------------------
BAS_0_MINUS_F   GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_1800_F      GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_BAC_F       GOTOCTL LEGACY_AABS_C        GOTOCTL LEGACY_AABS_C
          GOTOCTL LEGACY_AABS_C        GOTOCTL LEGACY_AABS_C
                GOTOCTL LEGACY_AABS_C        GOTOCTL LEGACY_AABS_C
          GOTOCTL LEGACY_AABS_C        GOTOCTL LEGACY_AABS_C
BAS_BAC_SEQ_F   GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_1800_SEQ_F  GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_0_MIN_SEQ_F GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_0_PLUS_F    GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_0PLUS_SEQ_F GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
                GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
          GOTOCTL BAS_ENG_AGENT_C      GOTOCTL BAS_ENG_AGENT_C
BAS_0_PLUS_DA_F GOTOCTL LEGACY_AABS_DA_C     GOTOCTL LEGACY_AABS_DA_C
          GOTOCTL LEGACY_AABS_DA_C     GOTOCTL LEGACY_AABS_DA_C
                GOTOCTL LEGACY_AABS_DA_C     GOTOCTL LEGACY_AABS_DA_C
          GOTOCTL LEGACY_AABS_DA_C     GOTOCTL LEGACY_AABS_DA_C
```

OAINPARM

Datafill in table OAINPARM specifies various office-wide parameters for OSSAIN processing. In the following example, all ADACC calls that require alternate billing are sent to the BAS_0_PLUS_DA_C control list (63 in table OACTLDEF). All operator calls that are handed off for billing acceptance are sent to OSSAIN processing using the BAS_BAC_OSSAIN CT4Q. The no-answer timer is set at 30 seconds.

Figure 353 MAP display example for table OAINPARM

```
PARMNAME                    PARMVAL
-----------------------------------------------------
ADACC_ALT_BILL_INDEX        Y 63
ALT_BILL_HANDOFF_METHOD     OSSAIN BAS_BAC_OSSAIN
NOANS_TIMER                 30
```

OFCENG

Datafill in table OFCENG specifies various office parameters, including recording units. In the following example, 96 RUs are specified for OSSAIN call processing traffic.

Figure 354 MAP display example for table OFCENG

```
PARMNAME            PARMVAL
----------------------------------
OSSAIN_NUM_RU       96
```

AABSOST

Datafill in table AABSOST enables legacy AABS service for a given trunk. Calls that are routed to a TOPSAUTO function in table OAFUNDEF must have their trunks datafilled as SERV in table AABSOST to route to the legacy AABS node.

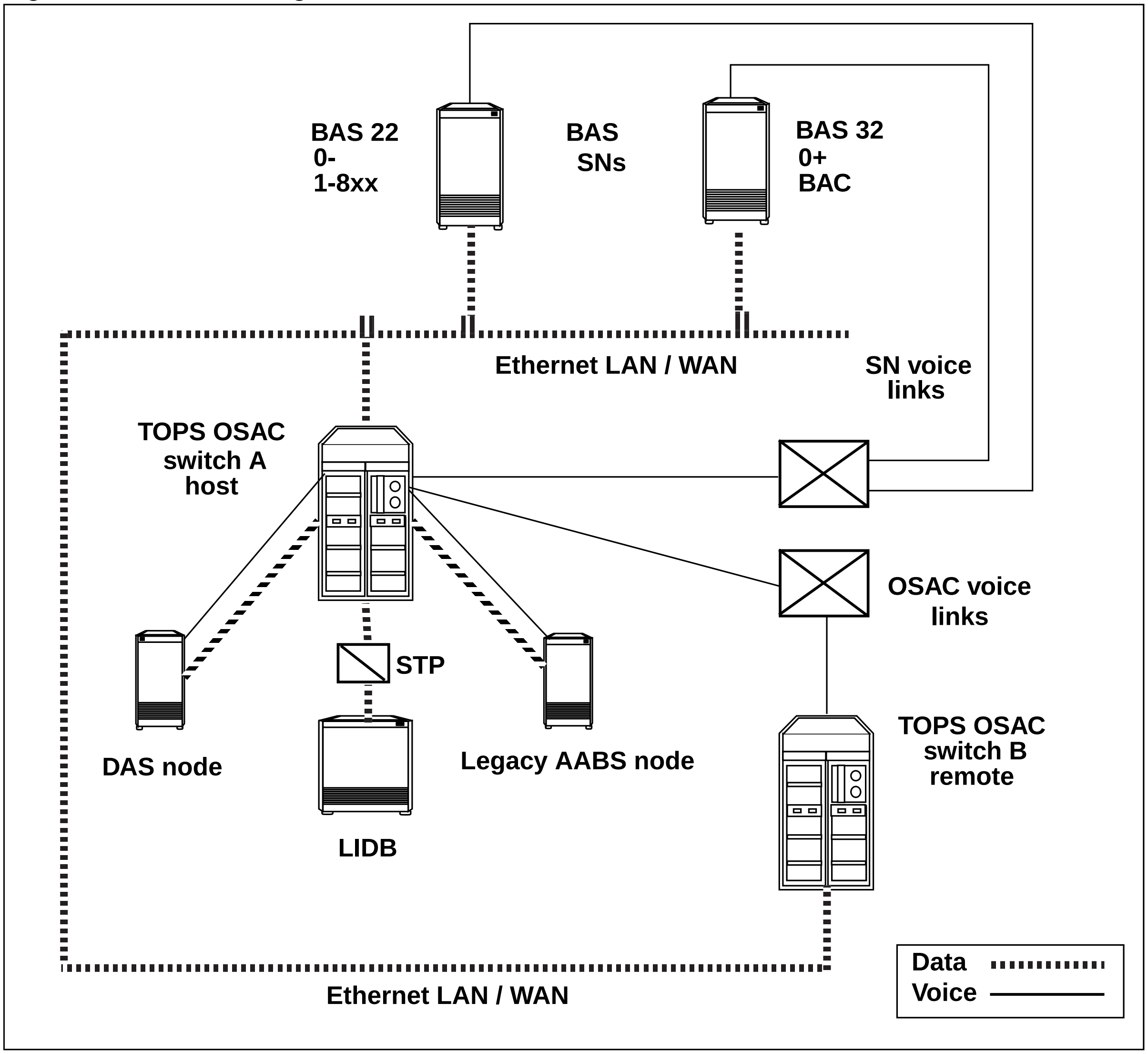
Figure 355 MAP display example for table AABSOST

```
GRPKEY       PUBLIC         PRIVATE
-------------------------------------------
LAKESH_4001  SERV TONEANN   SERV TONEANN
```

Example OSAC configuration

The following figure shows an example of a basic OSAC configuration with two TOPS switches (host, remote) and two centralized BAS SNs.

Figure 356 OSAC configuration with centralized BAS SNs



Example OSAC datafill for Billing and Access Services

In an OSAC configuration, both the OSAC host switch and the OSAC remote switch require datafill. Datafill is needed in all the tables listed on page 625, except where noted. Furthermore, datafill in the following five tables *must be parallel* between the OSAC host switch and OSAC remote switch:

- OANODNAM
- OANODINV
- OASESNPL
- OAFUNDEF
- OAVLMAP

Two other tables require OSAC datafill:

- OSCVLGRP (host-remote voice links datafilled in the OSAC host only)
- OFCENG (OSAC recording units datafilled in the OSAC host only; OSSAIN recording units datafilled in both the OSAC host and remote)

This section provides a brief explanation and example of the OSAC-specific Billing and Access Services datafill in these tables.

OANODNAM

Datafill in table OANODNAM must be parallel between the OSAC host and the OSAC remote. In the following examples, two BAS SNs are defined: BAS_22 provides 0- and 1-800 services; BAS_32 provides 0+ and BAC. And two switches are defined: SWITCHA_HOST (OSAC host) and SWITCHB_REM (OSAC remote).

Figure 357 MAP display example for table OANODNAM (OSAC host)

```
NODEID   NODENAME
---------------------
11       BAS_22
15       BAS_32
30       SWITCHA_HOST
60       SWITCHB_REM
```

Figure 358 MAP display example for table OANODNAM (OSAC remote)

```
NODEID   NODENAME
---------------------
11       BAS_22
15       BAS_32
30       SWITCHA_HOST
60       SWITCHB_REM
```

OANODINV

Datafill in table OANODINV must be parallel between the OSAC host and the OSAC remote. In the following examples, the two switches have a node type of OSAC. SWITCHA_HOST is the host, so its datafill specifies SELF at the host switch and OTHER at the remote switch. Likewise, SWITCHB_REM is a remote, so its datafill specifies OTHER at the host switch and SELF at the remote switch.

The two BAS SNs are centralized at SWITCHA_HOST, so they have a node type of OSNM at the host switch and a node type of OSN at the remote switch.

Figure 359 MAP display example for table OANODINV (OSAC host)

```
NODENAME       NODEAREA                                        AUDTRIES AUDRTIMR AUDFREQ
------------------------------------------------------------------------------------
SWITCHA_HOST  OSAC SELF                                       9        8        5
SWITCHB_REM   OSAC OTHER 60 UDP IPV4 47 245 8 11 EIU OAIN 1 B 11 LOCB REMOTE_B
   9        8        5
BAS_22         OSNM 11 UDP IPV4 47 67 17 22 8000 EIU Y 2 30 30 OAIN 2 C 11 BAS_SN
SHELF_1      8         6        60
BAS_32         OSNM 15 UDP IPV4 47 67 17 32 8000 EIU Y 2 30 30 OAIN 2 C 11 BAS_SN
SHELF_5      8         6        60
```

Figure 360 MAP display example for table OANODINV (OSAC remote)

```
NODENAME       NODEAREA                                        AUDTRIES AUDRTIMR AUDFREQ
------------------------------------------------------------------------------------
SWITCHA_HOST  OSAC OTHER 32 UDP IPV4 47 245 9 1 EIU OAIN 1 B 11 LOCA HOST_A
   9        8        5
SWITCHB_REM   OSAC SELF                                        9        8        5
BAS_22        OSN 11 UDP IPV4 47 67 17 22 EIU SWITCHA_HOST Y 2 OAIN 2 C 11
BAS_SN SHELF_1   8         6        60
BAS_32        OSN 15 UDP IPV4 47 67 17 32 EIU SWITCHA_HOST Y 2 OAIN 2 C 11
BAS_SN SHELF_5   8         6        60
```

OASESNPL

Datafill in table OASESNPL must be parallel between the OSAC host and the OSAC remote. In the following examples, two BAS SN session pools (BAS22_1 and BAS32_1) are datafilled, both of which are used for subscriber originations.

One host-remote session pool is datafilled. At the OSAC host, its name is HOST_TO_REM and it is used to receive messages from the OSAC remote switch (SWITCHB_REM). Its ORIGTYPE is OSACTERM. At the OSAC remote, its name is REM_TO_HOST and it is used to originate messages to the OSAC host switch (SWITCHA_HOST). Its ORIGTYPE is OSACORIG.

Figure 361 MAP display example for table OASESNPL (OSAC host)

```
SESNPLID     SESNPLNM     MAXSESN     NODENAME
         ORIGAREA
-------------------------------------------------------------------------
30           BAS22_1      48          BAS_22
         SUBSCRIBER SH USEDEFLT USEDEFLT N 10 UDP 8000 8000
34           BAS32_1      48          BAS_32
         SUBSCRIBER SH USEDEFLT USEDEFLT N 12 UDP 8000 8000
61           HOST_TO_REM  96          SWITCHB_REM OSACTERM
```

Figure 362 MAP display example for table OASESNPL (OSAC remote)

```
SESNPLID     SESNPLNM     MAXSESN     NODENAME
         ORIGAREA
-------------------------------------------------------------------------
30           BAS22_1      48          BAS_22
         SUBSCRIBER R USEDEFLT USEDEFLT N UDP 8000
34           BAS32_1      48          BAS_32
         SUBSCRIBER R USEDEFLT USEDEFLT N UDP 8000
61           REM_TO_HOST  96         SWITCHA_HOST OSACORIG
```

OAFUNDEF

Datafill in table OAFUNDEF must be parallel between the OSAC host and the OSAC remote. In the following examples, 14 functions are defined: nine are provided by the SN; three by the operator; and two by a legacy AABS system.

For SN functions at the OSAC host switch, the QMS CAMHERE subfield is set to Y and the call queue is identified. For SN functions at the OSAC remote switch, the QMS CAMHERE subfield is set to N and the host-remote session pool is identified.

Figure 363 MAP display example for table OAFUNDEF (OSAC host)

```
FUNCID  FUNCNAME          FUNCAREA
--------------------------------------------------------------------------------
50      BAS_0_MINUS_F     SN TASERV N N Y 7 N Y Y CQ32 N
51      BAS_1800_F        SN TASERV N N Y 7 N Y Y CQ33 N
52      BAS_BAC_F         SN TASERV N N Y 7 N Y Y CQ34 N
54      BAS_BAC_SEQ_F     SN TASERV N N Y 7 N Y Y CQ36 N
56      BAS_1800_SEQ_F    SN TASERV N N Y 7 N Y Y CQ38 N
57      BAS_ENG_AGENT_F   TOPSOPER BAS_NODE_TO_OPR N
58      LEGACY_AABS_F     TOPSAUTO AABS BAS_NODE_TO_OPR
60      BAS_0_MIN_SEQ_F   SN TASERV N N Y 7 N Y Y CQ60 N
61      BAS_0_PLUS_F      SN TASERV N N Y 7 N Y Y CQ61 N
62      BAS_0PLUS_SEQ_F   SN TASERV N N Y 7 N Y Y CQ62 N
63      BAS_0_PLUS_DA_F   SN DASERV N N Y 7 N Y Y CQ63 N
65      BAS_DA_AGENT_F    TOPSOPER BAS_DA_NODE_TO_OPR N
67      LEGACY_AABS_DA_F  TOPSAUTO AABS BAS_DA_NODE_TO_OPR
68      BAS_DA_CASH_F     TOPSOPER BAS_DA_NEED_CASH_Q N
```

Figure 364 MAP display example for table OAFUNDEF (OSAC remote)

```
FUNCID  FUNCNAME          FUNCAREA
--------------------------------------------------------------------------------
50      BAS_0_MINUS_F     SN TASERV N N N N Y N REM_TO_HOST
51      BAS_1800_F        SN TASERV N N N N Y N REM_TO_HOST
52      BAS_BAC_F         SN TASERV N N N N Y N REM_TO_HOST
54      BAS_BAC_SEQ_F     SN TASERV N N N N Y N REM_TO_HOST
56      BAS_1800_SEQ_F    SN TASERV N N N N Y N REM_TO_HOST
57      BAS_ENG_AGENT_F   TOPSOPER BAS_NODE_TO_OPR N
58      LEGACY_AABS_F     TOPSAUTO AABS BAS_NODE_TO_OPR (see Note)
60      BAS_0_MIN_SEQ_F   SN TASERV N N N N Y N REM_TO_HOST
61      BAS_0_PLUS_F      SN TASERV N N N N Y N REM_TO_HOST
62      BAS_0PLUS_SEQ_F   SN TASERV N N N N Y N REM_TO_HOST
63      BAS_0_PLUS_DA_F   SN DASERV N N N N Y N REM_TO_HOST
65      BAS_DA_AGENT_F    TOPSOPER BAS_DA_NODE_TO_OPR N
67      LEGACY_AABS_DA_F  TOPSAUTO AABS BAS_DA_NODE_TO_OPR (see Note)
68      BAS_DA_CASH_F     TOPSOPER BAS_DA_NEED_CASH_Q N
```

Note: For these legacy AABS functions to work, an AABS system must be connected to the OSAC remote switch.

OAVLMAP

Datafill in table OAVLMAP maps the logical voice channel to the physical voice circuit datafilled in table TRKMEM. In the following examples, these voice links are specified:

- 96 host-remote voice links: OG_OSAC_B_VL at the host switch and IC_OSAC_A_VL at the remote switch

  Note: Host-remote voice links in table TRKGRP must be TOPSVL trunks set to OG (outgoing) at the OSAC host switch and IC (incoming) at the OSAC remote switch.

- 96 BAS SN voice links: 48 for BAS_SNVL22 and 48 for BAS_SNVL32 (only at the host switch)

Figure 365 MAP display example for table OAVLMAP (OSAC host)

```
NDANCH                  CLLI            EXTRKNUM    BCSTAREA
---------------------------------------------------------
SWITCHB_REM  1         OG_OSAC_B_VL    1           N
   |
   |
SWITCHB_REM  96        OG_OSAC_B_VL    96          N
BAS_22       1000      BAS_SNVL22      1           N
   |
   |
BAS_22       1047      BAS_SNVL22      48          N
BAS_32       1000      BAS_SNVL32      1           N
   |
   |
BAS_32       1047      BAS_SNVL32      48          N
```

Figure 366 MAP display example for table OAVLMAP (OSAC remote)

```
NDANCH              CLLI            EXTRKNUM    BCSTAREA
----------------------------------------------------------
SWITCHA_HOST   1     IC_OSAC_A_VL     1           N
   |
   |
SWITCHA_HOST   96    IC_OSAC_A_VL     96          N
```

OSCVLGRP

Datafill in table OSCVLGRP identifies the voice trunk group used to connect to another OSAC switch. This is defined only at the host switch. In the following example, the OG_OSAC_B_VL voice link group is defined.

Figure 367 MAP display example for table OSCVLGRP (OSAC host)

```
NODENAME         CLLI
----------------------------------
SWITCHB_REM      OG_OSAC_B_VL
```

OFCENG

Datafill in table OFCENG specifies the number of recording units for OSSAIN and OSAC call processing traffic. OSAC RUs and OSSAIN RUs are datafilled at the host switch. In the following example, 100 OSSAIN RUs are specified for call processing at the OSAC host; 96 OSAC RUs are specified for OSAC call processing.

Figure 368 MAP display example for table OFCENG (OSAC host)

```
PARMNAME            PARMVAL
---------------------------------------
OSSAIN_NUM_RU       100 (See Note 1)
OSAC_NUM_RU         96 (see Note 2)
```

Note 1: This value applies only to call processing in the standalone/host.

Note 2: This value must be large enough for all OSAC remote switches in the OSAC configuration. Refer to Chapter 6: "OSSAIN engineering" for a calculation of OSAC recording units.

Figure 369 MAP display example for table OFCENG (OSAC remote)

```
PARMNAME            PARMVAL
----------------------------------
OSSAIN_NUM_RU       96
```

Billing and Access Services calling scenarios

This section lists the steps in several Billing and Access Services call flows. These scenarios use datafill shown in the examples.

0- BAS call in the standalone/host

1 TOPS QMS refines the call to the BAS_0_MINUS_OSSAIN call type for queuing (CT4Q).

2 Table OAINCTLA associates this CT4Q with the BAS_0_MINUS_C control list.

3 Table OACTLDEF associates this control list with the BAS_0_MINUS_F function.

4 Table OAFUNDEF associates this function with the CQ32 call queue.

5 Table OQCQPROF associates this call queue with an OSSAIN call queue profile of 10.

6 Table OASESNPL associates this profile with session pool 30 (BAS22_1) and the BAS_22 SN node name.

7 Table OANODINV associates this node name with node ID 11.

8 BAS_22 receives the session begin inform message and requests a voice channel.

9 Table OAVLMAP associates the BAS_22 logical ID to a particular BAS_SNVL22 voice trunk.

Disposition routing

In case the session pool or SN is out of service or no RUs are available, table OAFNDISP determines the disposition routing for each function. For the BAS_0_MINUS_F function, the call routes to the BAS_ENG_AGENT_C control list. This control list maps to the BAS_ENG_AGENT_F *operator function* in table OAFUNDEF.

0- BAS call and handoff to BAC node in the standalone/host

1 The call follows steps 1 to 9 from the previous scenario.

2 If billing acceptance (BAC) is required for the call, the SN requests a transfer to the BAS_BAC_C control list.

3 Table OACTLDEF associates this control list with the BAS_BAC_F function.

4 Table OAFUNDEF associates this function with the CQ34 call queue.

5 Table OQCQPROF associates this call queue with an OSSAIN call queue profile of 12.

6 Table OASESNPL associates this profile with session pool 34 (BAS32_1) and the BAS_32 SN node name.

7 Table OANODINV associates this node name with node ID 15.

8 BAS_32 receives the session begin inform message and requests a voice channel.

9 Table OAVLMAP associates the BAS_32 logical ID to a particular BAS_SNVL32 voice trunk.

Disposition routing

In case the session pool or SN is out of service or no RUs are available, table OAFNDISP determines the disposition routing. For the BAS_BAC_F function, the call routes to the LEGACY_AABS_C control list. This control list maps to the LEGACY_AABS_F *automated system function* in table OAFUNDEF.

Operator handoff to BAC node in the standalone/host

1 The operator has a call that is eligible for handoff to AABS.

2 The operator keys the handoff sequence and releases the call.

3 Table OAINPARM (parameter ALT_BILL_HANDOFF_METHOD) determines that the call goes to OSSAIN call processing using the BAS_BAC_OSSAIN CT4Q.

4 Table OAINCTLA associates this CT4Q with the BAS_BAC_C control list.

5 The call follows steps 3 to 9 from the previous scenario, to establish a voice connection with the BAS_32 SN.

Disposition routing

In case the session pool or SN is out of service or no RUs are available, table OAFNDISP determines the disposition routing. For the BAS_BAC_F function, the call routes to the LEGACY_AABS_C control list. This control list maps to the LEGACY_AABS_F *automated system function* in table OAFUNDEF.

0- BAS call in the OSAC remote

At the OSAC remote switch:

1 TOPS QMS refines the call to the BAS_0_MINUS_OSSAIN CT4Q.

2 Table OAINCTLA associates this CT4Q with the BAS_0_MINUS_C control list.

3 Table OACTLDEF associates this control list with the BAS_0_MINUS_F function.

4 Table OAFUNDEF associates this function with the REM_TO_HOST OSAC session pool.

5 Table OASESNPL associates this session pool with the SWITCHA_HOST (OSAC host) node name.

6 Table OANODINV associates this node name with node ID 30.

7 The OSAC remote sends an OSAC Session Begin Request message to the OSAC host.

At the OSAC host switch:

8 Table OAFUNDEF associates the BAS_0_MINUS_F function with the CQ32 call queue.

9 Table OQCQPROF associates this call queue with an OSSAIN call queue profile of 10.

10 Table OASESNPL associates this profile with session pool 30 (BAS22_1) and the BAS_22 SN node name.

11 Table OANODINV associates this node name with node ID 11.

12 The OSAC host responds to the OSAC remote with an OSAC Session Begin Request Success message, which specifies session pool 30 (BAS22_1) and the session ID.

At the OSAC remote switch:

13 The session begin message is sent to BAS_22. BAS_22 requests a voice channel.

14 BAS_22 sends the logical voice channel ID in the OSAC Voice Connect Request message to the OSAC host.

At the OSAC host switch:

15 Table OAVLMAP associates the BAS_22 logical ID to a particular BAS_SNVL22 voice trunk.

16 The OSAC host selects a host-remote voice channel to the remote. Table OSCVLGRP associates the OG_OSAC_B_VL voice link group with the OSAC remote switch. The trunk is selected from this group.

17 A connection is made in the OSAC host between the BAS_SNVL22 voice trunk and the selected OG_OSAC_B_VL voice trunk.

18 Table OAVLMAP associates the selected OG_OSAC_B_VL voice trunk to a particular SWITCHB_REM logical channel ID. This channel ID is sent back to the OSAC remote in the OSAC Voice Connect Success Response.

At the OSAC remote switch:

19 A connection is made in the OSAC remote between the incoming TOPS trunk and the OSAC voice trunk associated with the logical channel ID received from the OSAC host. This completes the voice path from the OSAC remote switch to the BAS_22 SN.

Disposition routing

In case the session pool or SN is out of service or no RUs are available, table OAFNDISP determines the disposition routing. For the BAS_0_MINUS_F function, the call routes to the BAS_ENG_AGENT_C control list. This control list maps to the BAS_ENG_AGENT_F *operator function* in table OAFUNDEF.

0+ DA BAS call

These calls work the same as 0- BAS calls, except at the end of the SN interaction, when the SN requests a transfer to a control list for a DA operator.

Note: Ensure that the TA and DA call queues and functions are datafilled with the correct service (DASERV or TASERV).

Appendix E: DA Sequence Calling

DA Sequence Calling allows the subscriber to press one key to receive subsequent DA listings after the call completion portion of a DA call. Operating companies can implement the DA Sequence Calling service at the TOPS switch using datafill for QMS queuing and OSSAIN functions, control lists, and triggers. Appendix E: “DA Sequence Calling,” describes how to set up the service, focusing on the following topics:

- understanding how DA Sequence Calling works
- datafilling the tables required for the service
- stepping through a call flow scenario
- analyzing fields in AMA records and operational measurements (OM)
- other considerations, such as hardware and software requirements, and feature interactions and restrictions

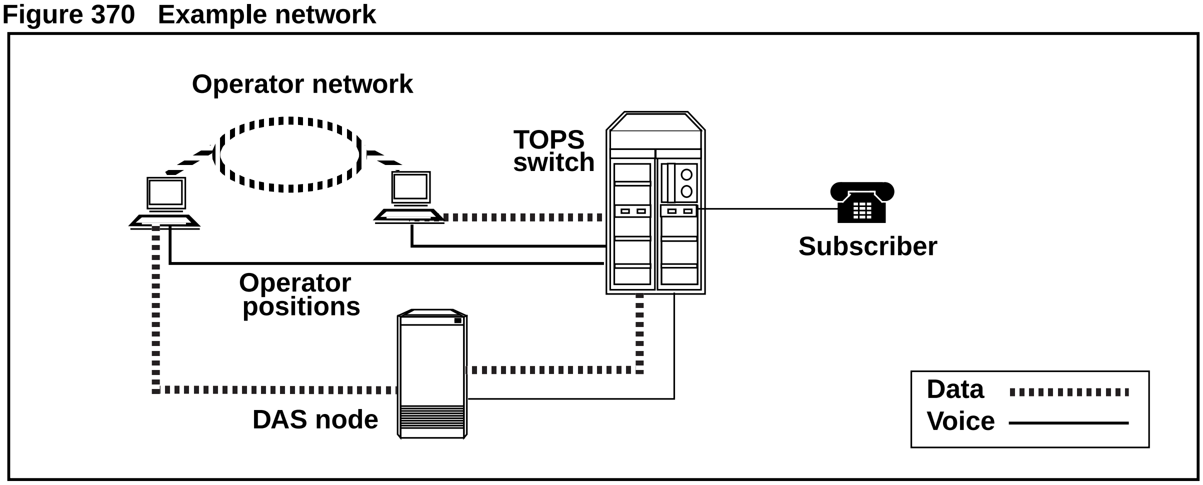
Understanding how DA Sequence Calling works

This section provides a sample network diagram and a description of call processing.

Network diagram

The following figure shows a sample network used in the DA Sequence Calling flow.

Figure 370 Example network



Call processing

In a DA Sequence Calling scenario, the TOPS switch routes the call back to a DA operator when the calling party presses a pre-defined DTMF digit. This digit serves as a trigger event for OSSAIN call floated trigger processing.

During trigger processing, the switch can monitor for a trigger event in the following call states:

- *connecting*, which occurs after float and before answer
- *talking*, which occurs after answer and before disconnect
- *disconnecting*, which occurs after disconnect

Note: In this chapter, the example scenario defines * (STAR) as the DTMF digit trigger event; however, * is only one of the DTMF digits that can be defined as triggers. See "Datafilling other DTMF digits as triggers" on page 660 for more information.

With a combination of datafill for QMS and OSSAIN, setting up DA Sequence Calling consists of the following two broad steps:

1 Route DA calls through an OSSAIN *control list* that contains a DA operator function.

   QMS refinement determines whether to route the call to OSSAIN or to TOPS. Calls routed to OSSAIN can have a control list associated with an operator function. This operator function has a corresponding DA call type for queuing (CT4Q). So, calls can use OSSAIN to route directly to a DA operator.

2 Include the DTMF trigger in the *trigger profile index* that is associated with the CT4Q.

   When the call is floated, OSSAIN uses the trigger profile index to detect the DTMF trigger during various call states. After the trigger is detected, the call is routed to the DA operator.

Datafilling DA Sequence Calling

To implement DA Sequence Calling at the TOPS switch, the following areas of processing require datafill:

- QMS queuing
- OSSAIN functions, control lists, and triggers

Assumption

Users must be familiar with standard TOPS QMS processing and the QMS datafill specific to their network. For details on QMS, refer to the *Translations Guide*.

Datafill sequence

Table 203 lists the steps needed to datafill DA Sequence Calling. Each step corresponds to a user task and a switch data table. A brief explanation and datafill example for each step follows Table 203.

Note: For complete details on the fields in the OSSAIN tables, refer to Chapter 7: "OSSAIN data schema."

Table 203 Datafill sequence for setting up DA Sequence Calling

Step	Task	Table
1	Define OSSAIN and TOPS QMS call type for queuing (CT4Q) names and initial trigger profile index	CT4QNAMS
2	Assign an initial CT4Q for the call origination (CO) type	QMSTOPS
3	Define OSSAIN functions	OAFUNDEF
4	Define OSSAIN control lists	OACTLDEF
5	Specify DTMF digit trigger profile	OADTFPRF
6	Specify connecting state trigger profile	OACNNPRF
7	Specify talking state trigger profile	OATLKPRF
8	Specify disconnecting state trigger profile	OADSCPRF
9	Specify an overall trigger profile index	OATPRFIX
10	Associate an OSSAIN CT4Q name with a control list name	OAINCTLA
11	Specify the maximum number of triggers per call	OAINPARM

Note: DA calls can be routed to an operator through either an OSSAIN control list or a TOPS operator CT4Q. This example routes the call through an OSSAIN control list for the purpose of associating a unique network service ID with a control list (see Step 4). For more information on routing through a TOPS operator CT4Q, refer to "CT4Q and trigger profile index—table CT4QNAMS" on page 83.

Step 1—CT4QNAMS

Any CT4Q name used in call queue refinement must first be datafilled in this table. In this example, the first tuple routes the call to OSSAIN, using the CT4Q DA_411. The second tuple routes the call to a TOPS operator, using the CT4Q DA_OPER.

Figure 371 MAP display example for table CT4QNAMS

```
CT4QNUM      CT4QNAME   NOAMA  ITRIGIDX   SYSAREA
----------------------------------------------------------------------
1            DA_411     N        Y 10     OSSAIN
2            DA_OPER    N        Y 10     TOPSOPER Y DA_CQ N N N
```

Step 2—QMSTOPS

In this example, the initial CT4Q for the 411 CO type is DA_411.

Figure 372 MAP display example for table QMSTOPS

```
CO                     CT4Q
-----------------------------------------------------------------
411                    DA_411
```

Note: In the sample DA Sequence Calling scenario, no further call queue refinements are datafilled. However, the operating company can set up call queue refinements as appropriate for the network. If additional refinements are performed, the last NEWCT4Q is used to index table CT4QNAMS in order to distinguish between OSSAIN and TOPSOPER.

Step 3—OAFUNDEF

In this example, the OSSAIN function name, DA_OPER_FN, is datafilled with a function type of TOPSOPER and an operator CT4Q name of DA_OPER.

Figure 373 MAP display example for table OAFUNDEF

```
FUNCID     FUNCNAME      FUNCAREA
-------------------------------------------------------------------
66         DA_OPER_FN    TOPSOPER DA_OPER  N
```

Step 4—OACTLDEF

In this example, the OSSAIN control list DA_CTL is associated with the function DA_OPER_FN (from table OAFUNDEF). DA_CTL is used in initial DA calls. The control list DA_SEQ_CTL is also associated with the function DA_OPER_FN. DA_SEQ_CTL is used in subsequent (sequence) DA calls.

Figure 374 MAP display example for table OACTLDEF

```
OACTLNUM   OACTLNAM     NETWRKID     OAFUNCTS
-------------------------------------------------------------------
50         DA_CTL       42           (DA_OPER_FN)
51         DA_SEQ_CTL   43           (DA_OPER_FN) $
```

Note: The NETWRKID value is arbitrary in this example. Because the NETWRKID value from OACTLDEF is used in AMA module code 185, operating companies can use the ID to identify which control list was in effect for the subsequent portion of a DA Sequence call. For an example AMA record, see “AMA records” on page 654.

Step 5—OADTFPRF

In this example, the STAR digit is datafilled and is associated with the control list DA_SEQ_CTL, which maps to the function DA_OPER_FN (table OACTLDEF). When the subscriber presses *, the call is routed to the DA_OPER_FN function, which routes the call to a DA operator (table OAFUNDEF).

Figure 375 MAP display example for table OADTFPRF

```
DTMFIDX      PROFILE                                        HOLDRCVR
------------------------------------------------------------------
200          (STAR Y Y N  CTRLLIST  DA_SEQ_CTL) $   N
```

Note: When triggering for a subsequent DA listing, an AMA record is not automatically generated for the portion of the call prior to the trigger. In the example, the GENAMA and RLSFWD fields have Y Y datafilled to allow the switch to generate AMA and release the forward party. For more information, refer to “AMA interactions” on page 656.

Step 6—OACNNPRF

In this example, the connecting trigger specifies a DTMF profile trigger index of 200 (table OADTFPRF).

Figure 376 MAP display example for table OACNNPRF

```
CONNIDX        PROFILE       DTMFPRF      CAUSEPRF
-----------------------------------------------------------------
100            $             Y 200        N
```

Step 7—OATLKPRF

In this example, the talking trigger specifies the control list DA_SEQ_CTL as the STAR trigger action.

Figure 377 MAP display example for table OATLKPRF

```
TALKIDX    PROFILE     STAR
                                                  OCTO    HOLDRCVR
-----------------------------------------------------------------
400        $          Y Y Y Y N CALLING CTRLLIST DA_SEQ_CTL
                                                  N      N
```

Step 8—OADSCPRF

In this example, the disconnecting trigger specifies a DTMF profile trigger index of 200 (table OADTFPRF).

Figure 378 MAP display example for table OADSCPRF

```
DISCIDX      PROFILE          DTMFPRF
-------------------------------------------------------------------
300          $                Y 200
```

Step 9—OATPRFIX

In this example, the overall trigger profile index of 10 points to a profile index for the three call states: 100 for connecting (table OACNNPRF); 400 for talking (table OATLKPRF); and 300 for disconnecting (table OADSCPRF).

Figure 379 MAP display example for table OATPRFIX

```
IDX     CPROFIDX     TPROFIDX    DPROFIDX
-------------------------------------------------------------------
10      Y 100        Y 400       Y 300
```

Step 10—OAINCTLA

In this example, the CT4Q DA_411, which routes the call to OSSAIN, is associated with the OSSAIN control list name DA_CTL (table OACTLDEF).

Figure 380 MAP display example for table OAINCTLA

```
CT4Q           OALISTNM
--------------------------------------------------------------------
DA_411         DA_CTL
```

Step 11—OAINPARM

In this example, the parameter MAX_NUM_TRIGGERS specifies a maximum number of 15 triggers allowed for an individual call. (The range of values is 0 to 100, with 100 being the default value.)

Figure 381 MAP display example for table OAINPARM

```
PARMNAME                 PARMVAL

--------------------------------------------------------------------

MAX_NUM_TRIGGERS         15
```

Sample DA Sequence Calling scenario

Steps in the sample DA Sequence Calling scenario are listed in this section. The scenario reflects two parts: the initial DA call and the subsequent DA call. Specific datafill values are referenced from the previous examples.

Initial DA call

1 Subscriber dials 411.

2 Call arrives at the TOPS office with a CO type of 411. Table QMSTOPS identifies an initial CT4Q of DA_411. No further call queue refinements are made to the call.

3 Table CT4QNAMS identifies CT4Q of DA_411, which routes the call to OSSAIN and associates this CT4Q with a trigger profile index of 10.

4 Table OAINCTLA associates this CT4Q with the OSSAIN DA_CTL control list.

5 Table OACTLDEF associates the DA_CTL control list with the DA_OPER_FN function.

6 Table OAFUNDEF routes the DA_OPER_FN to a DA operator.

7 Table CT4QNAMS identifies the CT4Q of DA_OPER for the operator. From here, standard QMS datafill routes the call to the appropriate operator.

8 Subscriber receives a listing and accepts call completion. The call is floated. The switch generates an AMA record for the initial DA portion of the call.

9 During float trigger processing, a trigger profile index of 10 is in effect (from step 3). Table OATPRFIX specifies a disconnecting trigger profile index of 300, which maps to a DTMF digit profile of 200 in table OADSCPRF. The DTMF digit profile of 200 in table OADTFPRF is set to monitor for the * trigger.

10 Called party disconnects. The switch generates an AMA record for the call completion portion of the call.

11 The switch begins to monitor for *, because the * trigger is in the disconnecting trigger profile.

Subsequent DA call

12 Subscriber enters * to trigger to a DA operator.

13 Table OADTFPRF specifies the STAR DTMF digit as a trigger for the DA_SEQ_CTL control list.

14 Table OACTLDEF associates the DA_SEQ_CTL control list with the DA_OPER_FN function.

15 Table OAFUNDEF routes the DA_OPER_FN to a DA operator.

16 Table CT4QNAMS identifies the CT4Q of DA_OPER for the operator. From here, standard QMS datafill routes the call to the appropriate team of operators.

17 Subscriber receives another listing and accepts call completion. Call is floated. The switch generates an AMA record for the subsequent DA portion of the call.

18 Subscriber talks to called party and disconnects. The switch generates an AMA record for the subsequent call completion portion of the call.

AMA records

This section shows examples of the four Bellcore AMA Format (BAF) billing records generated during the sample DA Sequence Calling scenario:

- Record 1—call code 194 for the initial 411 DA portion
- Record 2—call code 192 for the initial DA call completion portion
- Record 3—call code 194 for the subsequent trigger to DA operator (calling party presses * after called party goes onhook) portion
- Record 4—call code 192 for the subsequent DA call completion portion

Specific datafill values are referenced from the previous examples. All four records have a module code 185 appended, which records the NETWRKID (table OACTLDEF) in the SERV CODE field. The NETWRKID corresponds to the control list that was in effect for that portion of the call.

Note 1: For billing purposes, operating companies can use different NETWRKID values to distinguish between the initial DA call and any subsequent DA calls.

Note 2: For details on OSSAIN module codes, refer to Chapter 9: “OSSAIN billing.”

Figure 382 Record 1—initial 411 DA portion

```
*HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:194C  SENSOR TYPE:036C
SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC OFFICE ID:0123456C
DATE:81118C  TIMING IND:00000C  STUDY IND:0200002C  SERVICE OBSERVED:2C
ORIG NPA:619C  ORIG NUMBER:3204373C  CONNECT TIME:1327034C
ELAPSED TIME:000000040C  OPERATOR IDS:1060604009999C
ACC OPERATOR WORK TIME:00020C  SERVICE FEATURE:000C
STATION SIGNALING IND:2C  SCREENING CODE:000C  CALLED NUMBER INPUT:1C
CALLING NUMBER SOURCE:1C  MODULE CODE:311C  ORIG CALL TYPE:036C
MODULE CODE:185C  SERV CODE:00042C  MODULE CODE:307C
LINE NUMBER TYPE:001C NPA LINE:00201C  LINE NUMBER:2200008C
MODULE CODE:310C  KEYING ACTIONS:111111111C  MODULE CODE:055C
SERVICE ID:001C  MEANS OF INFORMATION INPUT:1C  MEANS OF LSDB ACCESS:1C
LSDBS BOC ID:11234C  LSDB ACCESSES:FF  LISTING RESPONSE:2C
LISTING STATUS:999999900C  REQUEST COUNTER:001C  MODULE CODE:104C
TRUNK FACILITY ID:101360001C  MODULE CODE:000C
```

Figure 383 Record 2—initial DA call completion portion

```
*HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:192C  SENSOR TYPE:036C
SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC OFFICE ID:0123456C
DATE:81118C  TIMING IND:00000C  STUDY IND:0200000C  SERVICE OBSERVED:2C
ORIG NPA:619C  ORIG NUMBER:3204373C  CONNECT TIME:1327113C
ELAPSED TIME:000000056C  OPERATOR IDS:0009999009999C
ACC OPERATOR WORK TIME:00000C  SERVICE FEATURE:000C
STATION SIGNALING IND:2C  SCREENING CODE:000C  CALLED NUMBER INPUT:4C
CALLING NUMBER SOURCE:1C  MODULE CODE:316C  LOCAL INDICATOR:1C
MODULE CODE:311C ORIG CALL TYPE:003C  MODULE CODE:051C  OVERSEAS IND:0C
TERM NPA:00201C  TERM NUMBER:2200008C  COMPLETION IND:001C  RATE IND:1C
OSS CALL COMPLETION CONDITIONS:3111710C  MODULE CODE:185C
SERV CODE:00042C  MODULE CODE:104C  TRUNK FACILITY ID:101360001C
MODULE CODE:000C
```

Figure 384 Record 3—subsequent DA portion

```
*HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:194C  SENSOR TYPE:036C
SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC OFFICE ID:0123456C
DATE:81118C  TIMING IND:00000C  STUDY IND:0200000C  SERVICE OBSERVED:0C
ORIG NPA:619C  ORIG NUMBER:3204373C  CONNECT TIME:1327229C
ELAPSED TIME:000000040C  OPERATOR IDS:1060604009999C
ACC OPERATOR WORK TIME:00020C  SERVICE FEATURE:000C
STATION SIGNALING IND:2C  SCREENING CODE:000C  CALLED NUMBER INPUT:4C
CALLING NUMBER SOURCE:1C  MODULE CODE:311C  ORIG CALL TYPE:003C
MODULE CODE:185C  SERV CODE:00043C  MODULE CODE:307C
LINE NUMBER TYPE:001C NPA LINE:00201C  LINE NUMBER:2200009C
MODULE CODE:310C  KEYING ACTIONS:111111111C  MODULE CODE:055C
SERVICE ID:001C  MEANS OF INFORMATION INPUT:1C  MEANS OF LSDB ACCESS:1C
LSDBS BOC ID:11234C  LSDB ACCESSES:FF  LISTING RESPONSE:2C
LISTING STATUS:999999900C  REQUEST COUNTER:001C  MODULE CODE:104C
TRUNK FACILITY ID:101360001C  MODULE CODE:000C
```

Figure 385 Record 4—subsequent DA call completion portion

```
*HEX ID:AA  STRUCTURE CODE:40752C  CALL CODE:192C  SENSOR TYPE:036C
SENSOR ID:0123456C  REC OFFICE TYPE:036C  REC OFFICE ID:0123456C
DATE:81118C  TIMING IND:00000C  STUDY IND:0200000C  SERVICE OBSERVED:0C
ORIG NPA:619C  ORIG NUMBER:3204373C  CONNECT TIME:1327308C
ELAPSED TIME:000000057C  OPERATOR IDS:0009999009999C
ACC OPERATOR WORK TIME:00000C  SERVICE FEATURE:000C
STATION SIGNALING IND:2C  SCREENING CODE:000C  CALLED NUMBER INPUT:4C
CALLING NUMBER SOURCE:1C  MODULE CODE:316C  LOCAL INDICATOR:1C
MODULE CODE:311C  ORIG CALL TYPE:003C  MODULE CODE:051C  OVERSEAS IND:0C
TERM NPA:00201C  TERM NUMBER:2200009C  COMPLETION IND:001C  RATE IND:1C
OSS CALL COMPLETION CONDITIONS:3111710C  MODULE CODE:185C
SERV CODE:00043C  MODULE CODE:104C  TRUNK FACILITY ID:101360001C
MODULE CODE:000C
```

AMA interactions

It is recommended in table OADTFPRF that the values for GENAMA and RLSFWD both be set to Y to allow the switch to generate AMA and release the forward party (see page 651). However, the following interactions apply to calls that trigger for subsequent DA listings, *regardless* of the datafill for GENAMA and RLSFWD:

- Triggering in the *connecting state*—If a call triggers during the connecting state before the B-party answers, the B-party connection is released and an AMA record is generated before the call is presented to the operator.
- Triggering in the *talking state*—If a call triggers during the talking state and the B-party is still attached, the time of the request is used as the disconnect time for the AMA record, but the B-party is not released. The time of the request is also used as the connect time for the AMA record.
- Triggering in the *disconnecting state*—If a call triggers during the disconnecting state and the B-party connection is not yet released, the B-party connection is released and an AMA record is generated.

OMs

The OM group OAFLTRIG (OSSAIN Float Triggers) provides peg counts during a DA Sequence Calling scenario, as described in this section.

OAFLTRIG

OM group OAFLTRIG counts actions related to OSSAIN float trigger processing that cause call control to transfer to a function or a control list. The following table describes each register.

Table 204 OM group OAFLTRIG

Register	Description
OAFLTFUN	OSSAIN float to function. This register is pegged each time a call in the floated state triggers and call control passes successfully to a function.
OAFLTCTL	OSSAIN float to control list. This register is pegged each time a call in the floated state triggers and call control passes successfully to a control list.

Table 204 OM group OAFLTRIG

Register	Description
OATRIGFL	OSSAIN trigger failure. This register is pegged each time a call fails an attempt to trigger. Failure can occur under one of the following conditions: - passing control to an OSSAIN function that is defined as a TOPS automated system - passing control to an OSSAIN control list whose associated function is defined as a TOPS automated system - failing to enable the required OSAN SOC options - attempting to trigger to the operator during an OSSAIN conference

The following figure shows an example for OM group OAFLTRIG.

Figure 386 MAP display example for OM group OAFLTRIG

```
CLASS: ACTIVE
START:1999/06/07 13:30:00 WED; STOP: 1999/06/07 13:33:14 WED;
SLOWSAMPLES:         2; FASTSAMPLES:         4;

     OAFLTFUN    OAFLTCTL    OATRIGFL

            0           2           0
```

Other considerations

Other considerations for DA Sequence Calling focus on the following areas:

- hardware requirements
- software requirements
- interactions
- restrictions
- datafilling other DTMF digits as triggers
- additional CT4Q refinements for operator calls

Hardware requirements

If the * digit trigger is defined for the talking state, two specialized tone receiver (STR) circuits are needed on each digital trunk controller (DTC). DA Sequence Calling has no other unique hardware requirements. Implementing DA Sequence Calling does not affect any hardware configurations for operator centralization, operator positions, or voice links.

Software requirements

The OSAN0101 OSSAIN SOC option must be enabled (ON) at the TOPS switch.

Interactions

This section describes how DA Sequence Calling interacts with various TOPS features.

DA call completion

Datafill in tables DABILL and ADACCOPT is required for DA call completion, as follows:

- A 411 STATION tuple is needed in both tables for the initial DA call.
- A DD STATION tuple is needed in both tables because the 411 CO type is changed to DD if the caller accepts call completion and bills the call as station paid.
- An OA STATION tuple is needed in both tables because the 411 CO type is changed to OA if the caller accepts call completion and uses alternate billing.

The following figures show example datafill.

Figure 387 MAP display example for table DABILL

```
DABKEY          REQNUM  CLGREQ   SERVNAME   ENHAMA  CCSURCHG   DACC
--------------------------------------------------------------------
411 STATION     N       Y        LOCALDA    Y       Y          AUTO $
DD  STATION     N       Y        LOCALDA    Y       Y          AUTO $
OA  STATION     N       Y        LOCALDA    Y       Y          AUTO $
```

Figure 388 MAP display example for table ADACCOPT

```
ACCKEY          BILTYPES     DISPLAY
---------------------------------------------------------------------
411 STATION     ALL     $     C CC_RCL
DD  STATION     SENTPD  $     C CC_RCL
OA  STATION     CONTBIL $     C CC_RCL
```

Note: When SOC OSAN0102 is ON for OSSAIN calls, an entry of NONE in the BILTYPES field in table ADACCOPT indicates that NO billing types are valid except auto-collect.

DA automation

If a call that triggers to a DA operator is eligible for ADAS or ADAS Plus service, the service is offered before the call is presented to the DA operator.

For ADAS Plus, parallel datafill is required between the DMS switch and the Directory One system, as follows:

- The ADNUM value from table CLLI in the DMS switch should be datafilled in the Directory One 411 Trunk table.
- The final CT4QNUM value from table CT4QNAMS in the DMS switch should be datafilled in the Directory One Call Control Datablock.

Release link trunking

RLT is allowed on DA sequence calls only for the two trigger events supported by the UCS-250 switch: OCTO (#) and PTYD (party disconnect). The UCS-250 switch transmits the applicable trigger value to the TOPS switch. So, if the TOPS switch takes calls from a UCS-250 switch, datafill at the TOPS switch should be set up to monitor for the # trigger instead of the * trigger.

Note: To support the DA Sequence Calling service, a software load of UCS08 or greater is required in the UCS-250 switch. For more information on OSSAIN interactions with RLT, refer to Chapter 5: “OSSAIN interactions.”

Operator centralization

DA Sequence Calling does not affect OC.

Restrictions

The following restrictions apply to calls that trigger for subsequent DA listings:

- At the time the subscriber accepts call completion, the CO type is changed to either DD or OA, depending on how the call is billed. Also, the base service is changed from DASERV to TASERV.

  When the call triggers during call completion, the CO type is not changed. If the final call queue upon triggering to an operator function has a base service of DASERV, the base service is changed from TASERV to DASERV.
- If a call is billed to a calling card or third number, the switch does *not* re-launch another LIDB query. However, the operator can re-class charge the call at the position to force another LIDB query.
- Datafilling a TOPSOPER function or control list against the CLGD (calling disconnect) trigger event in table OACNNPRF is not allowed. This is because during the connecting state there is no called party, and if the calling party disconnects, there are no parties remaining on the call.
- Triggering to an operator is not allowed when the call is in an OSSAIN conference. If triggering is attempted, an OAIN303 log is produced.
- When triggering to a function instead of to a control list, a new NETWRKID (see page 650) is not applied to the call. In this case, the NETWRKID cannot be used to distinguish between the initial DA call and any subsequent DA calls.

Datafilling other DTMF digits as triggers

Table OADTFPRF specifies DTMF triggers. STAR (*), OCTO (#), and the digits 0 to 9 are all valid values for DTMF triggers. This table is associated with the DTMFPRF field in both the connecting trigger profile table (OACNNPRF) and the disconnecting trigger profile table (OADSCPRF). The talking trigger profile table (OATLKPRF), however, allows only the STAR and the OCTO DTMF digits as triggers. For details on these tables, refer to Chapter 7: "OSSAIN data schema."

Additional CT4Q refinements for operator calls

Additional QMS CT4Q refinements can be applied to OSSAIN calls that route to an operator. By allowing CT4Q refinements, the existing QMS capability of segregating operator traffic is extended to OSSAIN calls. So, instead of datafilling multiple functions to segregate OSSAIN operator traffic, *one* function can funnel the calls to an operator CT4Q that is refined based on the queuing criteria (such as class of service, language, time of day, and so on). For details, refer to Chapter 4: "OSSAIN enhancements."

List of terms

AABS

Automated Alternate Billing Service

ACCS

Automated Calling Card System

ACTS

Automated Coin Toll Service

Address Resolution Protocol (ARP)

A protocol used by the Internet Protocol (IP) routing service to translate IP addresses into Ethernet addresses.

ADS

AudioGram / Automated Delivery Service

agent

A concept representing the external entities that connect with the switch. Examples include subscribers, conference port circuits, operator voice and data, and service node (SN) voice and data.

AMA

automatic message accounting

ARP

Address Resolution Protocol

Automated Alternate Billing Service (AABS)

A DMS TOPS feature that allows automated call completion of a calling card, collect, and third-number billed calls. AABS is the only existing TOPS automated system that can be datafilled in an OSSAIN control list.

Automated Coin Toll Service (ACTS)

A feature package that allows the operating company to handle long distance (1+) coin calls from a coin station without operator assistance.

Automated Credit Card System (ACCS)

A feature that allows the subscriber to dial a call and bill it to a calling card number provided by the operating company.

Automatic Call Distribution

A set of MDC features that assigns answering priorities to incoming calls and then queues and distributes the calls to a pre-determined group of telephone sets designated as agent positions.

automatic message accounting (AMA)

An automatic recording system that documents all the necessary billing data of subscriber-dialed long distance calls.

BAF

Bellcore AMA format

Bellcore AMA format (BAF)

The standard format for AMA data used by Bell operating companies. The format consists of a structure code that identifies the format of the data fields in the call record, a call code that identifies the type of call recorded in the call record, other data fields that define the attributes of the call, and if needed, one or more module codes that identify the format of any additional data appended to the call record.

branding

A feature that allows operating companies the option to connect customer-definable announcements to directory assistance (DA) or toll calls before placing them in a queue or connecting them to an available operator or automated operator system.

call code

A call type descriptor used in AMA recording. The call code defines the type of call or statistic being recorded.

call context block

A generic block of data that contains additional information about a call or the parties involved in a call. It is passed by a service node (SN) or operator terminal to the DMS switch.

call floated trigger processing

An OSSAIN capability that allows a call to recall back to a service node (SN) after the call has been floated. Floated calls can transfer to a function or a control list.

call type for queuing (CT4Q)

In TOPS and OSSAIN, a method of characterizing an incoming call based on certain criteria, so that the call can be assigned a queue to receive service.

CCITT

From the French for International Telegraph and Telephone Consultative Committee (Commite Consultatif International Telegraphique et Telephonique). The CCITT is one of the four permanent groups within the International Telecommunications Union (ITU). The CCITT is responsible for studying technical, operating, and tariff issues. This organization also prepares recommendations relating to telegraphy and telephony.

CCS7

common channel signaling 7

central office (CO)

A central office arranged for terminating subscriber lines and provided with switching equipment trunks for establishing connections to and from other switching offices.

central processing unit (CPU)

The hardware unit of a computing system that contains the circuits that control and perform the execution of instructions.

centralized OSSAIN (OSAC)

A TOPS architecture in which call processing control is distributed among more than one switch and several *centralized* SNs. OSAC allows the services offered by the centralized SNs to be shared by multiple switches.

centralized service node (SN)

An SN in a centralized OSSAIN (OSAC) configuration. Services provided by a centralized SN can be shared by multiple switches. A centralized SN has the node type of either OSNM or OSN. *See also* OSN and OSNM.

CI

command interpreter

CM

computing module

CO

central office

command interpreter (CI)

A component in the Support Operating System that functions as the main interface between the machine and the user.

common channel signaling 7 (CCS7)

A digital message-based network signaling standard, defined by the CCITT, that separates call signaling information from voice channels so that interoffice signaling is exchanged over a separate signaling link.

computing module (CM)

The processor and memory of the dual-plane combined core used by the DMS SuperNode. Each CM consists of a pair of CPUs with associated memory that operate in a synchronous matched mode on two separate planes. Only one plane is active; it maintains overall control of the system while the other plane is on standby.

control list

The mechanism used to interwork OSSAIN with TOPS functionality. The control list contains the name of a function, such as branding, that OSSAIN applies to the call. Control lists allow the operating company to use their installed equipment base and software in conjunction with OSSAIN functions.

country direct

A TOPS functionality that allows a caller in a foreign country to dial an access code to reach an operator in the country in which the call is to be billed (usually the home country).

CPU

central processing unit

CT4Q

call type for queuing

DAS

directory assistance system

data block

A logical grouping of data fields in the Open Automated Protocol (OAP) message format. Each protocol operation or response can have zero or more data blocks.

deflection

In OSSAIN processing, a disposition that causes a call to be routed to treatment, transferred to another function, or transferred to another control list, because the wait time for the call queue has been exceeded.

Digital Multiplex System (DMS)

A central office (CO) switching system in which all external signals are converted to digital data and stored in assigned time slots. Switching is performed by reassigning the original time slots.

digital recorded announcement machine (DRAM)

A peripheral module in which voice messages are stored in digital form, providing access to up to 30 different voice announcements.

digital trunk controller (DTC)

A peripheral module that connects DS30 links from the network with digital trunk circuits.

directory assistance system (DAS)

A system that provides directory assistance information and call intercept service.

directory number (DN)

The full complement of digits required to designate a subscriber’s station within one numbering plan area (NPA)—usually a three-digit central office (CO) code followed by a four-digit station number.

direct transfer

In OSSAIN processing, a type of transition that allows a call to transfer directly from a service node (SN) or an operator to a function provider.

DMS

Digital Multiplex System

DMS SuperNode

A central control complex for the DMS-100 switch. The two major components of the DMS SuperNode are the computing module (CM) and the message switch (MS). Both are compatible with the network module (NM), the input/output controller (IOC), and the XMS-based peripheral modules (XPM).

DN

directory number

DRAM

digital recorded announcement machine

DS-0

A protocol for data transmission that represents one channel in a 24-channel DS-1 trunk.

DS-1

The 8-bit 24-channel 1.544-Mb/s digital signaling format used in the DMS-100 Family switches. The DS-1 signal is the North American standard for digital trunks. It is closely specified bipolar pulse stream. DS-1 is the standard signal used to interconnect Nortel digital systems. DS-1 carries 24 information channels of 64 kb/s each (DS-0).

DS30

A 10-bit 32-channel 2.048-Mb/s speech-signaling and message-signaling link as used in the DMS-100 Family switches.

DS512

The fiber optic transmission link implemented in the DMS SuperNode processor. The DS512 is used for connecting the computing module to the message switch. One DS512 is equivalent to 16 DS30 links.

DTC

digital trunk controller

DTMF

dual-tone multifrequency

dual-tone multifrequency (DTMF) signaling

A signaling method that uses set combinations of two specific voice-band frequencies. One of these voice-band frequencies is selected from a group of four low frequencies, and the other is selected from a group of three or four relatively high frequencies.

EBAF

expanded Bellcore AMA format

EIU

Ethernet interface unit

end office (EO)

A switching office (SO) arranged for terminating subscriber lines and provided with trunks for establishing connections to and from other SOs.

ENET

enhanced network

enhanced network (ENET)

A channel-matrixed time switch that provides pulse code modulated voice and data connections between peripheral modules. ENET also provides message paths to the DMS-bus components.

EO

end office

Ethernet interface unit (EIU)

The unit that connects the DMS SuperNode to the local area network.

expanded Bellcore AMA format (EBAF)

The ability to append additional data in modular form to Bellcore AMA format (BAF) call records. Module codes are used to identify the format for the data appended to the BAF call record. One or more modules can be appended to a BAF record.

failed message threshold

The maximum number of failed messages originating from a service node (SN) that, once exceeded, determines the switch should take the node out of service.

Fiber Link Interface Shelf (FLIS)

The DMS SuperNode component that connects the computing module (CM) with the message switch (MS) using a DS512 fiber link. In the OSSAIN network, the Ethernet interface units (EIU) can be provisioned on a FLIS.

file transfer protocol (FTP)

A protocol used to transfer files, such as load files and patch files, across the Ethernet local area network (LAN) facility.

FLIS

Fiber Link Interface Shelf

FTP

file transfer protocol

function

A service or portion of a service that is provided by a service node (SN), an operator, or an existing TOPS automated system. Examples of functions are branding and alternate billing.

function provider

In OSSAIN processing, a service node (SN), an operator, or an existing TOPS automated system.

HMI

human-machine interface

human-machine interface (HMI)

The series of commands and responses used by operating company personnel to communicate with the DMS-100 Family switches. Communication takes place through the MAP terminal and other input/output devices.

initial program load (IPL)

The initialization procedure that causes a computer operating system to start operation.

input/output controller (IOC)

An equipment shelf that provides an interface between up to 36 input/output devices and the central message controller. The IOC contains a peripheral processor that independently performs local tasks, thus relieving the load on the CPU.

integrated services digital network (ISDN)

A set of standards proposed by the CCITT to establish compatibility between the telephone network and various data terminals and device. ISDN is a fully digital network, in general evolving from a telephone integrated digital network. It provides end-to-end connectivity to support a wide range of services, including circuit-switched voice, circuit-switched data, and packet-switched data over the same local facility.

Internet addressing

Physical or subnet addressing used by the Internet Protocol (IP) in which each host is assigned a unique integer address, written in the form of decimal notation. The address is referred to as IP address.

Internet Protocol (IP)

A protocol used at the network layer in data communication across the Ethernet local area network (LAN).

IOC

input/output controller

IP

Internet Protocol

IPL

initial program load

ISDN

integrated service digital network

ISDN user part (ISUP)

A common channel signaling 7 (CCS7) message-based signaling protocol that acts as a transport carrier for ISDN services. ISUP provides the functionality in a CCS7 network for voice and data services.

ISUP

ISDN user part

JNET

junctored network

junctored network (JNET)

A time-division multiplexed system that allows switching of 1920 channels per network pair (fully duplicated).

LAN

local area network

LIDB

line information database

LIM

link interface module

line information database (LIDB)

A database used to query alternate billed intra-LATA calls. The LIDB relays to the DMS switch information regarding billing number verification for a given dialing number (for example, the collect bill-to-third calls that are always refused and the collect bill-to-third calls that are always accepted).

link interface module (LIM)

A peripheral module (PM) that controls messaging between link interface units (LIU) in a link peripheral processor (LPP). The LIM also controls messaged between the LPP and the DMS-Bus component. An LIM consists of two local message switches and two frame transport buses. One LMS operates in a load-sharing mode with the other LMS.

link interface unit (LIU)

A peripheral module (PM) that processes messages entering and leaving a link peripheral processor (LPP) through an individual signaling data link.

link peripheral processor (LPP)

The DMS SuperNode equipment frame for DMS STP that contains two types of peripheral modules (PM): a link interface module (LIM) and a link interface unit (LIU). In the OSSAIN network, the Ethernet interface units (EIU) can be provisioned on an LPP.

LIU

link interface unit

LMS

local message switch

local area network (LAN)

A network that permits the interconnection and intercommunication of a group of computers. In the OSSAIN network, the DMS switch uses an Ethernet LAN to exchange call control and maintenance messages with the service nodes (SN).

local message switch (LMS)

A component that controls messaging between link interface units (LIU) and application processor units in a link peripheral processor (LPP). The LMS also controls messages between the LPP and the DMS-Bus.

local number portability (LNP)

A circuit switched network capability that allows telephone subscribers to keep their directory number (DN) when they change service providers. The subscriber keeps the same DN when the DN is moved, or *ported*, to a different end office. Other subscribers can connect to the ported DN without changing their dialing procedure.

LNP

local number portability

LPP

link peripheral processor

maintenance and administration position (MAP)

A group of components that provides a user interface between operating company personnel and the DMS-100 Family of switches. The interface consists of a video display unit and keyboard, a voice communications module, test facilities, and special furniture.

MAP

maintenance and administration position

MAU

media access unit

MCCS

mechanized calling card service

media access unit (MAU)

The local area network circuitry required by Ethernet interface units (EIU) to connect to the hubs in the LAN-bay.

message switch (MS)

A high-capacity communications facility that functions as the messaging hub of the dual-plane combined core of a DMS SuperNode processor. The MS controls messaging between the DMS-Bus components by concentrating and distributing messages and by allowing other DMS-STP components to communicate directly with each other.

MF

multifrequency

MIS node

An external reporting facility that receives data, which is used to report statistics on the functioning of call queues and agents (or service node sessions). *See also* Queue Management System Management Information System (QMS MIS).

module code

An identifier that defines a set of additional data fields to be appended o the base AMA record.

MS

message switch

multifrequency (MF)

A signaling method that makes use of pairs of standard toes to transmit signaling codes, digit pulsing, and coin-control signals. The method is used by interregister signaling on analog trunks.

NBP

network broadcast port

network broadcast port (NBP)

A DS30 port between a peripheral module and a network module that has broadcast voice links assigned to at least one of its channels.

Network File System (NFS)

A distributed file system protocol that allows a computer on a network to use the files and peripherals of another networked computer as if they were local.

network module (NM)

The basic building block of the DMS-100 Family switches. The NM accepts incoming calls and uses connection instructions fro the central control complex to connect the incoming calls to the appropriate outgoing channels. Network module controllers control the activities in the NM.

network subgroup (NSG)

A group of peripheral module ports (either 8 or 16) on a junctored network that are associated with a particular crosspoint card.

NFS

Network File System

NM

network module

node audit

A message sent by the switch to a service node (SN) when the switch has not received a message from the SN within a pre-defined period.

Northern Telecom publication (NTP)

A document that contains descriptive information about Northern Telecom (Nortel) hardware or software modules and performance-oriented practice for installing, testing, or maintaining the system. The document is often supplied as part of the standard documentation package provided to an operating company.

NSG

network subgroup

NTP

Northern Telecom publication

OAP

Open Automated Protocol

OM

operational measurements

Open Automated Protocol (OAP)

The protocol required to communicate data between a TOPS switch and an external OSSAIN service node (SN).

Open Position Protocol (OPP)

The protocol required to communicate data between a TOPS switch and an OPP-compatible terminal, such as the TOPS IWS.

operational measurements (OM)

The hardware and software resource of the DMS-100 Family switches that control the collection and display of measurements taken on an operating system. The OM subsystem organizes the measurement data and manages its transfer to displays and records. The OM data is used for maintenance, traffic, accounting, and provisioning decisions.

Operator Services Node (OSN)

A centralized service node datafilled at the OSAC remote switch. The OSN node is associated with the OSNM node at the OSAC host switch. The OSN node type is not used in a standalone OSSAIN network.

Operator Services Node Maintained (OSNM)

A service node (SN) that has a maintenance relationship with the DMS TOPS switch. The OSNM node type is used for SNs in the standalone OSSAIN network and for centralized SNs in the OSAC network.

Operator Services System Advanced Intelligent Network (OSSAIN)

A generic switch-to-service node (SN) interface that allows SNs to control switch functionality associated with operator services. There are two basic OSSAIN network configurations: standalone OSSAIN and centralized OSSAIN (OSAC).

OPP

Open Position Protocol

OSAC

centralized OSSAIN

OSN

Operator Services Node

OSNM

Operator Services Node Maintained

OSSAIN

Operator Services System Advanced Intelligent Network

OSSAIN preprocessing

A limited session with an OSSAIN service node provided to a TOPS call prior to connecting to an operator or automated system.

overflow

In OSSAIN processing, a disposition that causes a call to be routed to treatment, transferred to another function, or transferred to another control list, because no agents are available for the call queue.

pass-through message

In an OSSAIN simultaneous connection, pass-through messages are blocks of data sent in the Open Position Protocol (OPP) between the switch and an OPP-compatible position; or sent in the Open Automated Protocol (OAP) between the switch and a service node (SN).

PCL

product computing module load

PEC

product engineering code

peripheral module (PM)

A generic term referring to all hardware modules in the DMS-100 Family switches that provide interfaces between external line, trunk, or service facilities. A PM contains peripheral processors that perform routines, thus relieving the load on the CPU.

PIC

point in call

PM

peripheral module

point in call (PIC)

A call state consisting of null, initial call setup, call control, and call floated.

positive assertion

A way that the DMS switch handles receiving a call on an active session (service node session or host-remote session). In positive assertion, the switch takes down the first call and begins a new call on the same session.

preprocessing

See OSSAIN preprocessing

product computing module load (PCL)

The software load delivered to the operating company. A PCL contains both base and optional functionalities.

product engineering code (PEC)

An eight-character unique identifier for each marketable hardware item manufactured by Nortel.

QCALL utility

A utility that details the refinement and call assignment information of a single call having a unique set of characteristics. The QCALL command is accessed from the command interpreter at the MAP.

QMS

Queue Management System

QMS CAM

Queue Management System Call and Agent Manager

QMS MIS

Queue Management System Management Information System

Queue Management System (QMS)

A software package that provides enhanced capabilities for the management of call and agent queues.

Queue Management System Call and Agent Manager (QMS CAM)

The call queue management system for OSSAIN processing that has its own set of call queues and call queue profiles.

Queue Management System Management Information System (QMS MIS)

A switch application that collects event-driven data about TOPS and OSSAIN calls and sends this data to an external reporting facility, or *MIS node*. The data is used to report statistics on the functioning of call queues and agents (or service node sessions).

QVIEW utility

A utility that details the refinement and call assignment information of a set of calls. The QVIEW command is accessed from the command interpreter at the MAP.

release link trunking (RLT)

A feature that increases the capacity of ISUP trunks by releasing ISUP connections between a previous DMS-250 switch and a TOPS switch. After RLT is performed, ISUP connections are released, which makes circuits available for additional traffic.

Remote Operations Service Element (ROSE)

An application layer protocol that provides the capability to perform remote operations at a remote process.

right-to-use (RTU)

The permission granted to an operating company that allows the operating company to change the state of a software option and use the option. The operating company must receive a password for the option from Nortel before RTU is granted.

RIP

Routing Information Protocol

RLT

release link trunking

ROSE

Remote Operations Service Element

Routing Information Protocol (RIP)

An industry standard protocol that allows EIUs to exchange dynamic routing information with other IP routers on the Ethernet LAN.

RTU

right-to-use

sanity timer

A timer set by the DMS switch to ensure that resources are not left connected to dead calls. When a sanity timer expires, the switch automatically frees up any resources associated with the timer.

SCP

service control point

service control point (SCP)

A node in a common channel signaling 7 (CCS7) network that supports application databases. The function of an SCP is to accept a query for information, retrieve the requested information from one of its application databases, and send a response message to the originator of the request.

service node (SN)

An external node that interacts with the switch to provide OSSAIN services.

service provider identifier (SPID)

A code that uniquely identifies the service provider of the originating party.

service switching point (SSP)

A common channel signaling 7 (CCS7) node that interacts with the service control point (SCP) to implement special service code features.

session

In OSSAIN processing, an agent that serves call queues (equivalent to an operator).

session pool

Entire groups of sessions that service specific call queues. Session pools are associated with OSSAIN SNs (SN session pools) or with OSAC nodes (host-remote session pools).

signaling transfer point (STP)

A node in a common channel signaling 7(CCS7) network that routes messages between nodes. Signaling transfer points transfer messages between incoming and outgoing signaling links but, with the exception of network management information, do not originate or terminate messages. Signaling transfer points are deployed in pairs. If one STP fails, the mate takes over, ensuring that service continues without interruption.

simultaneous connection

Two function providers that are connected to a single call at the same time. OSSAIN allows a service node (SN) to connect to a call simultaneously with another SN or with an operator.

SLRN

special location routing number

SMS

Short Message Service

SN

service node

SO

switching office

SOC

software optionality control

software optionality control (SOC)

A tool for controlling and monitoring the options in a product computing module load (PCL).

special location routing number

An LRN that is associated with an incoming ISUP call that routes to the TOPS switch based on the LRN. When the call origination type is SLRN, the TOPS switch stores this special LRN for use in Queue Management System (QMS) routing.

SPID

service provider identifier

SSP

service switching point

state transition

A node change from one maintenance state to another; for example, from system busy to in service.

STP

signaling transfer point

structure code

An identifier that defines and provides structure to a set of data fields in an AMA record.

switching office (SO)

A node in the common channel signaling 7 (CCS7) network that originates and terminates signaling messages related to the set up and take down of associated ISDN user part (ISUP) trunks.

T1

The standard 24-channel 1.544-Mb/s pulse code modulation (PCM) system used in North America. This digital carrier carries a signal whose designation is a DS-1 link.

TA

toll and assist

TDP

trigger detection point

TLI

Transport Layer Interface

toll and assist (TA)

A facility that allows an operator to help a subscriber complete a dialed toll call.

TOPS

Traffic Operator Position System

TOPS IWS

Traffic Operator Position System Intelligent Workstation System

TOPS position controller (TPC)

A control unit that functions as a workstation-based microcomputer with networking capabilities.

TPC

TOPS position controller

Traffic Operator Position System (TOPS)

A call processing system made up of a number of operator positions. Each operator position consists of a visual display unit (VDU), a controller, a keyboard, and a headset.

Traffic Operator Position System Intelligent Workstation System (TOPS IWS)

An integrated operator assistance, intercept, and DA position, which uses a personal computer with customized software, keyboard, and interface.

transition

An activity that bypasses control list processing to allow for greater control of the OSSAIN call by the service node (SN).

Transport Layer Interface (TLI)

A generic interface used by applications to access transport layer protocols, such as User Datagram Protocol (UDP).

trigger

An event that causes an OSSAIN call to be redirected to an SN or operator.

trigger detection point (TDP)

The point in call processing when the switch determines the action associated with a trigger. Two possible actions are associated with a TDP: continue with normal call processing (no trigger is hit) or bridge on an SN or operator (trigger is hit).

trigger profile

A mechanism that allows operating companies to control trigger processing on an individual call basis.

UDP

User Datagram Protocol

User Datagram Protocol (UDP)

A member of the Internet protocol suite of protocols, UDP is used at the transport layer for OSSAIN data messaging.

VID

virtual terminal identifier

virtual terminal identifier (VID)

In OSSAIN, an internal data communications representation of a session in a session pool.

VLB

voice link broadcasting

voice link broadcasting (VLB)

A single one-way voice link that can connect up to 1023 different calls at the same time.

voice service node (VSN)

A processor external to the DMS switch that communicates with the switch through an application protocol to provide the voice recognition and prompt generation components of Automated Alternate Billing Service (AABS).

VSN

voice service node

WAN

wide area network

wide area network

A large-scale, high-speed communications network used primarily for interconnecting local area networks (LAN) located in different cities or nations.

Index

A

alternate routing methods 196, 215
AMA
  BAF format 375, 379–385
  custom AMA 375
  example TDR AMA record 391
  TDR format 375, 388–392
  See also billing
architecture
  centralized OSSAIN 36
  standalone OSSAIN 34
authorization codes 203
automatic carrier selection 204
automatic message accounting. *See* AMA
autoprovisioning 245

B

BAF billing format 375, 379–385
base service, changing 213
billing
  BAF call codes 379
  BAF fields 383
  BAF format 375, 379–385
  BAF module codes 380
  BAF modules 380, 381
  BAF structure codes 379
  custom AMA 375
  custom BAF modules 382, 384
  TDR fixed templates 391
  TDR format 375, 388–392
  TOPS Charge Adjust service 392
Billing and Access Service 623–646
blocking of OPP requests 229
branding 212
broadcast voice links 170, 175, 232, 241, 318
buffering MIS messages 194

C

Call and Agent Manager. *See* QMS CAM
call context block 111
call float restrictions 230
call floated trigger processing
  and trigger events 85
  interactions with operators 230
  interactions with SNs 230
  with operators and automated systems 101
  See also triggers
call flow scenario 99, 106
  standalone 44–45
call model
  call processing 49–55
  figure of 51
  TDPs in 79
call origination types 82, 228
call processing
  call model 49–55
  OSAC 131
  PICs 52–55
  transitions 50, 104
  triggers 79–102
call queues
  assignment of 84, 111
  call queue profiles 71, 292
  configuring 61–64
  datafilling 69, 292
  system limitations 237
call timing 167, 212
cause codes 88, 95
centralized SN 36, 39
commercial credit cards 203
conferencing 176, 217, 239
control lists
  datafilling 84, 107, 326
  direct transfers 103
  system limitations 237
country direct 185, 213, 364

D

DA Sequence Calling service 647–660
data blocks 121
data communications
  bandwidth requirements 272–273
  CM port numbers 593
  data messaging 586–589
  data port numbers 592
  software description 585–589
  TCP 588
  UDP 588
data store
  maximum datafill 247
  maximum IPL 247
datafilling
  Billing and Access Service 623–646
  call queue profiles 71, 292
  call queues 71, 281
  control lists 84, 326
  DA Sequence Calling 647–660
  deflections 73, 310
  EIUs 256–263
  functions 66, 301
  overflows 73, 310
  parallel datafill 315
  QMS CAM 69
  session pools 67, 148, 295
  transitions 325
  triggers 87, 332
  voice links 173, 318
  XA-Core Ethernet interfaces 267
deflections 73, 310
direct transfers. *See* transitions
disposition routing 108
DNSCRN screening 205
DTMF receivers 93, 240

E

EIU
  and LAN connectivity 248, 250
  engineering 237–264
  hardware datafill 257
  IP node datafill 256–263
  load balancing 253
  routing restrictions 250
  system limitations 237
engineering
  bandwidth requirements 272
  broadcast voice links 605
  conference three ports 238
  data store for OSSAIN application 247
  DTMF receivers 240
  EIUs 249–264
  hardware resources 238
  memory considerations 247
  specialized tone receivers 240
  XA-Core Ethernet interfaces 264–271
ENSV0014. *See* OSAN0101
ENSV0020. *See* OSAN0101
equal access interactions 214
estimate of charges 204
extension blocks 241, 243

F

feature interactions and restrictions 209–233
function blocking 105, 108, 329
function provider selection 109
functions
  call flow 105–106
  DAS 305
  datafilling 66, 109, 301
  identifier 56
  routing and queuing 56, 61
  system limitations 237

G

global support of OSSAIN 204, 215

H

hardware resources 238
  conference three ports 238
  DTMF receivers 240
  specialized tone receivers 240
  voice links 240

I

Internet Protocol
  address classes 589
  datafilling EIUs 256–263
  datafilling XA-Core Ethernet interfaces 267
  IP address restrictions 250, 265
  netmasks 590–592
  routing restrictions 250, 265
  subnets 590
  throttling numbers 262
ISUP cause codes 95
ISUP release classes 95

J

JNET office 605

L

LAN. *See* Internet Protocol and engineering
limitations
of OSSAIN features 209–233
of OSSAIN system 237, 250–251, 265–266
local number portability 163
logs
AUD logs 503
DAS logs 508
EXT logs 501
OAIN logs 453
OAP logs 490
OSAC logs 496
PM logs 505
QMIS logs 512
TCCI logs 510
TOPS logs 513

M

maintenance
alarms and SN state 452
MIS nodes 194
nodes 395–423
session pools 424–444
utilities for OSSAIN 444
MAP commands
OSSAINCI directory 444
QCALL directory 450
QVIEW directory 450
SOC 368
maximum datafill data store 247
maximum IPL data store 247
memory considerations 247
MIS. *See* QMS MIS

N

node types
datafilling 39, 282–292
maintenance of 395–423
OSAC 38, 396
OSN 39
OSNM 39
notification timer 92, 102, 230, 359

O

OAP
class headers 120
data blocks 121
documentation 27
in OSSAIN processing 122
message format 119
message responses 119
message types 119
operation header 120
overview 118
protocol version 123
OMs
country direct 520
DAS 521
extension blocks 522
float trigger processing 524
OAP call processing operations 531–549
OPP actIDs 578
OPP DIDs 579
OSSAIN data communications 522–565
OSSAIN QMS 526
service node call processing 566
session pool maintenance 565
Open Automated Protocol. *See* OAP
operational measurements. *See* OMs
operator handoff 160
OPP
blocking of requests 229
in transitions with SNs 114
OSAC
advanced configuration 597
architecture 36
call distribution 131–136
host-remote session pools 426
logs 496
messaging 131
OMs 566–576
OSAC node type 38, 396
OSN node type 39
OSNM node type 39
parallel datafill 143, 315
voice connections 139
OSAC node type 38, 396
OSAN0101 363, 364
OSAN0102 364
OSAN0103 366
OSN node type 39
OSNM node type 39
OSSAIN API 27
OSSAIN enhancements 153–206
OSSAIN support for DA automation 220
OSSAIN utility 444
outgoing ISUP 169, 224
overflows 73, 310

P

parallel datafill 143, 315
parameters

table OAINPARM 125–127, 357
table OFCAUT 245
table OFCENG 241
table TOPSPARM 361
pass-through messaging 158
PIC
call control PIC 53, 97, 98
call floated PIC 54, 97, 98
entry events and exit events 52–55
in call model 50
initial call setup PIC 52, 97, 98
null PIC 52
point in call. *See* PIC
port and channel mapping 615
port deloading 612
positive assertion 138
preprocessing 154, 218
protocols. *See* OAP, OPP, and OSAC
provisioning broadcast voice links 605–621

Q

QCALL utility 84
QMS CAM 55, 57, 69, 292
QMS CT4Q refinements 199, 210
QMS MIS 189, 364
Queue Management System. *See* QMS CAM
QVIEW utility 450

R

recording units 241, 243
release classes 95
release link trunking. *See RLT*
RLT 163
routing and queuing
components of 56
configurations 57–64
CT4Q refinements 199
datafilling 69
QMS CAM 55, 69
restrictions 225
session pools 56, 61
SN originated calls 78
table relationships 65
routing index 197
routing number 198

S

sanity timers
call sanity timer 127
on-hook timer 125
operation of 124–128
queued sanity timer 128
session begin timer 125
service nodes
in direct transfers 113
in routing and queuing 57–61
SN originated calls 78
system limitations 237
service provider identifier. *See SPID*
session counting 366
session limiting 369
session pool
identifier 56
session pools
datafilling 67, 148, 295
host-remote session pools 426
maintenance of 424–444
short message service 206
system limitations 237
short message service 206
simultaneous connections
active agent 157
passive agent 158
pass-through messaging 158
restrictions 226
SLRN 205, 226
SOC 363–368
functionality overview 363
MAP commands 368
OSAN0101 363, 364
OSAN0102 364
OSAN0103 366
OSSAIN option order codes 363
session counting 366
session limiting 369
software optionality control. *See* SOC
special location routing number. *See SLRN*
specialized tone receivers 91, 173, 240
SPID 161
state transitions
MIS nodes, figure of 195
nodes, figure of 403
related logs 405, 430
session pools, figure of 427
STR supervision 173
subnets. *See* Internet Protocol
system limitations 237

T

TCP 189, 588
TDP
and trigger events 81
call floated TDPs 85

in call model 50, 79
See also triggers
TDR billing format 375, 388–392
time switching 607
timers
notification timer 92, 102, 230, 359
sanity timers 125, 127, 128
toll-free services 221
TOPS LNP 163
transition mechanisms 111
transition types 112–117
transitions
between functions 111
direct transfers 103, 111–117, 325
function provider selection 105–111
restrictions 226–229
trigger profiles
connecting 87–90
disconnecting 93
system limitations 237
talking 91–92
triggers
call flow scenarios 99–100
call processing 50, 99
DA Sequence Calling 648
datafilling 87
restrictions 229
table relationships 97
trigger profile index 86
trunk group
TOPSVL type 75
voice connectivity 151, 316

U

UDP 406, 588
unbundling
SOC options 363–368
utilities
OSSAINCI directory 444
QCALL directory 84, 450
QVIEW directory 450

V

virtual terminal identifier 445, 478
VLB 170, 232
voice connectivity
auto voice link selection and connection 201, 211
broadcast voice links 170, 175, 232, 241, 318
datafilling 173, 318
in an OSAC environment 139, 151, 316, 322
standalone 75, 316
system limitations 237
voice link broadcasting. *See VLB*
voice link maintenance 77, 141

X

XA-Core Ethernet interface
capacity restrictions 266
datafill 267
engineering 264
IP address restrictions 265
routing restrictions 265
security restrictions 266

DMS-100 Family

OSSAIN

User's Guide

Publication number: 297-8403-901
Product release: SN09 and up
Document release: Standard 13.01
Date: July 2005
Printed in the United States of America

NORTEL NETWORKS

How the world shares ideas.