Critical Release Notice

Publication number: 297-2621-370
Publication release: Standard 10.03

The content of this customer NTP supports the SN07 (DMS) software release.

Bookmarks used in this NTP highlight the changes between the baseline NTP and the current release. The bookmarks provided are color-coded to identify release-specific content changes. NTP volumes that do not contain bookmarks indicate that the baseline NTP remains unchanged and is valid for the current release.

Bookmark Color Legend

Black: Applies to new or modified content for the baseline NTP that is valid through the current release.

Red: Applies to new or modified content for NA017 that is valid through the current release.

Blue: Applies to new or modified content for NA018 (SN05 DMS) that is valid through the current release.

Green: Applies to new or modified content for SN06 (DMS) that is valid through the current release.

Purple: Applies to new or modified content for SN07 (DMS) that is valid through the current release.

Attention!
Adobe Acrobat Reader 5.0 or higher is required to view bookmarks in color.

Publication History

September 2004

For the SN07 (DMS) release, 10.03, the following changes were added:

Volume 1

Added additional NetworkBuilder-related data schema information to the CAINPARM table to address CR Q00816405.

Volume 2

No changes

Volume 3

Added notes for CAIN parameter TRTMTCD_COMPCODE_ZAPPED_ZERO to address CR Q00816405.

Volume 4

No changes

Volume 5

No changes

September 2003

For the SN06 (DMS) release, 10.02, the following changes were added:

Volume 1

SN06 (DMS) Standard release 10.02. Added LNP_EVALUATE_AFTER_OTC_CIC information per CR Q00 509677-06.

Volume 2

No changes

Volume 3

No changes

Volume 4

No changes

Volume 5

No changes

297-2621–370

Digital Switching Systems

UCS DMS-250

NetworkBuilder Application Guide, Volume 4 of 5

UCS17 Standard 10.01 July 2002

NORTEL NETWORKS™

How the world shares ideas.

Digital Switching Systems

UCS DMS-250

NetworkBuilder Application Guide, Volume 4 of 5

Publication number: 297-2621–370
Product release: UCS17
Document release: Standard 10.01
Date: July 2002

Printed in the United States of America

NORTEL NETWORKS CONFIDENTIAL: The information contained herein is the property of Nortel Networks and is strictly confidential. Except as expressly authorized in writing by Nortel Networks, the holder shall keep all information contained herein confidential, shall disclose the information only to its employees with a need to know, and shall protect the information, in whole or in part, from disclosure and dissemination to third parties with the same degree of care it uses to protect its own confidential information, but with no less than reasonable care. Except as expressly authorized in writing by Nortel Networks, the holder is granted no rights to use the information contained herein.

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Contents

Volume 1 of 5

O_Active and O_Suspended PICs 9-1

T_Null PIC 10-1

Appendix A Service migration 11-1

Appendix B Engineering guidelines 12-1

Volume 2 of 5

NetworkBuilder call processing 1-1

Figures

Volume 3 of 5

Automatic Code Gapping 2-1

Event processing 3-1

Call configuration model 4-1

Incoming CAIN messages 10-1

Incoming IN/1 messages 11-1

Incoming CAIN message parameters 12-1

Volume 4 of 5

Volume 5 of 5

Conversational messages

ATTENTION
Extension parameters require the CAIN0200 SOC option. Refer to Volume 5, Chapter 5, “NetworkBuilder SOC functionality,” for more information on SOC.

SCP responses for digit collection

When the switch receives a **`Send_To_Resource`** or **`Connect_To_Resource`** message from the SCP in a Conversation package, the CAIN framework processes the response. Conversational digit collection evaluates the **`Send_To_Resource`** or **`Connect_To_Resource`** message and instructs the switch to play an announcement, play an announcement and collect digits, or connect to a resource capable of exchanging information with a subscriber.

The following table provides a list and describes the conversational digit collection messages.

Table 1-1
Conversational digit collection messages

Message	Description
`Call_Info_From_Resource`	This message provides intermediate information received from the IP through the switch during an active STR- or CTR-Connection.
`Call_Info_To_Resource`	This message provides a response to the intermediate information received from the IP through the switch during an STR- or CTR-Connection.
`Cancel_Resource_Event`	This message instructs the switch to discontinue caller interaction (which was initiated through a **`Send_To_Resource`** or **`Connect_To_Resource`** message from the SCP) and wait for further instructions from the SCP.
`CTR_Clear`	The UCS DMS-250 switch sends this message to the SCP to indicate the outcome of an SCP request for information (only used in response to a conversational **`Connect_To_Resource`** message).
`Resource_Clear`	The UCS DMS-250 switch sends this message to the SCP to indicate the outcome of an SCP request for information (only used in response to a conversational **`Send_To_Resource`** message).
`Send_To_Resource` or **`Connect_To_Resource`** with Play Announcement and Collect Digits (without a ***`DestinationAddress`*** parameter)	This message instructs the switch to play an announcement and collect digits.

Note 1: Messages are supported differently for LNP; refer to the *UCS DMS-250 Local Number Portability Application Guide* for more information.
Note 2: Messages are supported differently for AXXESS agents; refer to *UCS DMS-250 CAIN/FlexDial Interactions* for more information.
Note 3: Only one leg may enter digits which are collected by the switch. The ***`LegID`*** parameter determines from which call leg digits are collected. If no ***`LegID`*** parameter is present, then digits are collected from the controlling leg (***`LegID`*** 0).
Note 4: If a conversation package is received with a **`Connect_To_Resource`** with ***`FlexParameterBlock`***, then the switch sends a **`CTR_Clear`** message to the SCP in a conversation package with a ***`ClearCause`*** parameter of `taskRefused`. The call is not cleared toward the controlling leg or the passive leg.

—continued—

Table 1-1
Conversational digit collection messages (continued)

Message	Description
`Send_To_Resource` or **`Connect_To_Resource`** with ***`FlexParameterBlock`*** (without a ***`DestinationAddress`*** parameter)	This message instructs the switch to play an announcement(s) and collect a digit stream(s) (provides Virtual IP interactions). ***Note:*** The ***`FlexParameterBlock`*** message is not supported for **`O_Mid_Call`**.
`Send_To_Resource` or **`Connect_To_Resource`** (with ***`DestinationAddress`*** parameter)	This message instructs the switch to route to an IP (provides CONNECT_ONLY and 1129-STYLE IP interactions).

Note 1: Messages are supported differently for LNP; refer to the *UCS DMS-250 Local Number Portability Application Guide* for more information.
Note 2: Messages are supported differently for AXXESS agents; refer to *UCS DMS-250 CAIN/FlexDial Interactions* for more information.
Note 3: Only one leg may enter digits which are collected by the switch. The ***`LegID`*** parameter determines from which call leg digits are collected. If no ***`LegID`*** parameter is present, then digits are collected from the controlling leg (***`LegID`*** 0).
Note 4: If a conversation package is received with a **`Connect_To_Resource`** with ***`FlexParameterBlock`***, then the switch sends a **`CTR_Clear`** message to the SCP in a conversation package with a ***`ClearCause`*** parameter of `taskRefused`. The call is not cleared toward the controlling leg or the passive leg.

—end—

The following table provides a list and describes the messages sent by the SCP after digit collection.

Table 1-2
Incoming messages

Response	Description
`Analyze_Route`	This message instructs the UCS DMS-250 switch to resume CAIN call processing, using the address, billing, and routing information processing provided by the SCP. (Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for more information.) When received in conversation, this message should be accompanied by a `Request_Report_BCM_Event` component.
`Close`	This message indicates a nonfatal unexpected communication error. (Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for more information.)
`Continue`	This message instructs the UCS DMS-250 switch to route the call using in-switch routing information and perform as if the call had not triggered a query to the SCP. (Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for more information.) When received in conversation, this message should be accompanied by the `Request_Report_BCM_Event` component.
`Disconnect`	This message instructs the switch to disconnect the call and apply treatment. (Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for more information.)
`Request_Report_BCM_Event`	This non-call related component instructs the UCS DMS-250 switch to arm EDPs. (Refer to Volume 3, Chapter 3, “Event processing,” for more information.)

Note 1: For information on error messages received from the SCP, refer to Volume 3, Chapter 1, “TCAP messaging,” and the *UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition*.
Note 2: Response messages are supported differently for LNP, refer to the *UCS DMS-250 Local Number Portability (LNP) Application Guide* for more information.
Note 3: Response messages are supported differently for AXXESS agents, refer to *UCS DMS-250 CAIN/FlexDial Interactions* for more information.

The following table is applicable to **`Send_To_Resource`** or **`Connect_To_Resource`** connections and is not applicable to Virtual IP interactions. Due to DMS-250 size limitations for internal messaging the DMS-250 does not support the maximum size ***`STRParameterBlock`*** as specified in GR-1129 CORE and GR-1299 CORE.

Table 1-3
Size restrictions for FlexParameterBlock

Messaging scenario	**Parameter breakdown**	**IP trunk type**	
		PRI (Note1)	**ISUP**
`Send _To_Resource`	ParmID, tag and length bytes overhead (for both ***`STRParameterBlock`*** and ***`FlexParameterBlock`***	5	5
	Payload of ***`FlexParameterBlock`***	86	111
	Total	91	116
`Call_Info_To_Resource`	ParmID, tag and length bytes overhead (for both ***`STRParameterBlock`*** and ***`FlexParameterBlock`***	5	5
	Payload of ***`FlexParameterBlock`***	86	111
	Total	91	116
`Connect_To_Resource`	ParmID, tag and length bytes overhead (for both ***`STRParameterBlock`*** and ***`FlexParameterBlock`***	5	5
	Payload of ***`FlexParameterBlock`***	86	111
	Total	91	116
Note 1: Requires XPM load ELI81AZ or later.			
—end—			

The following table is applicable to `Send_To_Resource` or `Connect_To_Resource` connections and is not applicable to Virtual IP interactions. Due to DMS-250 switch size limitations for internal messaging the DMS-250 switch does not support the maximum size ***`IPReturnBlock`*** as specified in GR-1129 CORE and GR-1299 CORE.

Table 1-4
Size restrictions for IPReturnBlock

Messaging scenario	**Parameter breakdown**	**IP trunk type**	
		PRI (Note1)	**ISUP**
`Resource_Clear`	ParmID and length byte overhead	3	3
	Payload	92	115
	Total	95	118
`Call_Info_From_Resource`	ParmID and length byte overhead	3	3
	Payload	92	115
	Total	95 (Note2,3)	118
`CTR_Clear`	ParmID and length byte overhead	3	3
	Payload	92	115
	Total	95	118

Note 1: Requires XPM load ELI81AZ or later.

Note 2: Due to CM-XPM interactions, the `Resource_Clear` message is not sent when this limit is exceeded.

Note 3: This requires CCM10, XPM10 feature AJ5132- *PRI large FIE for incoming facility message* which increased the size of an incoming FIE from 57 bytes to 113 bytes.

—end—

Use

The SCP sends the **Call_Info_To_Resource** message to provide a response to the intermediate information received from the IP through the UCS DMS-250 switch during an active 1129-Style STR- or CTR-Connection. This message is used in response to the **Call_Info_From_Resource** message.

Message parameters

Table 1-5 provides the parameters the SCP returns to the UCS DMS-250 switch.

Table 1-5
Call_Info_To_Resource parameters

Parameter	Usage	Definition
ResourceType	Optional	This parameter indicates the type of resource for the connection.
StrParameterBlock	Optional	This parameter provides information the UCS DMS-250 switch or IP requires to perform the function requested by the ***ResourceType*** parameter.
ExtensionParameter	Optional	Extension parameters require the CAIN0200 SOC option.
billSequence Number	Optional	This extension parameter contains 32 bits of SCP-defined billing data that is stored in the CDR.
amaDigits	Optional	This extension parameter provides digit strings which are to be entered into the CDR for the call in progress.

Fatal application errors

None

Associated logs

CAIN100, CAIN101, CAIN200, CAIN201, VAMP901

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF

Call_Info_From_Resource (end)

Use

The UCS DMS-250 switch sends the `Call_Info_From_Resource` message to provide intermediate information from the IP to the SCP during an active 1129-Style STR- or CTR-Connection.

Message parameters

Table 1-6 provides the parameters that the SCP returns to the UCS DMS-250 switch.

Table 1-6
Call_Info_From_Resource parameters

Parameter	Usage	Definition
ExtensionParameter	Optional	Extension parameters require the CAIN0200 SOC option. Currently no extension parameters are sent for this message.
IPReturnBlock	Optional	This parameter contains the result of any user-interaction with the IP to send to the SCP

Fatal application errors

None

Associated logs

CAIN100, CAIN101, CAIN200, CAIN201, VAMP902

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF

Use

The SCP sends a `Cancel_Resource_Event` message when the UCS DMS-250 switch is processing an outstanding `Send_To_Resource`, `Connect_To_Resource`, or `Call_Info_From_Resource` operation initiated by the SCP. The `Cancel_Resource_Event` message directs the UCS DMS-250 switch to discontinue caller interaction and report to the SCP for further instructions.

Message parameters

None

Application errors

Table 1-7 provides the application errors associated with the `Cancel_Resource_Event`.

Table 1-7
Cancel_Resource_Event application errors

Error type	Package	Log generated	Reported to SCP?	Action performed
`Cancel_Resource_Event` in response to a query	Response	CAIN200 and CAIN201	Yes	Send `Report_Error` to SCP in a unidirectional package
`Cancel_Resource_Event` in response to a query	Conversation	CAIN200 and CAIN201	Yes	Send `Application_Error` to SCP
`Cancel_Resource_Event` in response to a `Resource_Clear` or `CTR_Clear`	Conversation	CAIN100 and CAIN101	Yes	`Cancel_Resource_Event` message is ignored
`Cancel_Resource_Event`	QWP	CAIN200 and CAIN201	Yes	Send `Application_Error` to SCP

Associated logs

CAIN100, CAIN101, CAIN200, CAIN201, VAMP902

Cancel_Resource_Event (end)

Associated OMs

CAINMSGR, CAINUIF

Use

Upon completing an interaction with the subscriber, the UCS DMS-250 switch sends a **`CTR_Clear`** message to the SCP indicating the outcome of the interaction. A **`CTR_Clear`** message is sent in response to a conversational **`Connect_To_Resource`** message.

Note 1: The UCS DMS-250 switch always sends a **`CTR_Clear`** message as the response to a conversational **`Connect_To_Resource`** messages, unless a TCAP error occurs.

Note 2: When the Mid Call Services 2 SOC option, CAIN0801, is enabled, and the UCS DMS-250 switch sends a **`CTR_Clear`** message while the call is at the *Timeout* event, the **`Disconnect`**, **`Continue`**, and **`Connect_To_Resource`** response messages are supported. Although the CAIN protocol allows the **`Analyze_Route`** and **`Collect_Information`** messages in response to **`CTR_Clear`**, a fatal application error occurs when one of these messages is received at the *Timeout* event.

Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for specific information regarding response messages; refer to Volume 3, Chapter 12, “Incoming CAIN message parameters,” for detailed descriptions of the response parameters; refer to Volume 3, Chapter 3, “Event processing,” for EDP specific message information. Refer Volume 5, Chapter 5, “NetworkBuilder SOC functionality,” for SOC specific information.

Table 1-8 provides the type of **`CTR_Clear`** messages that can be sent.

Table 1-8
CTR_Clear

Reason for message	Package	Contents of message
An SS7 DISCONNECT or SS7 RELEASE message is received without a component	Conversation	***`ClearCause`*** set to `abort`
An SS7 DISCONNECT or SS7 RELEASE message with a reject component is received by the switch	Conversation	***`ClearCause`*** set to `protocolError`

—continued—

CTR_Clear (continued)

Table 1-8
CTR_Clear (continued)

Reason for message	Package	Contents of message
An SS7 FAR, containing an RO parameter with a return result component, is received (while the switch is waiting for a CITR from the SCP)	Conversation	***ClearCause*** set to `protocolError`
An switch timer (not directly related to IP interactions) expires before an ISDN CONNECT or SS7 ANM message is received from the IP	Response	***ClearCause*** set to `strCancelled`
A timeout occurred while attempting to access the resource	Conversation	***ClearCause*** set to `timeout`
Call answered after **O_No_Answer** request to SCP	Response	***ClearCause*** set to `calledPartyAnswered`
Caller abandon	Response	***ClearCause*** set to `userAbandon` and ***CollectedDigits***
Dialing timeout	Conversation	***ClearCause*** set to `invalidCode` and ***CollectedDigits***
End of Dialing or Announcement complete	Conversation	***ClearCause*** set to `normal` and ***CollectedDigits***
For the Connect Only IPI, the switch is unable to connect to the IP at the remote switch	Conversation	***ClearCause*** set to `abort`
Indicates the long call duration timer expired during an STR or CTR Connection	Conversation	***ClearCause*** set to `strCancelled`
Invalid information received	Conversation	***ClearCause*** set to `invalidCode` and ***CollectedDigits***
Receipt of a **Call_Info_To_Resource** message without the ***ResourceType*** and ***StrParameterBlock*** from the SCP	Conversation	***ClearCause*** set to `normal`

—continued—

Table 1-8
CTR_Clear (continued)

Reason for message	Package	Contents of message
Resource unavailable	Conversation	***ClearCause*** set to `failure`, and ***FailureCause*** is set to `applicationError`
STR-Connection to a PRI IP and the selected trunk is not provisioned correctly.	Conversation	***ClearCause*** set to `abort`
Termination to an IMT that is not provisioned appropriately	Conversation	***ClearCause*** set to `taskRefused`
The combined size of the ***ResourceType*** and ***StrParameterBlock*** has exceeded the maximum size for the given terminating agency.	Conversation	***ClearCause*** set to `abort`
The ***DestinationAddress*** routes to a trunk that is not a PRI or SS7 IMT trunk.	Conversation	***ClearCause*** set to `abort`
The IPI is determined to be none.	Conversation	***ClearCause*** set to `taskRefused`
The IP initiates abnormal call clearing by sending an SS7 FAR message with a Cause Indicator of abnormal clearing and the message contains a reject component.	Conversation	***ClearCause*** set to `protocolError`
The IP initiates abnormal call clearing by sending an SS7 RELEASE COMPLETE or SS7 RELEASE message with a Cause Indicator of abnormal clearing and the message contains a reject component.	Conversation	***ClearCause*** set to `protocolError`
The IP initiates abnormal call clearing by sending an SS7 RELEASE COMPLETE or SS7 RELEASE message with a Cause Indicator of abnormal clearing and the message does not contain a component.	Conversation	***ClearCause*** set to `abort`

—continued—

CTR_Clear (continued)

Table 1-8
CTR_Clear (continued)

Reason for message	Package	Contents of message
The IP initiates normal call clearing with an SS7 DISCONNECT or SS7 RELEASE message with a Cause Indicator of normal and the message contains a Return Result component.	Conversation	***ClearCause*** set to `normal`
The local switch is unable to locate an idle trunk member within the route list.	Conversation	***ClearCause*** set to `channelsBusy`
The local switch encountered problems during an STR or CTR Connection.	Conversation	***ClearCause*** set to `abort`
The maximum time limit for an STR-Connection has expired.	Conversation	***ClearCause*** set to `ipTimeout`
The remote IP responds to a **Cancel_Resource_Event** with an SS7 DISCONNECT or RELEASE message containing a return result component.	Conversation	***ClearCause*** set to `resourceCancelled`
The SCP request has an encoding error and the switch can not determine a suitable action.	Conversation	***ClearCause*** set to `protocolError`
The SCP sends a **Cancel_Resource_Event** message and the switch is connected to a local IP.	Conversation	***ClearCause*** set to `resourceCancelled`
The specified resource is not supported.	Conversation	***ClearCause*** set to `resourceTypeNotSupported`
The UCS DMS-250 switch canceled the resource interaction.	Conversation	***ClearCause*** set to `resourceCancelled`
The UCS DMS-250 switch determined that the request is not allowed	Conversation	***ClearCause*** set to `taskRefused`

—continued—

Table 1-8
CTR_Clear (continued)

Reason for message	Package	Contents of message
The UCS DMS-250 switch was unable to terminate to the requested resource	Conversation	***ClearCause*** set to `resourceNotAvailable`
Timer TDISC expires before the IP responds to the SS7 FAR message with the cancellPresource operation	Response	***ClearCause*** set to `ipTimeout`
—end—		

Message parameters

Table 1-9 provides the parameters the UCS DMS–250 switch can return.

Table 1-9
CTR_Clear parameters

Parameter	Usage	Definition
ClearCause	Required	This parameter indicates caller abandon the call; the ***ClearCause*** parameter is set to `userAbandon`.
CcID	Optional	This parameter contains the current call configuration.
ClearCauseData	Optional	This parameter provides additional error information to the SCP when the IP responds with a Return Error component.
CollectedAddressInfo	Optional	This parameter contains the address collected by the switch in response to a request by the SCP for a “normal” number of digits.
CollectedDigits	Optional	The digits will be an undifferentiated stream of digits; the SCP is responsible for separating the digits into different numbers. CAIN call processing populates this field with digits collected during **`Connect_To_Resource`** conversational digit collection.
—continued—		

CTR_Clear (continued)

Table 1-9
CTR_Clear parameters (continued)

Parameter	Usage	Definition
FailureCause	Optional	This parameter indicates the reason a failure occured; this parameter is only returned when ***ClearCause*** is set to `failure`.
IPReturnBlock	Optional	This parameter contains the collected information requested by the **`Connect_To_Resource`** message.
LegID	Optional	This parameter normally contains 0, indicating the event was caused by the controlling leg of the call. In CC6, this parameter identifies the leg that disconnected.

—end—

The ***ClearCause*** parameter indicates the reason for termination of the connection between the resource and the caller. Table 1-10 provides the values CAIN supports.

Note: For a list of all possible ***ClearCause*** values, refer to the *UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition*.

Table 1-10
ClearCause values

Cause	Hex value	Meaning
`normal`	00	This value indicates the expected result for end of dialing or announcement is complete.
`timeout`	02	This values indicates a timeout occurred while attempting to access the resource.
`resourceCancelled`	03	This value indicates the switch cancelled the resource interaction.
`invalidLeg`	05	

—continued—

Table 1-10
ClearCause values (continued)

Cause	Hex value	Meaning
userAbandon	06	This value indicates the caller went onhook during the resource interaction.
invalidCode	07	This value indicates the reception of invalid digits or digit timeout.
failure	08	When set, the ***FailureCause*** parameter is sent, indicating the reason for failure.
channelsBusy	09	This value indicates the local, intermediate, or remote switch was unable to identify an idle trunk for termination to the IP.
calledPartyAnswered	0A	This value indicates an answer was received after **O_No_Answer** was sent to SCP.
resourceNotAvailable	0B	This value indicates the UCS DMS-250 switch was unable to terminate to the requested resource.
resourceTypeNotSupported	0D	This value indicates the specified resource is not supported.
taskRefused	0E	This value indicates the UCS DMS-250 switch determined that the request is not allowed.
invalidCallerResponse	0F	
capabilityFailure	10	
protocolError	11	This value indicates the SCP detects an encoding error and the UCS DMS-250 switch can not determine a suitable action.
abort	12	This value indicates the local switch encountered problems during an STR- or CTR-Connection.

—continued—

CTR_Clear (end)

Table 1-10
ClearCause values (continued)

Cause	Hex value	Meaning
temporaryFailure	15	
ipTimeout	16	
ctrCancelled	1F	This value indicates the long call duration timer expired during a CTR-Connection.
	—end—	

When ***ClearCause*** is set to failure, the ***FailureCause*** parameter is sent. ***FailureCause*** indicates the operation received from the SCP could not be performed because a hardware or software resource was unavailable. Table 1-11 provides the values that CAIN supports.

Table 1-11
FailureCause values

Cause	Digit value	Meaning
applicationError	10	A switch failure has occurred due to unavailability of a hardware or software resource.

Fatal application errors

None

Associated logs

The UCS DMS-250 switch generates various logs dependent on TCAP and UIF failures.

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF, CAINIP

Use

Upon completing an interaction with the subscriber, the UCS DMS-250 switch sends a **`Resource_Clear`** message to the SCP indicating the outcome of the interaction. A **`Resource_Clear`** message is sent in response to a conversational **`Send_To_Resource`** message.

Note: The UCS DMS-250 switch always sends a **`Resource_Clear`** message as the response to a conversational **`Send_To_Resource`** messages, unless a TCAP error occurs.

Table 1-12 provides the types of **`Resource_Clear`** messages that the UCS DMS-250 switch can send.

Table 1-12
Resource_Clear messages

Reason for message	Package	Contents of message
An SS7 DISCONNECT or SS7 RELEASE message is received without a component	Conversation	***`ClearCause`*** set to `abort`
An SS7 DISCONNECT or SS7 RELEASE message with a reject component is received by the switch	Conversation	***`ClearCause`*** set to `protocolError`
An SS7 FAR, containing an RO parameter with a return result component, is received (while the switch is waiting for a CITR from the SCP)	Conversation	***`ClearCause`*** set to `protocolError`
A switch timer (not directly related to IP interactions) expires before an ISDN CONNECT or SS7 ANM message is received from the IP	Response	***`ClearCause`*** set to `strCancelled`
A timeout occurred while attempting to access the resource	Conversation	***`ClearCause`*** set to `timeout`
Call answered after **`O_No_Answer`** request sent to SCP	Response	***`ClearCause`*** set to `calledPartyAnswered`

—continued—

Resource_Clear (continued)

Table 1-12
Resource_Clear messages (continued)

Reason for message	Package	Contents of message
Caller abandon	Response	***ClearCause*** set to `userAbandon` and ***CollectedDigits***
Dialing timeout	Conversation	***ClearCause*** set to `invalidCode` and ***CollectedDigits***
End of Dialing or Announcement complete	Conversation	***ClearCause*** set to `normal` and ***CollectedDigits***
For the Connect Only IPI, the SSP is unable to connect to the IP at the remote switch.	Conversation	***ClearCause*** set to `abort`
Indicates the long call duration timer expired during an STR or CTR Connection	Conversation	***ClearCause*** set to `strCancelled`
Invalid information received	Conversation	***ClearCause*** set to `invalidCode` and ***CollectedDigits***
Receipt of a **Call_Info_To_Resource** message without the ***ResourceType*** and ***StrParameterBlock*** from the SCP	Conversation	***ClearCause*** set to `normal`
Resource unavailable	Conversation	***ClearCause*** set to `failure`, and ***FailureCause*** is set to `applicationError`
STR-Connection to a PRI IP and the selected trunk is not provisioned correctly	Conversation	***ClearCause*** set to `abort`
Termination to an IMT that is not provisioned appropriately	Conversation	***ClearCause*** set to `taskRefused`
The combined size of the ***ResourceType*** and ***StrParameterBlock*** has exceeded the maximum size for the given terminating agency.	Conversation	***ClearCause*** set to `abort`

—continued—

Table 1-12
Resource_Clear messages (continued)

Reason for message	Package	Contents of message
The ***DestinationAddress*** routes to a trunk that is not a PRI or SS7 IMT trunk.	Conversation	***ClearCause*** set to `abort`
The IPI is determined to be none.	Conversation	***ClearCause*** set to `taskRefused`
The IP initiates abnormal call clearing by sending anr SS7 FAR message with a Cause Indicator of abnormal clearing and the message contains a reject component.	Conversation	***ClearCause*** set to `protocolError`
The IP initiates abnormal call clearing by sending an SS7 RELEASE COMPLETE or SS7 RELEASE message with a Cause Indicator of abnormal clearing and the message contains a reject component.	Conversation	***ClearCause*** set to `protocolError`
The IP initiates abnormal call clearing by sending an SS7 RELEASE COMPLETE or SS7 RELEASE message with a Cause Indicator of abnormal clearing and the message does not contain a component.	Conversation	***ClearCause*** set to `abort`
The IP initiates normal call clearing with an SS7 DISCONNECT or SS7 RELEASE message with a Cause Indicator of normal and the message contains a Return Result component.	Conversation	***ClearCause*** set to `normal`
The IP invokes a supplemental service on the local switch.	Response	***ClearCause*** set to `suppServiceInvoked`
The local switch is unable to locate an idle trunk member within the route list.	Conversation	***ClearCause*** set to `channelsBusy`

—continued—

Resource_Clear (continued)

Table 1-12
Resource_Clear messages (continued)

Reason for message	Package	Contents of message
The local switch encountered problems during an STR or CTR Connection.	Conversation	***ClearCause*** set to `abort`
The maximum time limit for an STR-Connection has expired.	Conversation	***ClearCause*** set to `ipTimeout`
The remote IP responds to a **Cancel_Resource_Event** with an SS7 DISCONNECT or RELEASE message containing a return result component.	Conversation	***ClearCause*** set to `resourceCancelled`
The SCP request has an encoding error and the switch can not determine a suitable action.	Conversation	***ClearCause*** set to `protocolError`
The SCP sends a **Cancel_Resource_Event** message and the switch is connected to a local IP.	Conversation	***ClearCause*** set to `resourceCancelled`
The specified resource is not supported.	Conversation	***ClearCause*** set to `resourceTypeNotSupported`
The UCS DMS-250 switch canceled the resource interaction.	Conversation	***ClearCause*** set to `resourceCancelled`
The UCS DMS-250 switch determined that the request is not allowed.	Conversation	***ClearCause*** set to `taskRefused`
The UCS DMS-250 switch was unable to terminate to the requested resource.	Conversation	***ClearCause*** set to `resourceNotAvailable`
Timer TDISC expires before the IP responds to the SS7 FAR message with the cancelIPresource operation	Response	***ClearCause*** set to `ipTimeout`

—end—

Message parameters

Table 1-13 provides the parameters the UCS DMS-250 switch can return.

Table 1-13
Resource_Clear parameter

Parameter	Usage	Definition
ClearCause	Required	This parameter indicates the reason a connection between a caller and a resource was terminated.
CollectedAddressInfo	Optional	This parameter contains the address collected by the UCS DMS-250 switch in response to a request by the SCP for a “normal” number of digits.
CollectedDigits	Required	The digits are an undifferentiated stream of digits; the SCP is responsible for separating the digits into different numbers. CAIN call processing populates this field with digits collected during **Send_To_Resource** conversational digit collection.
FailureCause	Optional	This parameter indicates the reason a failure occured; this parameter is only returned when ***ClearCause*** is set to failure.
ClearCauseData	Optional	This parameter provides additional error information to the SCP when the IP responds with a Return Error component.
IPReturnBlock	Optional	This parameter contains the collected information requested by the **Send_To_Resource** message.

ClearCause indicates the reason for termination of the connection between the resource and the caller. Table 1-14 provides the values CAIN supports.

Resource_Clear (continued)

Table 1-14
ClearCause values

Cause	Hex value	Meaning
normal	00	This value indicates the expected result for end of dialing or announcement completes.
timeout	02	This value indicates a timeout occurred while attempting to access the resource.
resourceCancelled	03	This value indicates the switch cancelled the resource interaction.
unansweredLeg	04	
invalidLeg	05	
userAbandon	06	This value indicates the caller went onhook during the resource interaction.
invalidCode	07	This value indicates the reception of invalid digits or digit timeout.
failure	08	This value indicates the received operation could not be performed due to the unavailability of a hardware or software resource.
channelsBusy	09	This value indicates the local, intermediate, or remote switch was unable to identify an idle trunk for termination to the IP.
calledPartyAnswered	0A	This value indicates an answer was received after **O_No_Answer** was sent to SCP.
resourceNotAvailable	0B	This value indicates the UCS DMS-250 switch was unable to terminate to the requested resource.
isdnTimeout	0C	
resourceTypeNotSupported	0D	This value indicates the specified resource is not supported.

—continued—

Resource_Clear (continued)

Table 1-14
ClearCause values (continued)

Cause	Hex value	Meaning
`taskRefused`	0E	This value indicaates the UCS DMS-250 switch determined that the request is not allowed due to SOC or CAIN_CONVERSATION_LIMIT exceeded.
`invalidCallerResponse`	0F	
`capabilityFailure`	10	
`protocolError`	11	This value indicates the SCP request encoding error and the UCS DMS-250 switch can not determine a suitable action.
`abort`	12	This value indicates the local switch encountered problems during an STR- or CTR-Connection.
`suppServiceInvoked`	13	This value indicates the IP invoked a supplemental service on the local switch (for example, RLT).
`strCancelled`	14	This value indicates the long call duration timer expired during an STR-Connection.
`temporaryFailure`	15	
`ipTimeout`	16	
	—end—	

When ***`ClearCause`*** is set to `failure`, the ***`FailureCause`*** parameter is sent. ***`FailureCause`*** indicates the operation received from the SCP could not be performed because a hardware or software resource was unavailable. Table 1- 15 provides the values CAIN supports.

Resource_Clear (end)

Table 1-15
FailureCause values

Cause	Digit value	Meaning
`applicationError`	10	A switch failure has occurred due to unavailability of a hardware or software resource.

Fatal application errors

None

Associated logs

The UCS DMS-250 switch generates various logs dependent on TCAP and UIF failures.

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF, CAINIP

Send_To_Resource and Connect_To_Resource (continued)

Use

The **Send_To_Resource** and **Connect_To_Resource** messages instruct the UCS DMS-250 switch to perform one of the following actions:

- play an announcement and disconnect (Refer to Volume 3, Chapter 10, “Incoming CAIN messages,” for more information.)
- play one or more announcements and collect digits
- route to an IP or perform Virtual IP
- play one or more announcements and collect one or more digit streams through in-switch Virtual IP

Note: Virtual IP is supported with **Connect_To_Resource** after reorigination, but not at ***O_Mid_Call*** *Timeout* events. If a **Connect_To_Resource** message is received at an ***O_Mid_Call*** *Timeout* event, a **CTR_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of taskRefused.

Additionally, the **Send_To_Resource** and **Connect_To_Resource** messages can identify the Flex Parameter Block. The Flex Parameter Block allows resources to be defined in a flexible manner. The Flex Parameter Block can be used to access any IP resource, whether it is a new resource or an existing one such as Play Announcement. This allows new resources to be encoded without requiring the assignment of a new ***ResourceType***.

Within the **Send_To_Resource** or **Connect_To_Resource** message, the ***DestinationAddress*** parameter identifies the location of an IP resource. The UCS DMS-250 switch interprets a **Send_To_Resource** or **Connect_To_Resource** message without a ***DestinationAddress*** parameter as a request to access an internal switch resource.

If a **Send_To_Resource** or **Connect_To_Resource** message is received within a conversation package, then, at the completion of the **Send_To_Resource** or **Connect_To_Resource** operation, the UCS DMS-250 switch returns a **Resource_Clear** or **CTR_Clear** message and waits for further information from the SCP.

The UCS DMS-250 switch accepts a **Send_To_Resource** or **Connect_To_Resource** message requesting an inswitch resource in a Response Package when the call is to be terminated after the interaction with the resource. When received in a Response Package, the ***DisconnectFlag*** parameter is mandatory.

Send_To_Resource and Connect_To_Resource (end)

The UCS DMS-250 switch accepts a **`Connect_To_Resource`** message in response to the **`Timeout`** EDP-Request. If the UCS DMS-250 switch receives a **`Connect_To_Resource`** message when it is not expected, then the UCS DMS-250 switch treats the message as containing a unexpected communication error and the following actions are performed:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- A CAIN200 Application Error log is generated.
- The call is not cleared toward the controlling leg or the passive leg.

Play announcements and collect digits

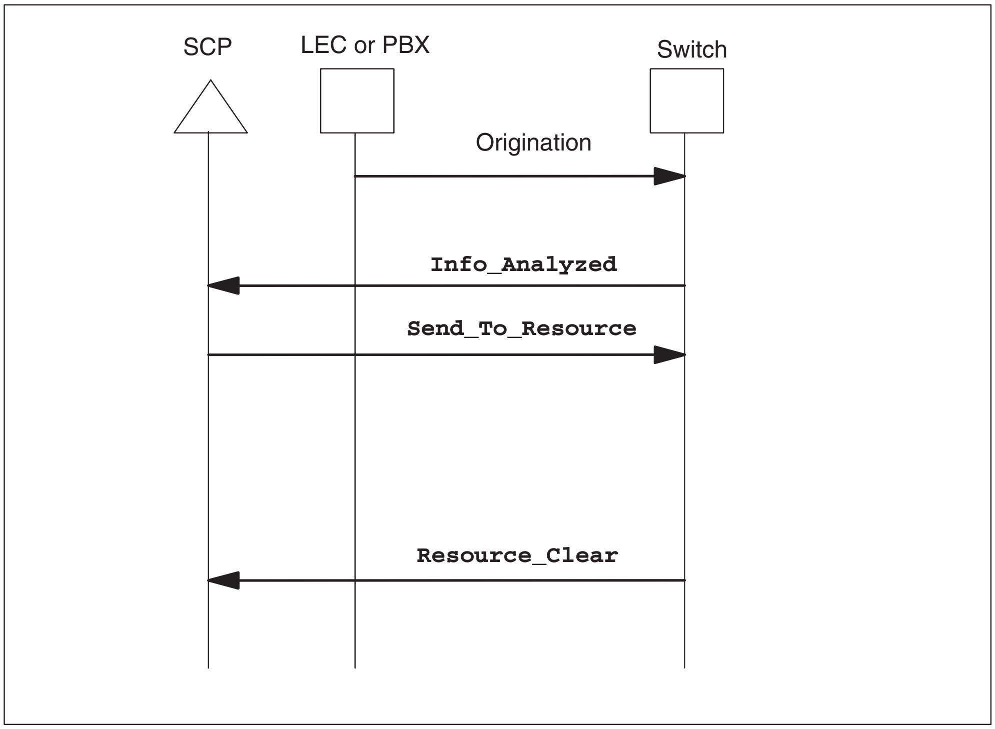
The play announcements and collect digits capability allows the switch to perform simple, prompted digit collection.

ATTENTION
Conversational digit collection requires the CAIN0600 SOC option. Refer to Volume 5, Chapter 5, "NetworkBuilder SOC functionality," for more information on CAIN SOC functionality.

A play announcement and collect digits `Send_To_Resource` or `Connect_To_Resource` message is transmitted in a Conversation package with a component of Invoke_Last.

Figure 1-1 shows the interaction between the switch and the SCP when the switch receives a `Send_To_Resource` message.

Figure 1-1
Send_To_Resource conversation



Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Table 1-16 provides the play announcement and collect digits parameters for the **Send_To_Resource** message.

Table 1-16
Send_To_Resource Play announcement and collect digits parameters

Parameter	Usage	Definition
ResourceType	Required	Contains Play Announcement and Collect Digits
STRParameterBlock	Required	Contains a Play Announcement and Collect Digits tag and one announcement digit block (Note 1):

Note 1: Announcements are played in the order received, uninterruptible announcements are played first, followed by the interruptible announcements.
Note 2: The switch ignores any information digits received.

—continued—

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Table 1-16
Send_To_Resource Play announcement and collect digits parameters (continued)

Parameter	Usage	Definition
		• AnnouncementBlock encoded as an AnnouncementDigitBlock – contains the following: — MaximumDigits – identifies the number of digits required – Fixed indicates that the switch should collect the specified number of digits. – Variable “UPTO” indicates that the switch should collect the number of digits specified in the range provided by the SCP. For example, collect 0 to 10 digits. – Variable indicates that the switch should collect 0 to 24 digits. – Normal number of digits indicates that the switch should collect an address using the normal dialing plan for the agent. — UninterAnnounceBlock – contains a Play Announcement tag and up to 3 uninterruptible, audible tone or announcement resource identifiers. The announcement elements contain a resource identifier (index into table CAINRSRC). — InterAnnounceBlock – contains a Play Announcement tag and up to 3 interruptible, audible tone or announcement resource identifiers. The announcement elements contain a resource identifier (index into table CAINRSRC).
Note 1: Announcements are played in the order received, uninterruptible announcements are played first, followed by the interruptible announcements. ***Note 2:*** The switch ignores any information digits received.		
—continued—		

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

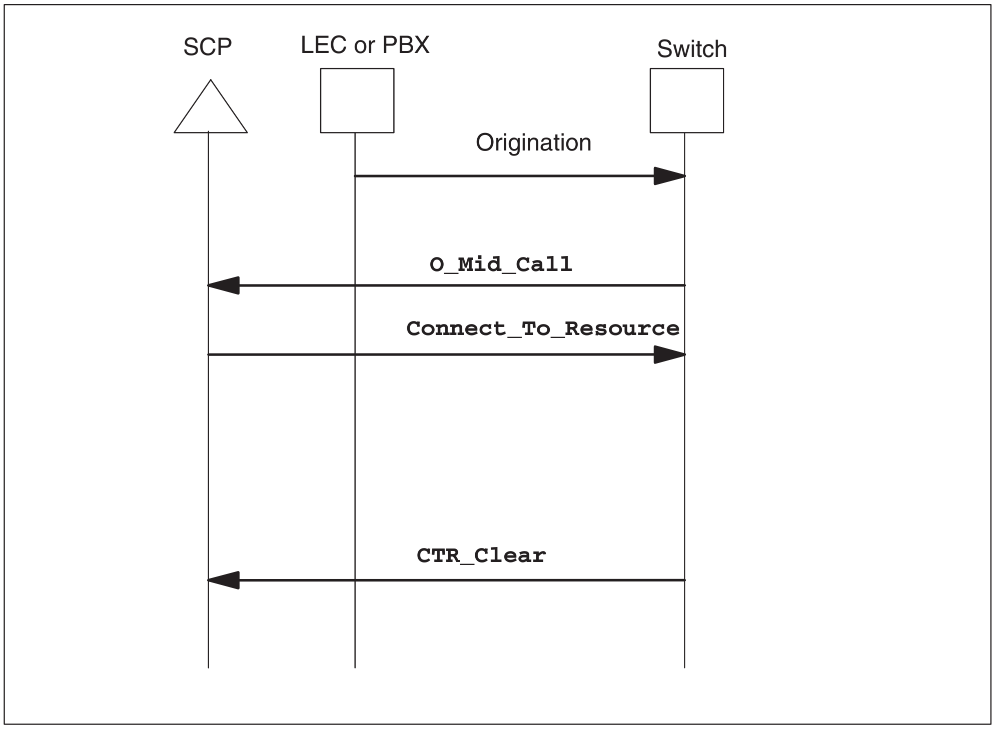
Table 1-16
Send_To_Resource Play announcement and collect digits parameters (continued)

Parameter	Usage	Definition
AnswerIndicator	Optional	Presence instructs the switch to provide answer supervision to the originating agent while the caller is connected to the resource. The switch sends answer indication to the caller in response to the Play Announcement request if answer indication has not already been sent. ***Note 1:*** ***AnswerIndicator*** is only used for SS7 and PRI originators. ***Note 2:*** ***AnswerIndicator*** does not affect billing (internal resources only) at the querying switch.
Note 1: Announcements are played in the order received, uninterruptible announcements are played first, followed by the interruptible announcements. ***Note 2:*** The switch ignores any information digits received.		
—end—		

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Figure 1-2 shows the interaction between the UCS DMS-250 switch and the SCP when the switch receives **`Connect_To_Resource`** message.

Figure 1-2
Connect_To_Resource conversation



Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Table 1-17 provides the play announcement and collect digits parameters for the `Connect_To_Resource` message.

Table 1-17
Connect_To_Resource Play announcement and collect digits parameters

Parameter	Usage	Definition
ResourceType	Required	Contains Play Announcement and Collect Digits
STRParameterBlock	Required	Contains a Play Announcement and Collect Digits tag and one announcement digit block (Note 1):

Note 1: Announcements are played in the order received, uninterruptible announcements are played first, followed by the interruptible announcements.
Note 2: The switch ignores any information digits received.
Note 3: For `Connect_To_Resource` on reorigination, the ***LegID*** parameter is always set to 0.

—continued—

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Table 1-17
Connect_To_Resource Play announcement and collect digits parameters (continued)

Parameter	Usage	Definition
		• AnnouncementBlock encoded as an AnnouncementDigitBlock – contains the following: — MaximumDigits – identifies the number of digits required – Fixed indicates that the switch should collect the specified number of digits. – Variable “UPTO” indicates that the switch should collect the number of digits specified in the range provided by the SCP. For example, collect 0 to 10 digits. – Variable indicates that the switch should collect 0 to 24 digits. – Normal number of digits indicates that the switch should collect an address using the normal dialing plan for the agent. — UninterAnnounceBlock – contains a Play Announcement tag and up to 3 uninterruptible, audible tone or announcement resource identifiers. The announcement elements contain a resource identifier (index into table CAINRSRC). — InterAnnounceBlock – contains a Play Announcement tag and up to 3 interruptible, audible tone or announcement resource identifiers. The announcement elements contain a resource identifier (index into table CAINRSRC).
`LegID`	Optional	0 (controlling leg) or 1 (passive leg); instructs the switch to connect the controlling leg (calling party) or passive leg (called party) to a resource. (Note 3)

Note 1: Announcements are played in the order received, uninterruptible announcements are played first, followed by the interruptible announcements.
Note 2: The switch ignores any information digits received.
Note 3: For **`Connect_To_Resource`** on reorigination, the ***`LegID`*** parameter is always set to 0.

—end—

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Normal Digit Collection

When a **Send_To_Resource** or **Connect_To_Resource** message is received with an indication of "normal number of digits to collect" in the ***StrParameterBlock*** parameter, it is a signal to the UCS DMS-250 switch to collect an address using the normal dialing plan for the agent. The current pretranslator name for the call is used (this could be the pretranslator name idicated by the pretranslatorName extension parameter). In this case, NetworkBuilder will send the dialed digits with a corresponding nature of address in a ***CollectedAddressInfo*** parameter in a **Resource_Clear** or **CTR_Clear** message to the SCP. The nature of address is set according to the dialing plan. For more information on the ***CollectedAddressInfo*** parameter, refer to Volume 3, Chapter 8, "Outgoing CAIN message parameters."

Note: The ***CollectedAddressInfo*** and ***CollectedDigits*** parameters are mutually exclusive, and are never included in the same message.

It is recommended that the SCP return the address collected through "normal digit collection" to the UCS DMS-250 switch in the ***CalledPartyID*** parameter of the **Analyze_Route** message. This is due to the fact that the collected address has gone through translations and the call's data, including the translations type, has been updated to reflect the new address for the call.

It is recommended that the SCP returns a pretranslatorName extension parameter with a pretranslator defined in the STDPRTCT table that allows access to the operator by dialing 0. Otherwise, CAIN may determine that the operator is requested and send back digits (or no digits) with the corresponding nature of address, but the ***ClearCause*** would be invalidCode if the number does not pretranslate.

When collecting the address digits for the "normal digit collection," the normal interdigit timing for the dialing plan is used before and after the first digit is received. Dialing ends when any one of the following conditions occurs:

- the dialing plan is complete
- an octothorpe (#) is dialed (though not as the first digit)
- a timeout occurs

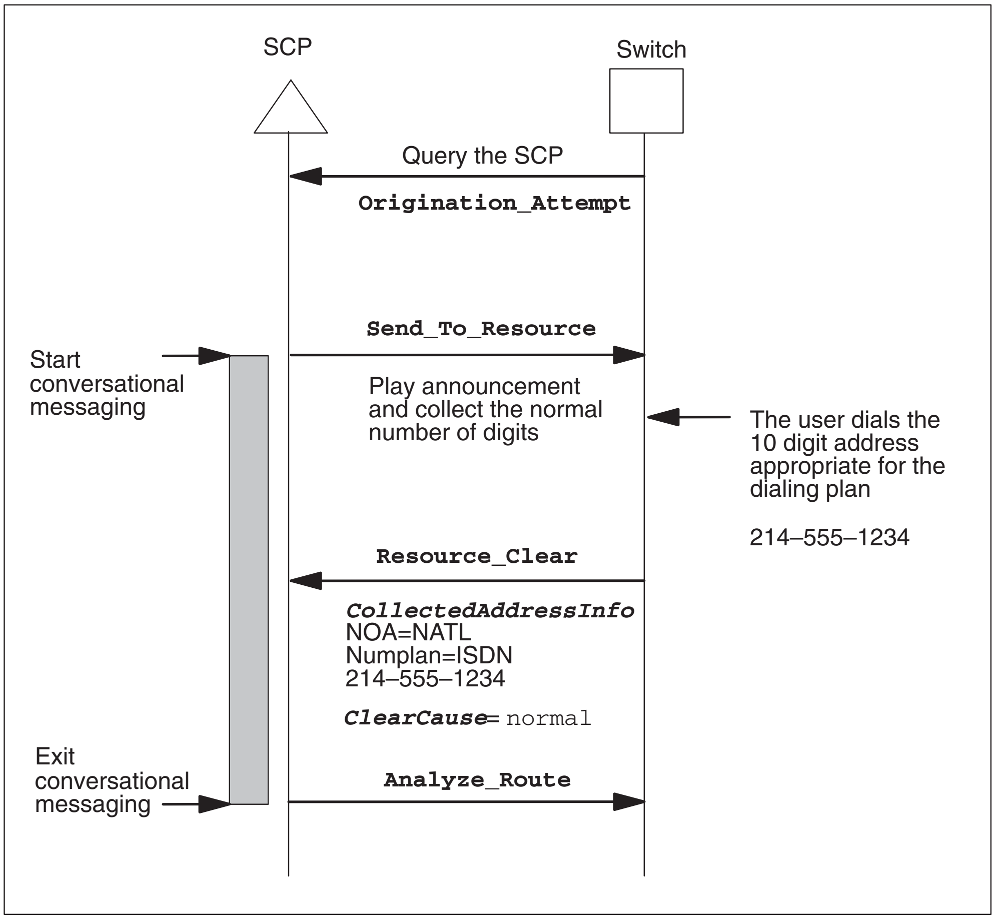
When the dialing plan is completed, the ***ClearCause*** parameter is set to normal. When a timeout occurs, not enough digits are dialed prior to the octothorpe, or the digits dialed are unrecognizable, the ***ClearCause***

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

parameter is set to `invalidCode`. Figure 1-3 gives an example of this scenario. Refer to Volume 3, Chapter 8, "Outgoing CAIN message parameters," for further information on the ***`ClearCause`*** parameter.

Note: When either an octothorpe (#) or an asterisk (*) is dialed as the first digit, the address goes through the appropriate pretranslator in table STDPRTCT (OCTR and ASTR, respectively). For more information on table STDPRTCT, refer to the *UCS DMS-250 Data Schema Reference Manual*.

Figure 1-3
"Normal digit collection" and normal dialing plan completion in conversation

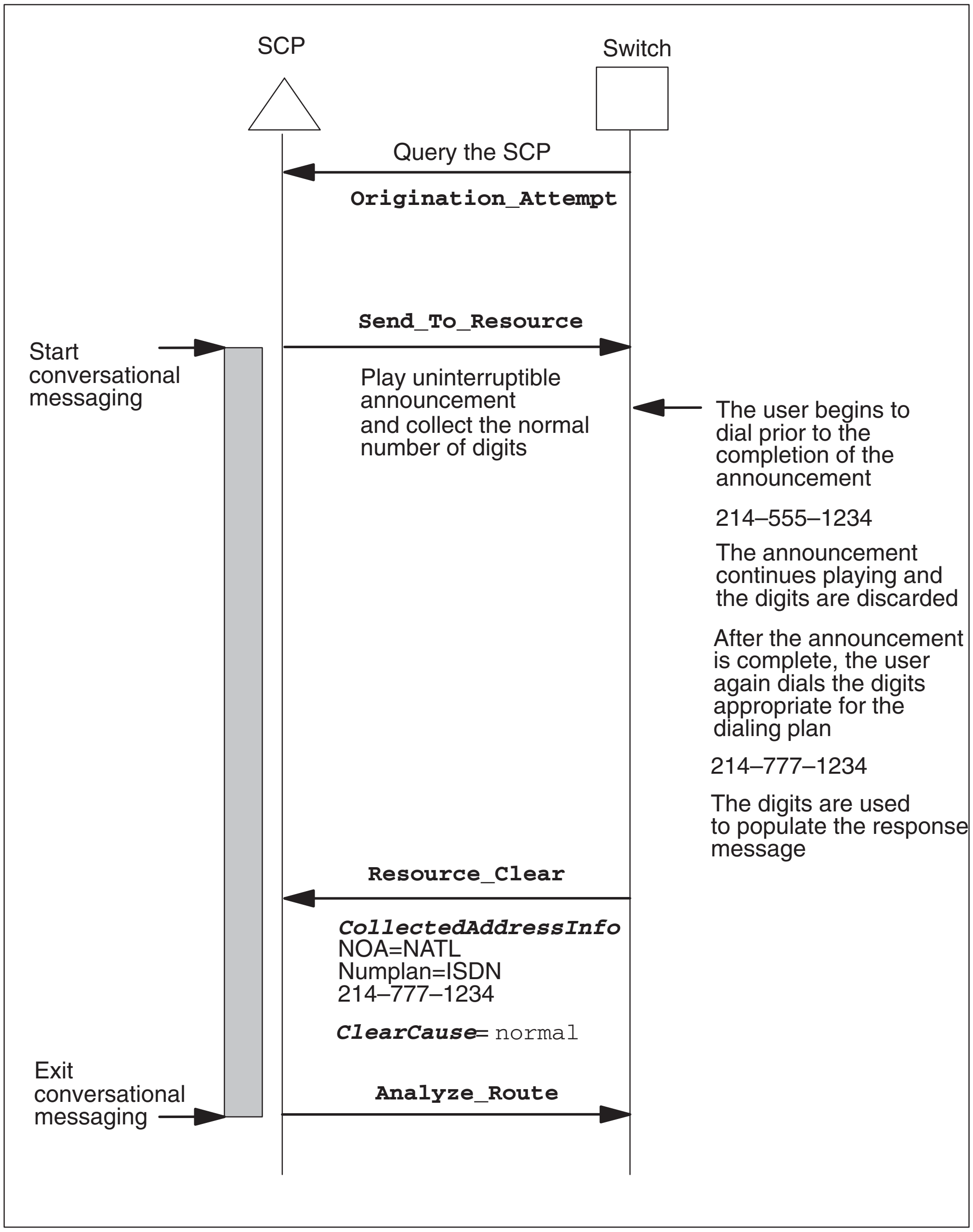

SCP
Switch
Query the SCP
Origination_Attempt
Send_To_Resource
Start conversational messaging
Play announcement and collect the normal number of digits
The user dials the 10 digit address appropriate for the dialing plan
214–555–1234
Resource_Clear
CollectedAddressInfo
NOA=NATL
Numplan=ISDN
214–555–1234
ClearCause= normal
Exit conversational messaging
Analyze_Route

When collecting the "normal number of digits," the digit collection is subject to the buffering rules of the User Interaction Framework (UIF). If the ***`StrParameterBlock`*** parameter indicates that the announcement is to be uninterruptible, then the "normal number of digits" will be collected after

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

the announcement has completed. Any digits entered prior to the completion of the announcement will be discarded. Figure 1-4 gives an example of this scenario.

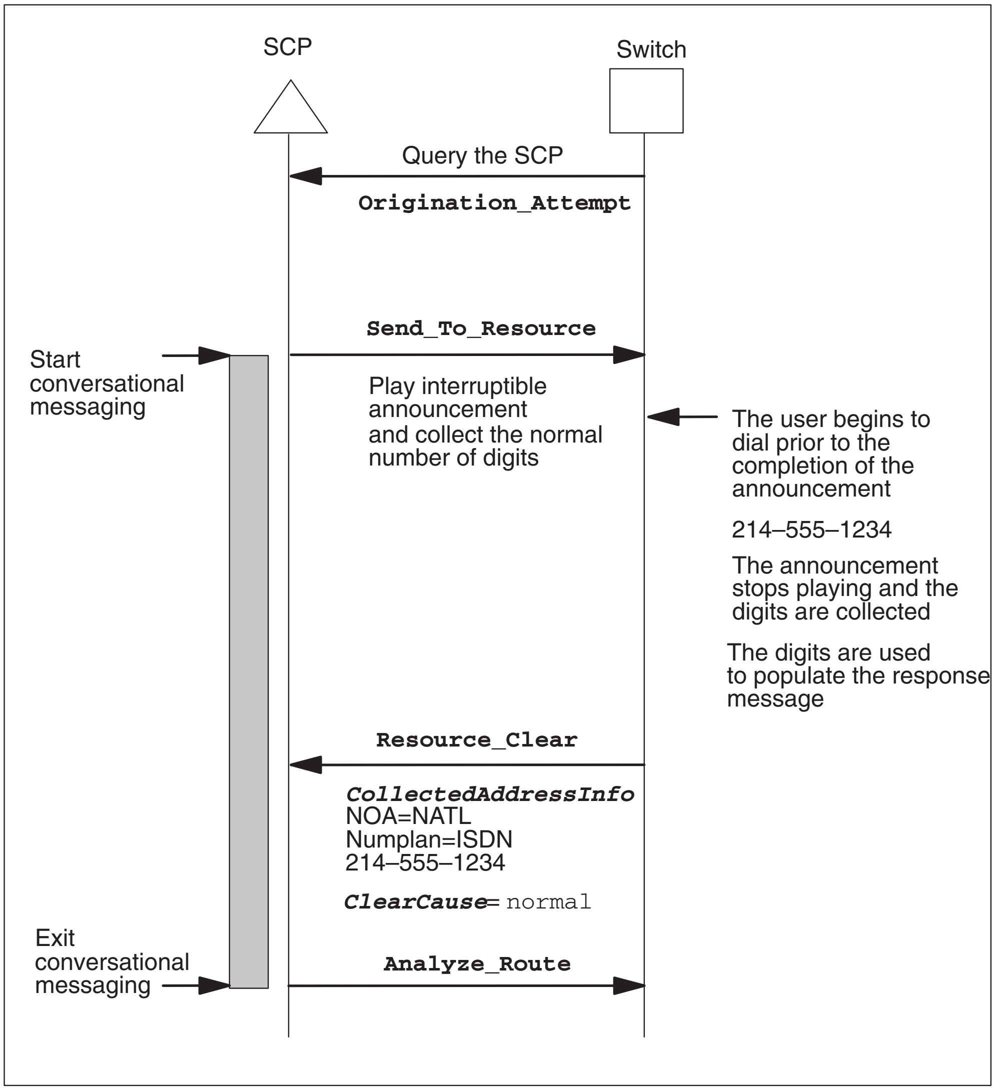
Figure 1-4
"Normal digit collection" and uninterruptible announcement in conversation


SCP
Switch
Query the SCP
Origination_Attempt
Send_To_Resource
Start conversational messaging
Play uninterruptible announcement and collect the normal number of digits
The user begins to dial prior to the completion of the announcement
214–555–1234
The announcement continues playing and the digits are discarded
After the announcement is complete, the user again dials the digits appropriate for the dialing plan
214–777–1234
The digits are used to populate the response message
Resource_Clear
CollectedAddressInfo
NOA=NATL
Numplan=ISDN
214–777–1234
ClearCause= normal
Exit conversational messaging
Analyze_Route

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

If the ***StrParameterBlock*** parameter indicates that the announcement is to be interruptible, then the “normal number of digits” are to be collected whenever the user begins dialing. Any digits entered prior to the completion of the announcement will cause the announcement to stop playing and the digits will be buffered for use. Figure 1-5 gives an example of this scenario.

Figure 1-5
“Normal digit collection” and interruptible announcement with buffering in conversation


SCP
Switch
Query the SCP
Origination_Attempt
Send_To_Resource
Start conversational messaging
Play interruptible announcement and collect the normal number of digits
The user begins to dial prior to the completion of the announcement
214–555–1234
The announcement stops playing and the digits are collected
The digits are used to populate the response message
Resource_Clear
CollectedAddressInfo
NOA=NATL
Numplan=ISDN
214–555–1234
ClearCause= normal
Exit conversational messaging
Analyze_Route

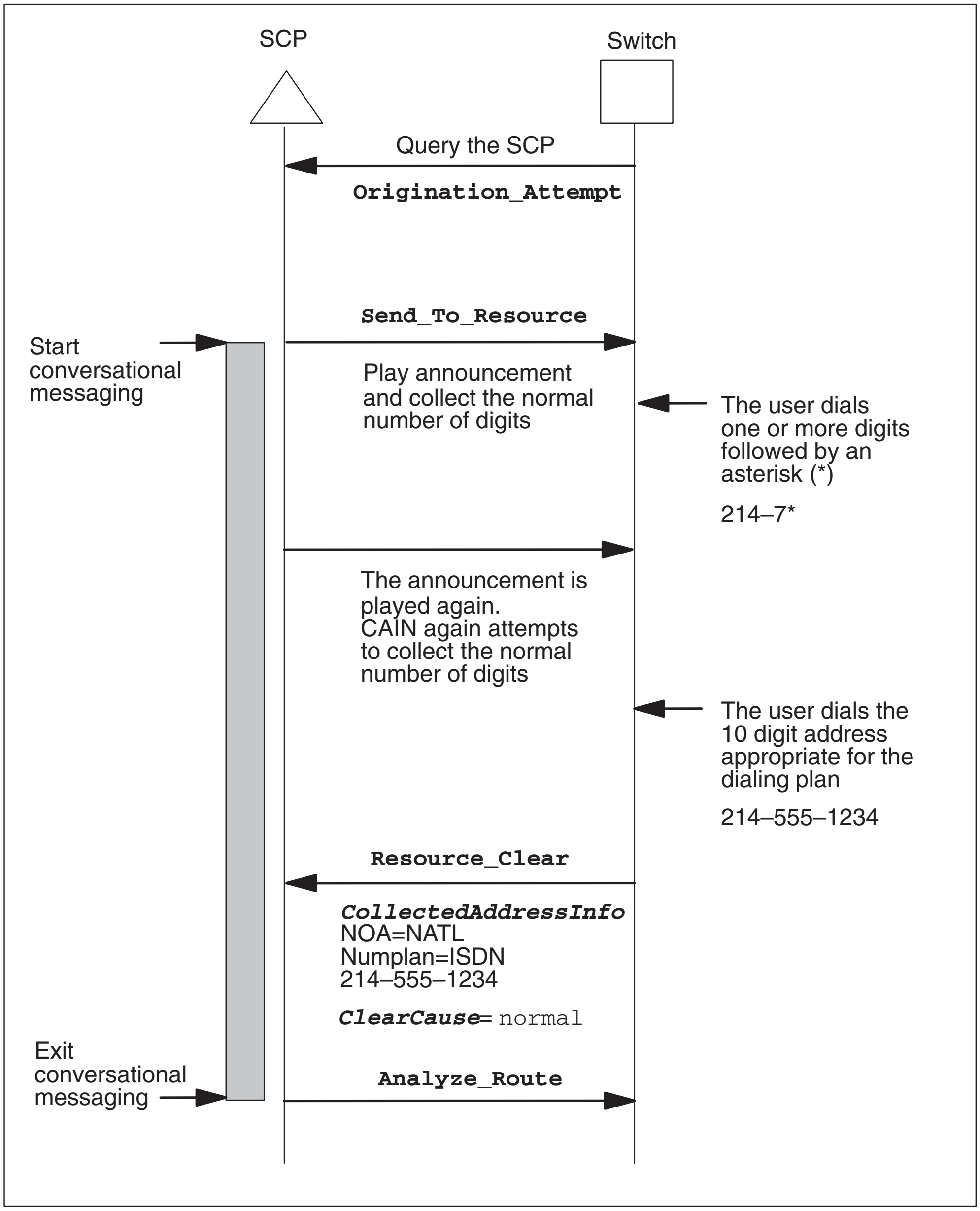
Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

If an asterisk (*) is received following one or more digits during the digit collection for “normal digits collection,” it is a signal to restart digit collection. At this point, the announcement is replayed, and digit collection is restarted. Figure 1-6 gives an example of this scenario.

Note: When either an octothorpe (#) or an asterisk (*) is dialed as the first digit, the address goes through the appropriate pretranslator in table STDPRTCT (OCTR and ASTR, respectively). For more information on table STDPRTCT, refer to the *UCS DMS-250 Data Schema Reference Manual*.

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Figure 1-6
“Normal digit collection” and an asterisk is received in conversation


SCP
Switch
Query the SCP
Origination_Attempt
Send_To_Resource
Start conversational messaging
Play announcement and collect the normal number of digits
The user dials one or more digits followed by an asterisk (*)
214–7*
The announcement is played again.
CAIN again attempts to collect the normal number of digits
The user dials the 10 digit address appropriate for the dialing plan
214–555–1234
Resource_Clear
CollectedAddressInfo
NOA=NATL
Numplan=ISDN
214–555–1234
ClearCause= normal
Exit conversational messaging
Analyze_Route

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

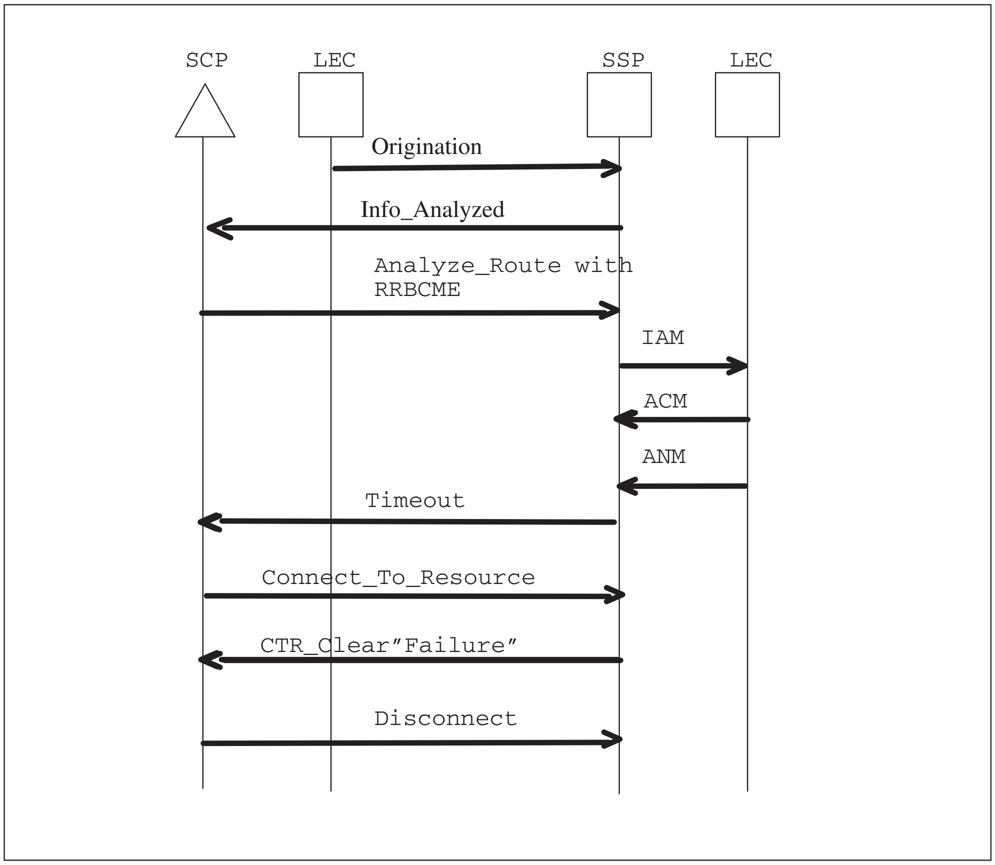
Caller interactions for Connect_To_Resource

The UCS DMS-250 switch procedures for processing a **`Connect_To_Resource`** message within a Conversation package are based on the procedures defined in *GR–1298–CORE*. The connection can be made to a single leg of the call, either originating or terminating, or to the entire call.

If the UCS DMS-250 switch cannot play the announcement or collect digits due to the unavailability or failure of switch hardware or switch resources, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `failure` and a ***`FailureCause`*** of `unavailableResources`. Figure 1-7 gives an example of this scenario.

Figure 1-7
Unavailability or failure of switch hardware or switch resources



Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

If the requested resource is not installed or implemented on the UCS DMS-250 switch, then it is treated by the switch as follows:

- A **CTR_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of `abort`.
- The call is not cleared toward the controlling leg or the passive leg.

The UCS DMS-250 switch plays the designated resource to the calling party or entire connection segment specified by the ***LegID*** and collects user dialed digits (if required).

- If the SCP requests that a non-interruptible resource be played and zero digits be collected, the switch plays the resource to the entire connection if no ***LegID*** is provided or to the specified ***LegID*** and, when the resource has ended, the following action is performed:
    - — A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `normal`.
- If the SCP requests that a non-interruptible resource be played and a non-zero number of digits be collected, the UCS DMS-250 switch discards any digits received while the resource is being played. When the resource has ended, the switch begins digit collection.
- If the SCP requests that an interruptible resource be played and a non-zero number of digits be collected, the UCS DMS-250 switch is prepared to start digit collection while the resource is being played and stops playing the resource as soon as the party specified by the ***LegID*** dials a digit. If no digits are received while the resource is being played, when the resource has ended, the UCS DMS-250 switch continues to wait for digits from the user or until permanent signal timeout occurs. If no ***LegID*** is specified, all parties hear the resource being played, but only the calling party can enter digits. Both parties hear the digits being dialed.
- During digit collection the following apply:
    - — Digit information can be requested by the SCP from the subscriber. The context of the digits are not known to the UCS DMS-250 switch, meaning the switch only knows that the SCP has requested a specific or variable number of digits be collected from the user. If the first digit is either an octothorpe (#) or an asterisk (*), the switch will treat it as a regular digit (that is, the digit will be included as a collected digit in the stream).

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

- Reset Dialing allows the originating user the option to reset when collecting digits. The subscriber can start over at the beginning after pressing the defined reset digit. If the asterisk digit is collected after the first digit, the UCS DMS-250 switch treats it as a signal to reset dialing (this only resets the current collection stream, any previously collected digit streams are not affected).
  - Carrier-AIN users may dial * to reset dialing to the beginning of collection after dialing at least one digit. After the * is dialed, any applicable resource prompt is played again.
  - The maximum number resets which are allowed to each user interaction for digit collection is datafilled in table CAINPARM. Tuple CAIN_STR_RESETS_ALLOWED {0 to 255} indicates the number of times (if any) the user may reset dialing during a **`Connect_To_Resource`** digit collection interaction. Refer to the *UCS DMS-250 Data Schema Reference Manual* for more information on this CAIN parameter. Upon exceeding the maximum number of resets during digit collection, the switch sends a **`CTR_Clear`** in a conversation package with a ***`ClearCause`*** parameter set to `invalidCode`.
  - When the switch detects that invalid information has been received (either due to a digit collection timeout or invalid digits received), the switch sends any collected information in a **`CTR_Clear`** message in a conversation package with a ***`ClearCause`*** parameter set to `invalidCode`.
- End Of Dialing allows the originating user the option to signal the end of dialing when collecting digits. If the user dials the octathorpe after the first digit, the switch treats it as an “end of digits” signal.
- Interdigit Timing (during digit collection) depends on the type of digit string requested by the SCP:
  - If the SCP requested a variable number of digits, the switch uses normal interdigit timing before and after the first digit is received, to determine end of dialing. The switch considers end of dialing to be when the required number of digits are received, a timeout occurs, or when the subscriber dials #.
  - If the SCP requested a fixed number of digits from 1 to 24, the switch uses normal interdigit timing before and after the first digit is received. The switch considers end of dialing to be when a timeout occurs, when the requested number of digits are collected, or when the subscriber dials #.

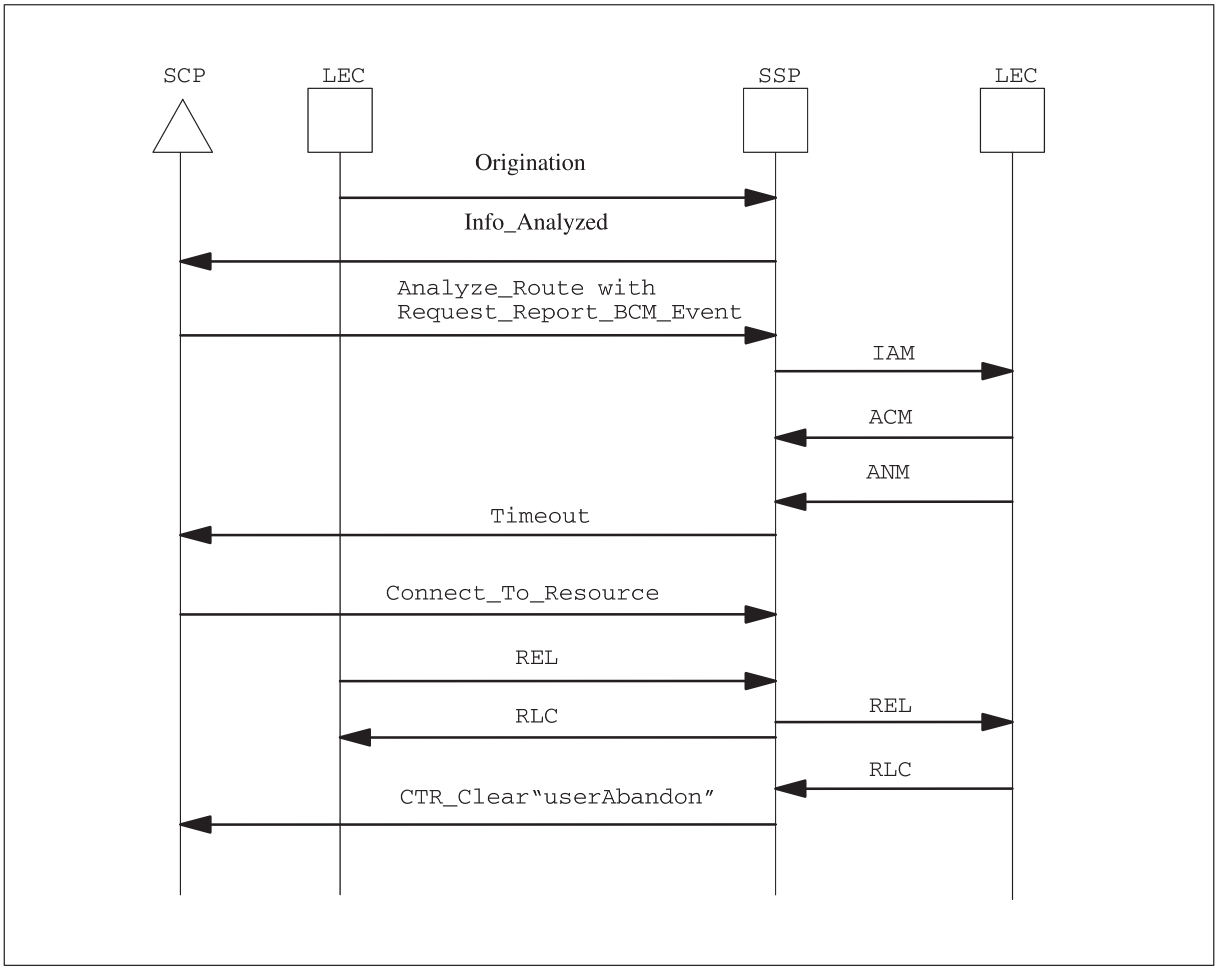
Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

- If a ***LegID*** is specified and the user on the leg abandons or is disconnected in some other way before the **Connect_To_Resource** is complete:
  - — The UCS DMS-250 switch cancels the resource interaction.
  - — The UCS DMS-250 switch sends a **CTR_Clear** in a response package with a ***ClearCause*** parameter set to userAbandon.
- If a ***LegID*** is specified and a non-specified leg abandons the call, the following actions are performed:
  - — The UCS DMS-250 switch continues the resource interaction.
  - — Upon completion of the resource interaction the UCS DMS-250 switch sends a **CTR_Clear** in a response package with a ***ClearCause*** parameter set to userAbandon.
- If no ***LegID*** is specified and all users abandon or are disconnected while prompting for digits, the following actions are performed:
  - — The switch cancels the resource interaction
  - — The switch sends a **CTR_Clear** in a response package with a ***ClearCause*** parameter set to userAbandon.
- If a ***LegID*** is specified and a non-specified leg abandons the call or is disconnected while prompting for digits, the following actions are performed:
  - — The UCS DMS-250 switch cancels the resource interaction
  - — The UCS DMS-250 switch sends a **CTR_Clear** in a response package with a ***ClearCause*** parameter set to userAbandon.
- If a ***LegID*** is specified and a non-specified leg abandons the call or is disconnected after prompting for digits is complete, the following actions are performed:
  - — The UCS DMS-250 switch cancels the resource interaction
  - — The UCS DMS-250 switch sends a **CTR_Clear** in a response package with a ***ClearCause*** parameter set to userAbandon.

Please refer to Figure 1-8, “LegID user abandons,” and Figure 1-9 , “All users abandon.”

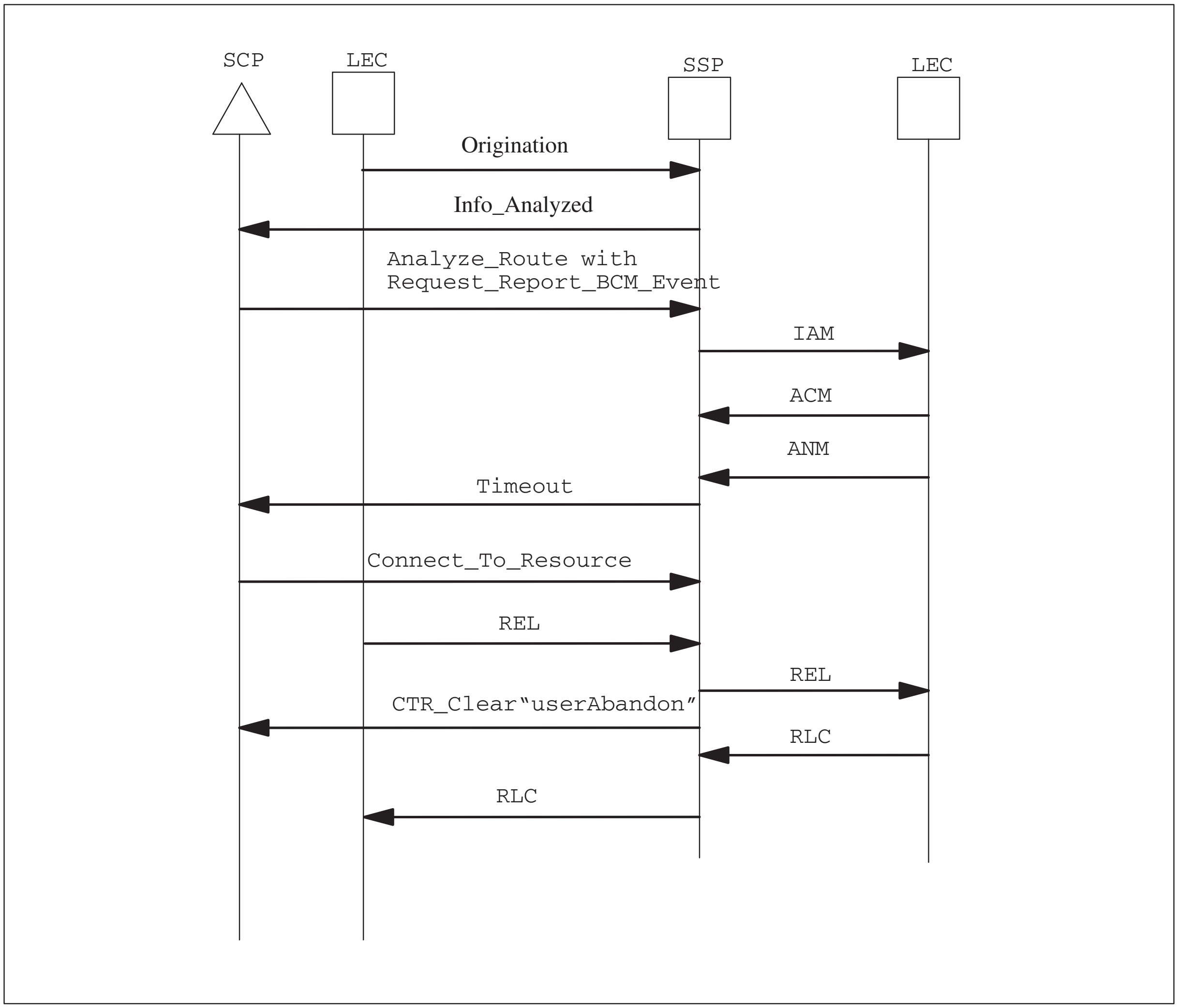
Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Figure 1-8
LegID user abandons



Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Figure 1-9
All users abandon



Fatal application errors

Fatal application errors occur when CAIN call processing is unable to continue due to an unexpected error. Table 1-18 describes the errors that can occur after an SCP returns a response.

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (continued)

Table 1-18
Send_To_Resource and Connect_To_Resource fatal application errors

Error type	Package	Log generated	Reported to SCP?	Action performed
DestinationAddress parameter included	Response	CAIN200	Yes	Switch applies AIND
ResourceType set to Play Announcement and Collect Digits in response package (for **Connect_To_Resource**)	Response	CAIN200	Yes	The switch sends a **CTR_Clear** message to the SCP with a ***ClearCause*** parameter value of abort; the call is not cleared toward the controlling leg or the passive leg.
ResourceType set to Play Announcement and Collect Digits in response package (for **Send_To_Resource**)	Response	CAIN200	Yes	Switch applies AIND
ResourceType set to an unexpected value (for **Connect_To_Resource**)	Response or Conversation	CAIN200	Yes	The switch sends a **CTR_Clear** message to the SCP with a ***ClearCause*** parameter value of abort; the call is not cleared toward the controlling leg or the passive leg.
Play Announcement and Collect Digits in response package with ***DisconnectFlag***	Response	CAIN200	Yes	Switch applies AIND
Play Announcement and Collect Digits in conversation package with ***DisconnectFlag***	Conversation	CAIN200	Yes	Switch applies AINF

Nonfatal application errors

Table 1-19 describes errors that can occur while the **Send_To_Resource** or **Connect_To_Resource** parameters are being processed.

Table 1-19
Send_To_Resource and Connect_To_Resource nonfatal application errors

Error type	Package	Log generated	Reported to SCP?	Action performed
Missing ***DisconnectFlag***	Response	CAIN100	No	log is generated and the interaction continues
Play Announcements in conversation package	Conversation	CAIN100	Yes	The switch processes the message as a Play Announcement request, and the switch applies AIND
Play Announcements in conversation package with ***DisconnectFlag***	Conversation	CAIN100	Yes	Switch applies AIND
ResourceType set to Play Announcement and Collect Digits, and ***STRParameterBlock*** set to any value other than Play Announcement and Collect Digits	Conversation	CAIN100		
treatment ***ExtensionParameter*** present	Conversation	CAIN100	No	Interaction proceeds
Invalid treatment ***ExtensionParameter*** received	Response Conversation	CAIN101	No	Interaction proceeds
SOC for extension parameters (CAIN0200) is idle	Response	CAIN102	No	Extension parameters are ignored

Limitations and restrictions

- Information digits are ignored.
- The UCS DMS-250 switch only expects one announcement block.

Associated logs

CAIN100, CAIN101, CAIN102, CAIN200

Send_To_Resource and Connect_To_Resource Play Announcement and Collect Digits (end)

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF

STR-Connections

In AIN 0.2, the Intelligent Peripheral (IP) was introduced as a component of the AIN Architecture. The IP contains functionality and resources capable of exchanging information with a subscriber, such as:

- playing pre-recorded announcements or music
- collecting dual tone multi-frequency (DTMF) digits
- recording voice or modulated voice information
- playing recorded voice or modulated voice information
- performing speaker-dependent or speaker-independent voice recognition

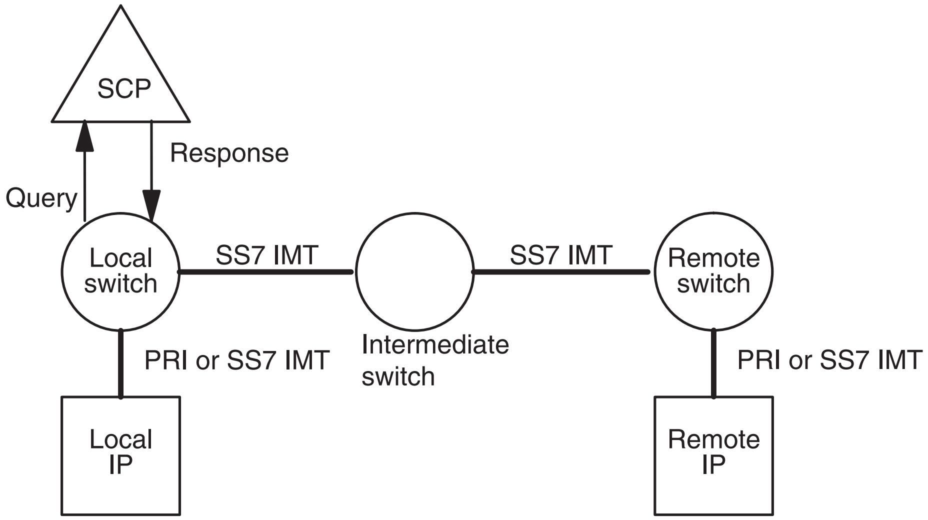
AIN 0.2 supports two types of connections to an IP:

- a STR-or CTR-Connection (an IP connection in response to a **`Send_To_Resource`** message is referred to as a STR-Connection; an IP connection in response to a **`Connect_To_Resource`** message is referred to as a CTR-Connection)
- a connection resulting from a normal termination attempt (for example, the subscriber dials an address served by an IP), referred to as a termination

Terminology

Figure 2-1 shows a network diagram and the terminology used in association with an STR-Connection.

Figure 2-1
Network diagram with an IP



Local switch — The switch where a TDP was encountered and trigger criteria met. The switch queries the SCP for data relating to the processing of the call.

Local IP — An IP with a direct ISDN or SS7 connection to the local switch. The IP is accessed when the SCP requests an STR-Connection from the local switch.

Intermediate switch — A tandem switch used to complete the connection between a local switch and a remote switch, when a direct connection between the local and remote switch is not available.

Remote switch — A tandem switch used to complete the connection between a local switch and a remote IP. A remote switch is used when the local switch does not have a direct ISDN or SS7 connection to the desired IP.

Remote IP — An IP that does not have a direct ISDN or SS7 connection to the local switch. The local switch must connect to another switch that has a direct ISDN or SS7 connection to the IP.

First Leg — During an STR-Connection, the local switch establishes a connection to the IP. This is known as the first leg of the call.

Second Leg — Following an STR-Connection to an IP, the local switch normally sends a **`Resource_Clear`** to the SCP in a conversation package. At this point, the SCP may provide additional routing information (through an **`Analyze_Route`** message) so that a connection is made to another party. This connection to the second party is referred to as the second leg of the call.

STR-Connection parameters

Although the ***DestinationAddress*** parameter is listed as an optional parameter for the **Send_To_Resource** message, this discussion of STR-Connections assumes the ***DestinationAddress*** parameter is present.

Table 2-1 provides the STR-Connections parameters for the **Send_To_Resource** message.

Note: Before UCS08, information that the ***AMAMeasure*** parameter currently supplies was supplied by the ***AnswerIndicator*** parameter.

Table 2-1
STR-Connection parameters

Parameter	Usage	Definition
ResourceType	Required	The CAIN framework ignores the contents of this parameter. However, any data passed to the UCS DMS-250 switch must be properly formatted and encoded to ensure proper decoding at the IP.
StrParameterBlock	Required	The CAIN framework ignores the contents of this parameter. However, any data passed to the UCS DMS-250 switch must be properly formatted and encoded to ensure proper decoding at the IP.
AnswerIndicator	Optional	This parameter instructs the UCS DMS-250 switch to provide answer supervision to the originating agent while the caller is connected to the resource. The UCS DMS-250 switch sends answer indication to the caller if answer indication has not already been sent. ***Note:*** **AnswerIndicator** is only used for SS7 and PRI originators.
AMADigitsDialedWC	Optional	This parameter contains digit strings to be populated into one or more of the following CDR fields: PINDIGS, ACCTCD, BILLNUM, CIC, ORIGPVN, or TERMPVN

—continued—

Table 2-1
STR-Connection parameters (continued)

Parameter	Usage	Definition
DestinationAddress	Optional	This parameter contains the address of the intelligent peripheral. This address is used with an STS to identify a route index for the STR-Connection. The UCS DMS-250 switch uses an STS from one of the following: • servTranslationScheme extension parameter in the **Send_To_Resource** message • default STS provisioned in table CAINXDFT • STS identified through existing call processing software (for example, table PARTOSTS)
AMAMeasure	Optional	When present with the value of connectTimeRecordedSSP, this parameter indicates that the call duration (CALLDUR field of the CDR) and includes the time connected to an IP. This parameter has no affect on answer indication. The ANSTYPE field of the CDR is updated to indicate early billing (prior to the second call leg answering). If the value is connectTimeRecordedDestinationSCP or connectTimeNotRecorded, no timing is begun. If more than one **Send_To_Resource** or **Connect_To_Resource** is sent during a single call, subsequent ***AMAMeasure*** parameters will not reset nor stop timing, but will start timing if it has not already begun. ***Note:*** If the ***AMAMeasure*** parameter is received with a value other than connectTimeRecordedSSP, it is treated as if the parameter is not present.
ExtensionParameter		Extension parameters require the CAIN0200 SOC option.
servTranslation Scheme	Optional	This extension parameter contains a serving translation scheme.
billSequence Number	Optional	This extension parameter contains 32 bits of SCP-defined billing data that is stored in the CDR.
strConnection Type	Optional	This extension parameter indicates the type of connection protocol (IPI) to be used to establish communication between the SCP, UCS DMS-250 switch, and an IP resource.

—end—

Connectivity to an IP

IP connectivity defined by Bellcore's *GR-1129-CORE* specifies that only ISDN signaling is allowed for connections between a UCS DMS-250 switch (local or remote) and an IP. It only supports SS7 signaling for connections between local and remote switches. However, NetworkBuilder expands upon the IP connectivity defined by Bellcore's *GR-1129-CORE* allowing direct SS7 connectivity to an IP in addition to direct ISDN connectivity.

Timer TDISC

Timer TDISC provides a maximum time limit for the IP to respond to a FAR message with the `cancelIPResource` operation. The TDISC_TIMER field in table CAINPARM determines the maximum time duration for the TDISC timer.

- Range – 1 to 4 seconds
- Default – The timer will have a default value of 4 seconds.

Timer TSTRC

Timer TSTRC provides a maximum time limit for a STR-Connection to an IP. It is started when the IP answers and canceled when the STR-Connection is cleared, either by the UCS DMS-250 switch or IP. The TSTRC_TIMER field in table CAINPARM determines the time duration for the TSTRC timer.

- Range – 0 to 60 minutes, with the value of 0 disabling the timer
- Default – The timer will have a default value of 6 minutes.

SS7 connectivity

The IPTRUNK option in table TRKGRP is used on an ISUP IMT trunk to determine if the trunk is directly connected to an IP or connected to another switch. The presence of the IPTRUNK option indicates that the ISUP IMT trunk is directly connected to an IP, where absence of the IPTRUNK option indicates the trunk is connected to another switch. The IPTRUNK option is only available when the GRPTYPE field in table TRKGRP for the trunk is either IMT or PRA250.

The local switch normally provides the same functionality, regardless of whether an ISUP IMT trunk is used to connect to a local IP or remote switch. However, because the TDISC and TSTRC timers are maintained by the switch directly connected to the IP, these timers are started at the local switch when directly connecting to an IP on an IMT trunk that includes the IPTRUNK option. Otherwise, the timers are started at the remote switch that is directly connected to the IP.

Note: For the IMT trunk type it is also necessary for the ISUPIDX field in table TRKGRP to contain the value UCS2UCS.

ATTENTION

For an IP to use this functionality, the SS7 signaling requirements defined by Bellcore's *GR-1129-CORE* and those stated here must be followed.

PRI connectivity

The IPTRUNK option in table TRKGRP is used on a PRI trunk to determine whether or not the trunk is directly connected to an IP. The presence of the IPTRUNK option indicates that the PRI trunk is directly connected to an IP, where absence of the IPTRUNK option indicates the trunk is not connected to an IP. The IPTRUNK option is only available when the GRPTYPE field in table TRKGRP for the trunk is either IMT or PRA250.

Note: PRI trunks cannot be used as tandem trunks.

The local switch normally provides the same functionality, regardless of whether the PRI trunk includes the IPTRUNK option or not. However, because the TDISC and TSTRC timers are maintained by the switch directly connected to the IP, these timers are started at the local switch when directly connecting to an IP on a PRI trunk that includes the IPTRUNK option. Otherwise, the timers are not started.

Intelligent Peripheral Interface (IPI) overview

In a STR-Connection the UCS DMS-250 switch and IP communicate through an IPI. An IPI provides an interface that is met by both a switch and IP. NetworkBuilder supports the following IPIs:

- CONNECT_ONLY
- CONNECT_1129_STYLE

ATTENTION

The CONNECT_ONLY IPI was intended to provide initial support for connecting to an IP until the development of the CONNECT_1129_STYLE IPI could be completed.

Nortel Networks now recommends the use of the CONNECT_1129_STYLE IPI since it provides similar functionality with additional enhancements.

Although both the CONNECT_ONLY and CONNECT_1129_STYLE IPIs are based upon the IPI defined by Bellcore's *GR-1129-CORE*, there is a significant difference between the two. As outlined below, the CONNECT_ONLY IPI does not support the exchange of data between the SCP and IP through the UCS DMS-250 switch.

Note: The Global IMT SOC (CAIN0605) is not supported. If a **`Send_To_Resource`** message is received with a ***`DestinationAddress`*** parameter, a **`Resource_Clear`** message with a ***`ClearCause`*** of `taskRefused` is sent to the SCP.

IPI determination

The method for determining the IPI for a STR-Connection is derived through the following implied precedence order:

- `strConnectionType` extension parameter received in the **`Send_To_Resource`** message
- `strConnectionType` extension parameter in table CAINXDFT
- STR_CONNECTION_TYPE parameter in table CAINPARM

Note: If the IPI derived through the above precedence is NONE, a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `taskRefused`.

CONNECT_ONLY IPI

The CONNECT_ONLY IPI is intended for use when there is a direct communication link between the SCP and IP. Bellcore does not define this interface, nor does it prohibit an IP from directly communicating to an SCP or a Service Management System (SMS). It provides the following capabilities:

- Creates a connection between a subscriber and an IP through a voice channel in the UCS DMS-250 switch and allows an IP to use its internal resources and functionality to exchange information with that subscriber.
- Allows calls to originate and terminate over the IPI.

CONNECT_1129_STYLE IPI

The CONNECT_1129_STYLE IPI is based upon the IPI defined by Bellcore's *GR-1129-CORE*. It provides the following capabilities:

- Creates a connection between a subscriber and an IP through a voice channel in the UCS DMS-250 switch and allows an IP to use its internal resources and functionality to exchange information with that subscriber.
- Allows a UCS DMS-250 switch to interwork an exchange of data between an IP and the SCP.

- Allows calls to originate and terminate over the IPI.

Signaling

For the CONNECT_ONLY IPI, ISDN and SS7 signaling are supported for call establishment and call clearing only. Without a communication link between the SCP and IP, services requiring an exchange of data will not function correctly.

For the CONNECT_1129_STYLE IPI, ISDN and SS7 signaling are supported for call establishment and call clearing, and direct ISDN and SS7 connections between a switch (local or remote) and an IP are supported.

Note: The IPI defined by Bellcore's *GR-1129-CORE* specifies that only ISDN signaling is allowed for connections between a switch (local or remote) and an IP. It only supports SS7 signaling for connections between local and remote switches.

Signaling for either IPI involves:

- messaging between the UCS DMS-250 switch and SCP
- messaging between the UCS DMS-250 switch and IP
- messaging between switches for remote IP access

Signaling using the CONNECT_ONLY IPI

Carrier AIN STR-Connections with the CONNECT_ONLY IPI have limited data exchange capabilities as listed above. The following example of a CONNECT_ONLY illustrates the limitations:

1 A call triggers at a TDP. The UCS DMS-250 switch temporarily suspends call processing and launches a query to an SCP.

2 The SCP responds with a **Send_To_Resource** message containing a ***DestinationAddress*** parameter, which indicates an STR-Connection to an IP is needed.

3 The UCS DMS-250 switch translates the address contained in the ***DestinationAddress*** parameter to identify a route list for terminating to the IP.

4 The UCS DMS-250 switch initiates a connection to the IP by sending a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a remote IP).

Limitation: Bellcore specification *GR-1129-CORE* specifies that the information contained in the ***StrParameterBlock*** is passed to the IP using a Facility Information Element (FIE) for ISDN, or a Remote Operations (RO) parameter for SS7. The CONNECT_ONLY IPI does not support this interaction. Data in the ***StrParameterBlock*** is discarded.

5 The UCS DMS-250 switch establishes a connection with the IP.

Limitation: During an active connection to an IP, Bellcore's *GR-1129-CORE* states that data may be exchanged between the SCP and IP using a FACILITY message between the UCS DMS-250 switch and IP and a **Call_Info_From_Resource** or **Call_Info_To_Resource** message between the UCS DMS-250 switch and SCP. The CONNECT_ONLY IPI does not support this functionality.

6 Once the IP completes its function, it releases the call and the UCS DMS-250 switch sends a **Resource_Clear** message to the SCP and awaits further instructions.

Limitation: When the IP releases the call, Bellcore's *GR-1129-CORE* specifies that the data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***IPReturnBlock*** parameter of the **Resource_Clear** message. The CONNECT_ONLY IPI does not support this functionality.

Due to these limitations with the CONNECT_ONLY IPI, services requiring an exchange of data will not function correctly, unless there is direct communication between the SCP and IP.

Limitations and restrictions

For the CONNECT_ONLY IPI, STR-Connections have the following limitations and restrictions:

- During a CONNECT_ONLY STR-Connection, originating call model triggers on the local switch are not evaluated. Once the STR-Connection is completed (the switch sends a **Resource_Clear** message to the SCP), the originating call model triggers may again be evaluated.
- During an active CONNECT_ONLY STR-Connection, manual and auto reorigination is blocked. Once the STR-Connection is completed, reorigination is re-enabled when necessary.
- If a fatal application error is detected during a CONNECT_ONLY STR-Connection, final treatment is always applied. Default routing is not supported for fatal errors encountered during a STR-Connection.
- In UCS08, terminating call model triggers on the local switch are not evaluated.

Signaling using the CONNECT_1129_STYLE IPI

The following example illustrates Carrier AIN STR-Connections with the CONNECT_1129_STYLE IPI:

1 A call triggers at a TDP. The UCS DMS-250 switch temporarily suspends call processing and launches a query to an SCP.

2 The SCP responds with a **`Send_To_Resource`** message containing a ***`DestinationAddress`*** parameter, which indicates an STR-Connection to an IP is needed. In addition to the address of the IP, the SCP provides an ***`StrParameterBlock`*** which contains variable information needed by the IP, such as the requested resource and the function to be performed.

3 The UCS DMS-250 switch translates the address contained in the ***`DestinationAddress`*** parameter to identify a route list for terminating to the IP.

4 The UCS DMS-250 switch initiates a connection to the IP by sending a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a local or remote IP). The information contained in the ***`StrParameterBlock`*** is passed to the IP using a Facility Information Element (FIE) for ISDN, or a Remote Operations (RO) parameter for SS7.

5 The UCS DMS-250 switch establishes a connection with the IP. During an active connection to an IP, data may be exchanged between the SCP and IP using a FACILITY message between the UCS DMS-250 switch and IP and a **`Call_Info_From_Resource`** or **`Call_Info_To_Resource`** message between the UCS DMS-250 switch and SCP.

6 During an active connection to an IP, the SCP can request the UCS DMS-250 switch to terminate the connection between a subscriber and an IP by sending a **`Cancel_Resource_Event`** message to the UCS DMS-250 switch. The UCS DMS-250 switch notifies the IP of the termination request using a FACILITY message. The IP responds with a DISCONNECT message and releases the call. The data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***`IPReturnBlock`*** parameter of the **`Resource_Clear`** message.

7 Once the IP completes its function, it releases the call and the UCS DMS-250 switch sends a **`Resource_Clear`** message to the SCP and awaits further instructions. When the IP releases the call, the data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***`IPReturnBlock`*** parameter of the **`Resource_Clear`** message.

Component and operation type background

The exchange of messages between the UCS DMS-250 switch and the IP is based upon a simple request/reply exchange. Basically, the invoker (switch or the IP) requests a service from the performer (IP or the switch). After providing the service, the performer is expected to respond with either success or failure to the request from the invoker.

Components are used for the messaging exchange. A component may consist of a request to perform an operation at the remote end. A component may also indicate the success or failure of the requested operation. An operation indicates the service which is to be provided by the performer.

The following four components are used for 1129-style IP interaction:

- Invoke: This component is send by the invoker to initiate a service at a remote end (service performed by the performer).
- Return Result: This component is sent by the performer to indicate to the invoker that the requested service was performed correctly.
- Return Error: This component is sent by the performer to indicate to the invoker that the requested service could not be performed.
- Reject: This component is sent by either the invoker or performer to reject a received component.

For the SS7 protocol, these components are placed into a Remote Operations (RO) parameter; for the PRI protocol, these components are placed into a Facility Information Element (FIE). The formats for the RO parameter and FIE are shown in Figure 3-2 and Figure 3-3 respectively.

Figure 3-2
Remote Operation (RO) parameter format

bit 8	7	6	5	4	3	2	bit 1	octet
Parameter ID = 00110010								1
Parameter Length								2
Ext	Spare	Protocol Profile						3
Component								4 – n

Figure 3-3
Facility Information Element (FIE) format

bit 8	7	6	5	4	3	2	bit 1	octet
Facility Information Element Identifier = 00110010								1
Length of Information Element								2
Ext	Spare	Protocol Profile						3
Component								4 – n

The four components used for exchanging messages between a UCS DMS-250 switch and an IP are briefly described in the following four sections.

Invoke Component

The Invoke component is used to initiate a service at the remote end. The Invoke component contains a parameter which identifies the operation, and any additional parameters needed by the remote end in order to perform the requested service. Two operation parameters are supported in the Invoke component:

- `sendToIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 1}.
- `cancelIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 2}.

Return Result Component

When the requested service is performed successfully, the performer sends a Return Result component to the invoker. The Return Result component may contain parameters to be returned to the invoker. The operation value parameter is included only when parameters are present in the component. One operation is supported in the Return Result component:

- `sendToIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 1}.

Return Error Component

When the requested service can not be performed, the performer sends a Return Error component to the invoker. The Return Error component contains an Error Value which indicates the reason for failure. It may also contain a parameter which provides additional information regarding the error.

Reject Component

A Reject component is sent by either the invoker or performer to reject a received component (for example, Invoke). The components are rejected for such reasons as protocol violations, unrecognized components, or unrecognized parameters.

Initiating a STR-Connection to a Local or Remote IP

A **`Send_To_Resource`** message containing the ***`DestinationAddress`*** parameter is used to initiate an STR-Connection. When the **`Send_To_Resource`** message is received by the switch, the contents of the message are first validated.

If the ***`DestinationAddress`*** parameter is present in a message received in a response package, a fatal application error is detected. The following actions are performed:

- A CAIN200 Fatal Application Error log is generated with the REASON field set to “UNEXPECTED COMMUNICATION”, and final treatment is provided.
- An **`Application_Error`** message is reported to the SCP. The ***`ErrorCause`*** parameter is set to `unexpectedCommunication`.

When both the ***`DestinationAddress`*** and ***`DisconnectFlag`*** parameters are present in a **`Send_To_Resource`** message, a fatal application error is detected. The following actions are performed:

- A CAIN200 Fatal Application Error log is generated with the REASON field set to “ERRONEOUS DATA VALUE”, and final treatment is provided.
- An **`Application_Error`** message is reported to the SCP. The ***`ErrorCause`*** parameter is set to `erroneousDataValue`.

The contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are not validated by the UCS DMS-250 switch when connecting to an IP. However, the parameter sizes are verified to ensure that the maximum size limits are not exceeded. If the maximum size is exceeded, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.

DestinationAddress Parameter

The ***`DestinationAddress`*** parameter contains the address of an IP. The address is used in conjunction with a Serving Translation Scheme (STS) to identify a route index for the STR-Connection.

When the ***DestinationAddress*** parameter contains a nature of address (NOA) value other than National, International, or VPN, a non-fatal application error is detected. The following actions are performed:

- A CAIN100 Non-Fatal Application Error log is generated.
- A National NOA is assumed for the ***DestinationAddress*** parameter.

Once the contents of the parameter are validated, the local switch performs standard routing using the ***DestinationAddress*** parameter. The STS used by standard routing is determined as follows:

- The STS provided by the STS extension parameter received in a **Send_To_Resource** message.
- The STS provisioned in table CAINXDFT for the associated CAIN group.
- The STS identified thru existing call processing software (for example, table PARTOSTS).

If the local switch is unable to identify a route index, a fatal application is detected. The following actions are performed:

- A CAIN200 Fatal Application Error log is generated with the REASON field set to "ERRONEOUS DATA VALUE", and final treatment is provided.
- An **Application_Error** message is reported to the SCP. The ***ErrorCause*** parameter is set to `erroneousDataValue`.

CAIN_CONVERSATION_LIMIT

A **Send_To_Resource** message from the SCP initiates a conversation between the UCS DMS-250 switch and the SCP. Regardless of how the conversation is eventually terminated, the total number of conversations allowed on a single call is controlled by the CAIN_CONVERSATION_LIMIT parameter in table CAINPARM. This capability includes conversational **Send_To_Resource** messages containing the ***DestinationAddress*** parameter. The value set by this parameter can therefore serve to limit the number of repeated attempts to connect to an IP that are unsuccessful. The range of this office parameter is 0 to 15, where 0 is interpreted as no conversations allowed, and 15 is interpreted as unlimited conversations allowed. When the number of conversations on a single call exceeds the CAIN_CONVERSATION_LIMIT a fatal application error is encountered. A **Resource_Clear** message with a ***ClearCause*** of `taskRefused` is returned to the SCP in a response package and the call receives AINF treatment.

Establishing STR-Connections to a local IP

For CONNECT_ONLY STR-Connections, when the UCS DMS-250 switch receives a **`Send_To_Resource`** message (with a ***`DestinationAddress`*** parameter) from the SCP the following occurs:

- Switch call processing translates the ***`DestinationAddress`*** and determines a routing list to the appropriate IP.
- Once the local switch identifies a route index, the local switch attempts to establish a connection to the local IP. It is important to note that existing UCS DMS-250 software is used to establish the connection. Therefore, inswitch features may interact with the STR-Connection. Several examples are listed below:
  - — During the **Authorize_Termination** PIC, the UCS DMS-250 performs bearer capability screening on the terminating trunk using table BCCOMPAT. When bearer capability screening fails, the switch route advances to the next available trunk group in the route list.
  - — During the **Present_Call** PIC, the UCS DMS-250 constructs the ISDN SETUP message to be sent to the IP. Delivery of the Calling Party Information Element is controlled using the ANIDELV option of table CALLATTR.
- The local switch selects a trunk group from the route list and attempts to locate an idle trunk member within the group. If no idle trunk members are present, the local switch route advances to the next available trunk group in the route list.
- If the local switch is unable to locate an idle trunk member within the route list, a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** value of `channelsBusy`.
- If the selected trunk group is not a PRI or SS7 IMT trunk, or it is a PRI trunk without the IPTRUNK option provisioned, a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- Once an idle trunk member is located, the local switch constructs a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a local IP) and sends it to the local IP.
  - — For PRI terminations a Called Party Information Element is built using the address contained in the ***`DestinationAddress`*** parameter, and the contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are discarded.
  - — For SS7 IMT terminations a Called Party Number is built using the address contained in the ***`DestinationAddress`*** parameter, and the contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are discarded.

- It is important to note that the SETUP or IAM message is built using the existing call processing logic. Therefore, the contents of the SETUP or IAM message may be influenced by switch software such as table RTEATTR or through non-standard routing. Once the SETUP or IAM message is sent, the local switch waits for a response message from the local IP.
- When the long call duration timer is enabled for the terminating PRI or SS7 IMT agent, the timer is started using the value provisioned in table TRKGRP1. If the timer expires before an ISDN CONNECT or SS7 ANM message is received from the local IP, a **`Resource_Clear`** is sent to the SCP in a response package with a ***`ClearCause`*** value of `strCancelled`. The long call duration feature is activated to handle the expired timer.

For CONNECT_1129_STYLE STR-Connections, when the UCS DMS-250 switch receives a **`Send_To_Resource`** message which includes a ***`DestinationAddress`*** parameter and variable information, such as the requested resource and the function to be performed by the IP, the following occurs:

- Switch call processing translates the ***`DestinationAddress`*** and determines a routing list to the appropriate IP.
- Once the local switch identifies a route index, the local switch attempts to establish a connection to the local IP. It is important to note that existing UCS DMS-250 software is used to establish the connection. Therefore, inswitch features may interact with the STR-Connection. Several examples are listed below:
  - — During the **Authorize_Termination** PIC, the UCS DMS-250 performs bearer capability screening on the terminating trunk using table BCCOMPAT. When bearer capability screening fails, the switch route advances to the next available trunk group in the route list.
  - — During the **Present_Call** PIC, the UCS DMS-250 constructs the ISDN SETUP message to be sent to the IP. Delivery of the Calling Party Information Element is controlled using the ANIDELV option of table CALLATTR.
- The local switch selects a trunk group from the route list and attempts to locate an idle trunk member within the group. If no idle trunk members are present, the local switch route advances to the next available trunk group in the route list.
- If the local switch is unable to locate an idle trunk member within the route list, a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** value of `channelsBusy`.

- If the selected trunk group is not a PRI or SS7 IMT trunk, or it is a PRI trunk without the IPTRUNK option provisioned, a **Resource_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of `abort`.
- Once an idle trunk member is located, the local switch constructs a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a local IP) and sends it to the local IP.
- Once an idle trunk member is located, the local switch constructs a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a local IP) and sends it to the local IP. The Called Party Information Element is built using the address contained in the ***DestinationAddress*** parameter.
  - — For PRI terminations, a Called Party Information Element is built using the address contained in the ***DestinationAddress*** parameter, and the Facility Information Element contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the local switch.
  - — For SS7 IMT terminations, a Called Party Number is built using the address contained in the ***DestinationAddress*** parameter, and the Remote Operations (RO) parameter contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the local switch.
- It is important to note that the SETUP or IAM message is built using the existing call processing logic. Therefore, the contents of the SETUP or IAM message may be influenced by switch software such as table RTEATTR or through non-standard routing. Once the SETUP or IAM message is sent, the local switch waits for a response message from the local IP.
- When the long call duration timer is enabled for the terminating PRI or SS7 IMT agent, the timer is started using the value provisioned in table TRKGRP1. If the timer expires before an ISDN CONNECT or SS7 ANM message is received from the local IP, a **Resource_Clear** is sent in to the SCP in a response package with a ***ClearCause*** value of `strCancelled`. The long call duration feature is activated to handle the expired timer.

Messaging

This section describes the messages that may be received while establishing a STR-Connection to a local IP.

Figures 2-4 and 2-5 provide examples of call establishment to a local IP using ISDN signaling.

Figure 2-4
SS7 call establishment to a local IP using ISDN signaling

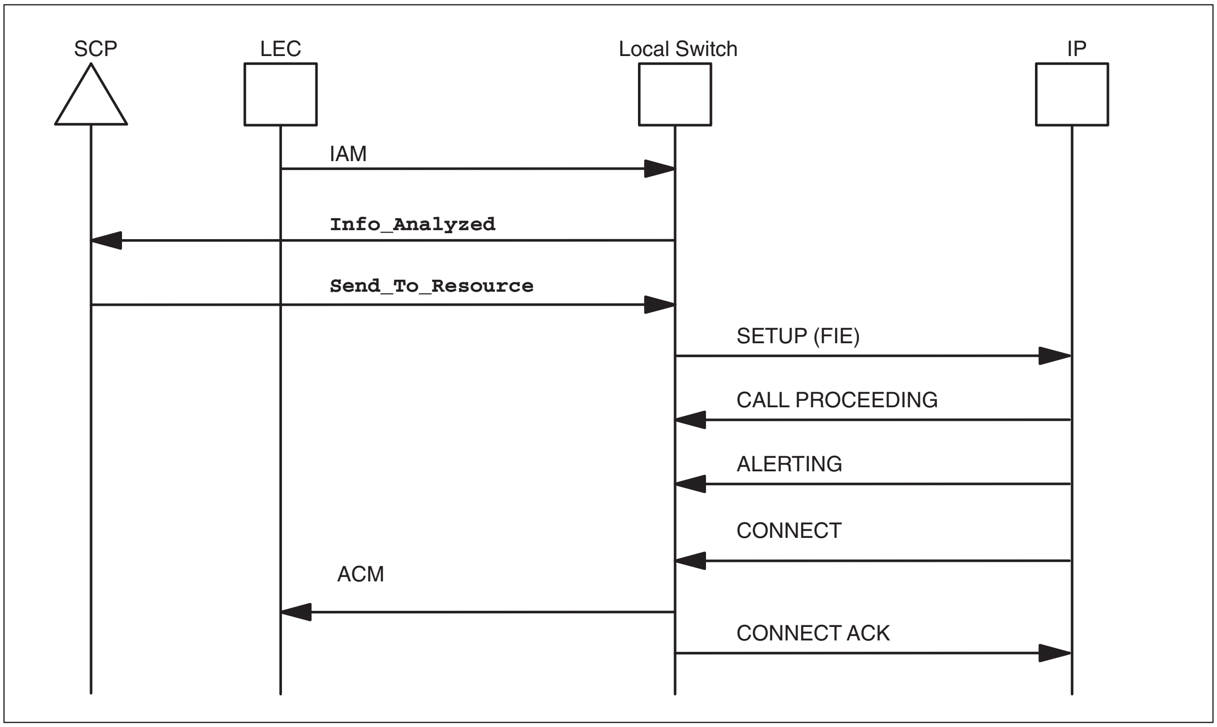


Figure 2-5
PRI call establishment to a local IP using ISDN signaling

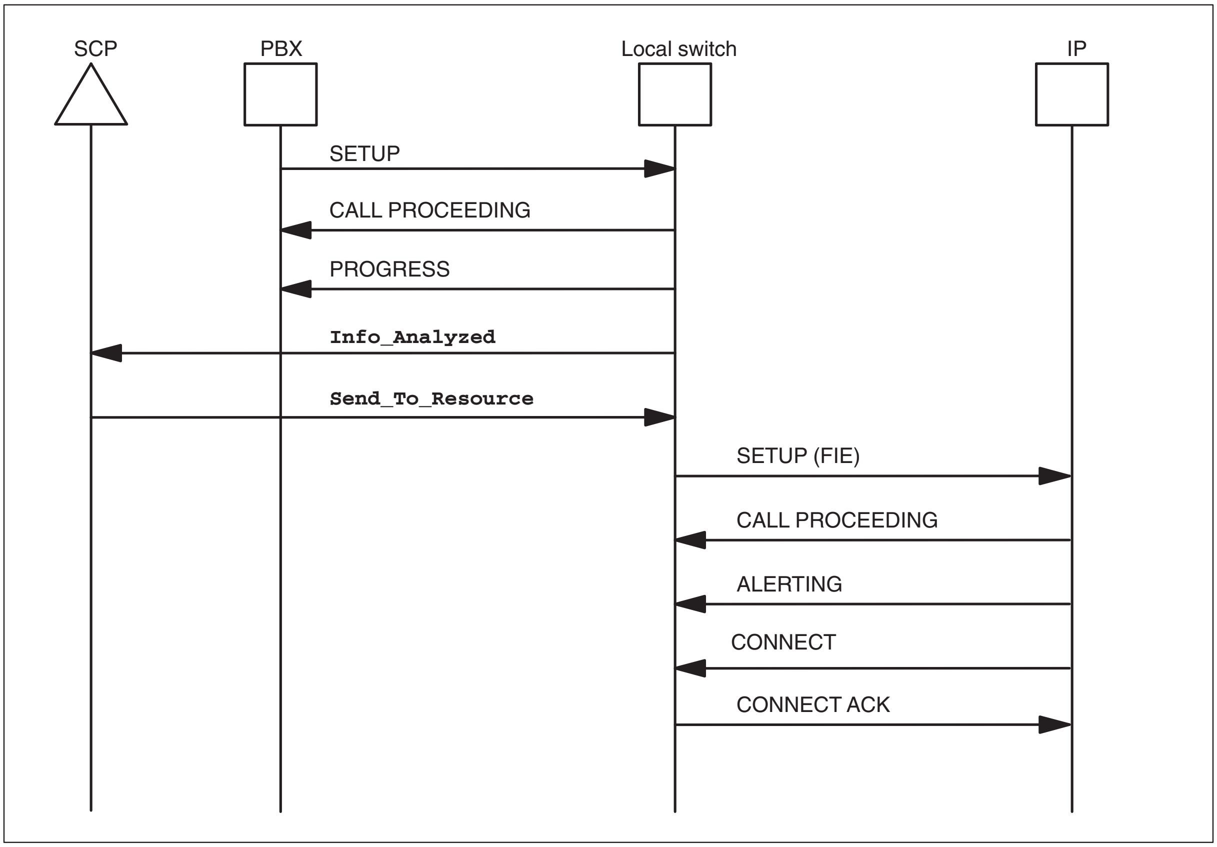


Table 2-2 describes the messages received from the local IP.

Table 2-2
Local switch processing of ISDN messages received from the local IP

Message	Definition
CALL PROCEEDING	This message indicates the local IP has initiated the call establishment requested by the local switch. ***Note:*** This message is not passed to the originating switch.
ALERTING	This message indicates that the local IP has been alerted to the call. This message is not passed to the originating switch.
CONNECT	This message indicates that the local IP has answered the call. For more information on what messages are passed to the originating switch, refer to the following four sections: —Send_To_Resource without AnswerIndicator parameter for SS7 originations —Send_To_Resource with AnswerIndicator parameter for SS7 originations —Send_To_Resource without AnswerIndicator parameter for PRI originations —Send_To_Resource with AnswerIndicator parameter for PRI originations

Figure 2-6 provides an example of call establishment to a local IP using SS7 signaling.

Figure 2-6
SS7 call establishment to a local IP using SS7 signaling

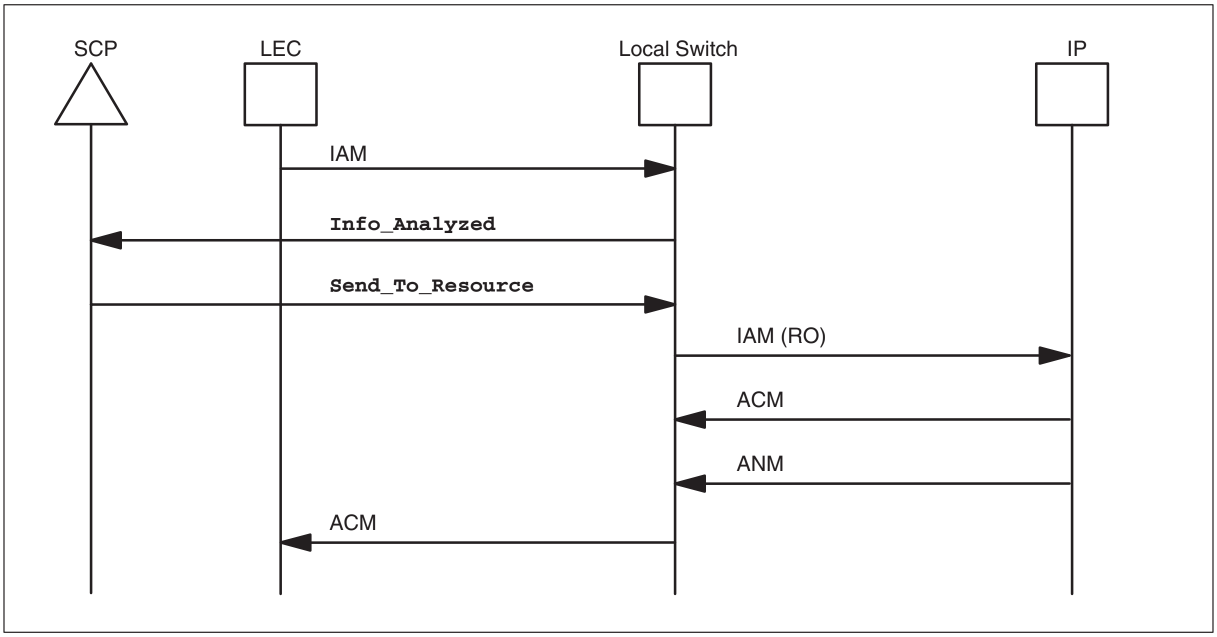


Table 2-3 describes the messages received from the local IP.

Table 2-3
Local switch processing of SS7 messages received from the local IP

Message	Definition
ACM	This message indicates the local IP has been alerted to the call.
ANM	This message indicates the local IP has answered the call. ***Note:*** This message is normally received following the ACM message. For more information on what messages are passed to the originating switch, refer to the following four sections: —Send_To_Resource without AnswerIndicator parameter for SS7 originations —Send_To_Resource with AnswerIndicator parameter for SS7 originations —Send_To_Resource without AnswerIndicator parameter for PRI originations —Send_To_Resource with AnswerIndicator parameter for PRI originations

Upon receipt of the ISDN CONNECT or SS7 ANM message the local switch starts the TSTRC (STR-Connection Timer). This timer provides a maximum time limit for a STR-Connection to an IP.

If the TSTRC timer expires before the IP has completed its service, the switch clears the connection and notifies the SCP. The timer is provisioned by the TSTRC_TIMER parameter in table CAINPARM and can be disabled if necessary.

The action taken upon receipt of an ISDN CONNECT or SS7 ANM message depends upon both the originating agent and the presence/absence of the ***AnswerIndicator*** parameter in the **Send_To_Resource** message.

The following examples describe actions performed based on originating agents and the presence/absence of the ***AnswerIndicator*** parameter.

Send_To_Resource without AnswerIndicator parameter for SS7 originations

- If an ACM has not already been sent to the originating switch, an ACM is sent with an optional backward call indicator parameter indicating `user-network interaction`.
- If an ACM has already been sent but did not indicate `user-network interaction`, a Call Progress (CPG) message is sent with an optional backward call indicator parameter indicating `user-network interaction`.
- No action is taken if an ACM or CPG has already been sent with an optional backward call indicator parameter indicating `user-network interaction`.

Figures 2-7 and 2-8 provide examples of call establishment to a local IP without answer indication.

Figure 2-7
SS7 call establishment to a local IP without Answer Indication, using SS7 signaling

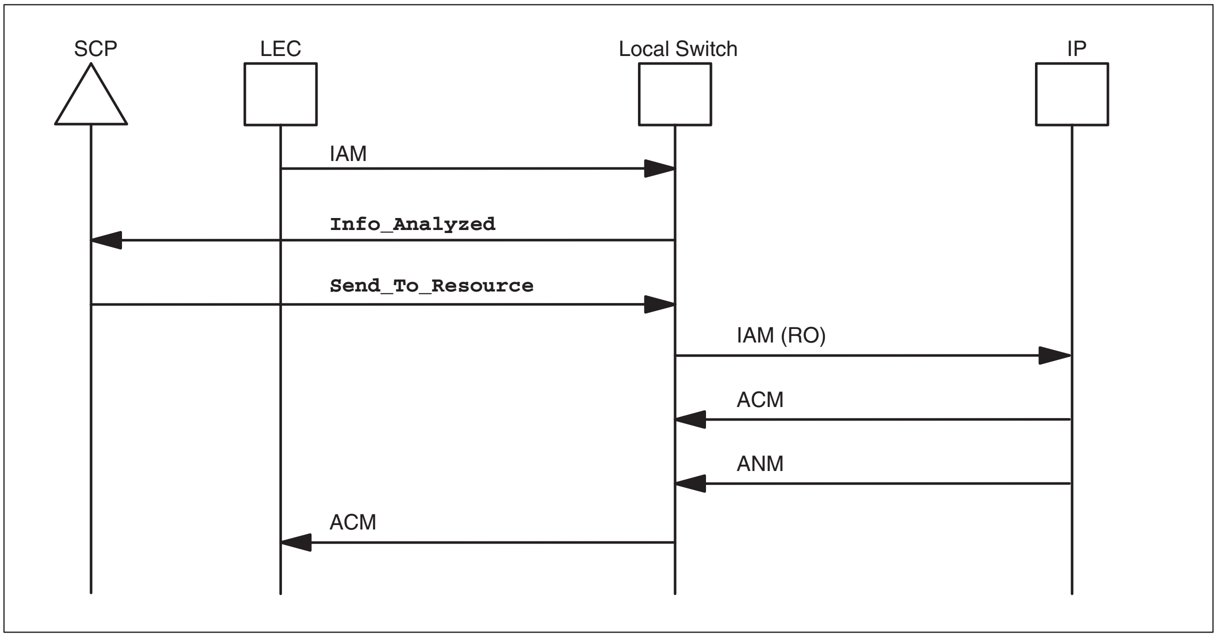
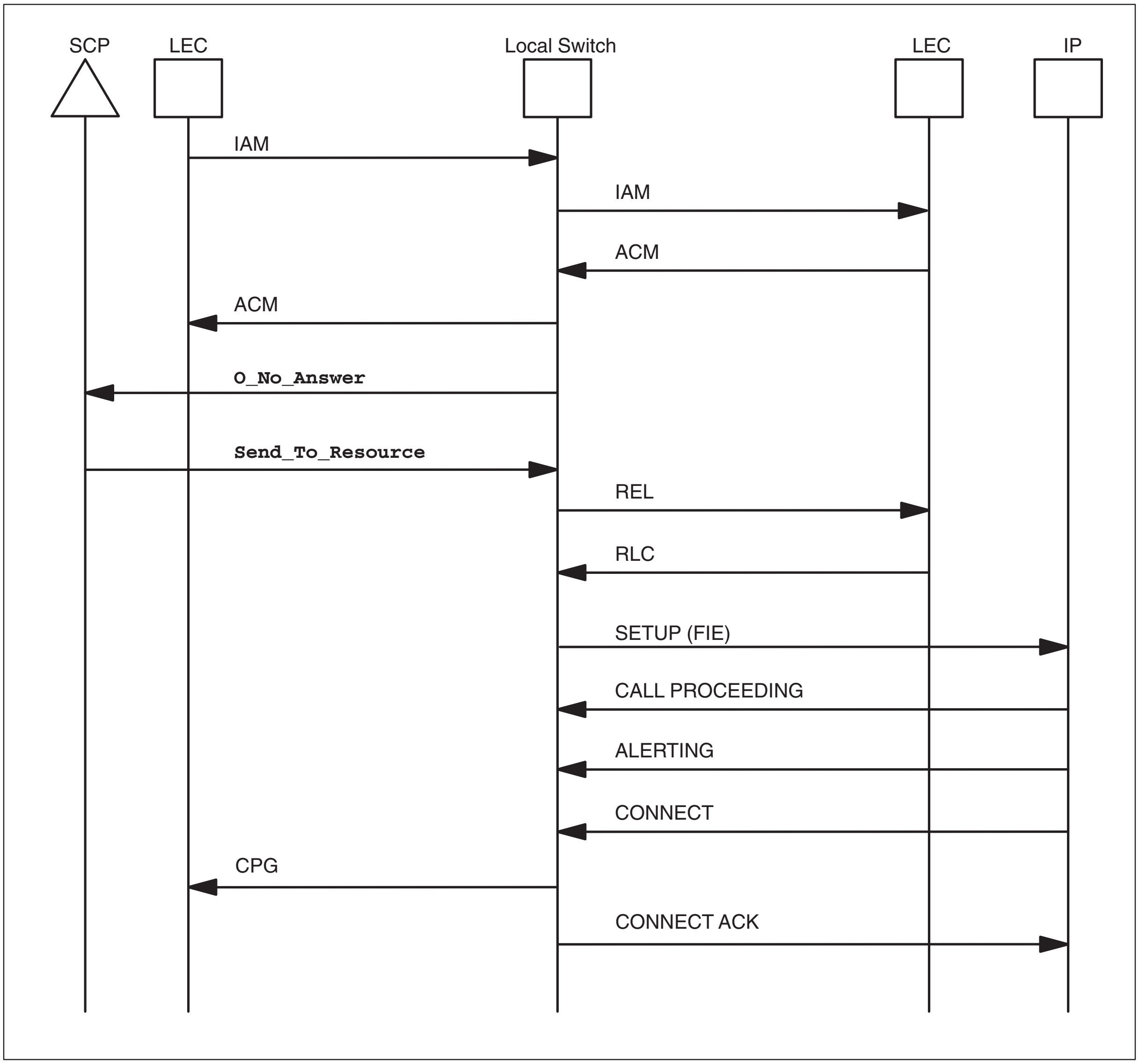


Figure 2-8
SS7 call establishment to a local IP without Answer Indication, using ISDN signaling



Send_To_Resource message with AnswerIndicator parameter for SS7 originations

- If an ACM has not already been sent to the originating switch, an ACM is sent with an optional backward call indicator parameter indicating `user-network interaction`.
- An ANM message is sent to the originating switch, if one has not previously been sent earlier in the call.

Figures 2-9 and 2-10 provide examples of call establishment to a local IP with answer indication.

Figure 2-9
SS7 call establishment to a local IP with Answer Indication, using ISDN signaling

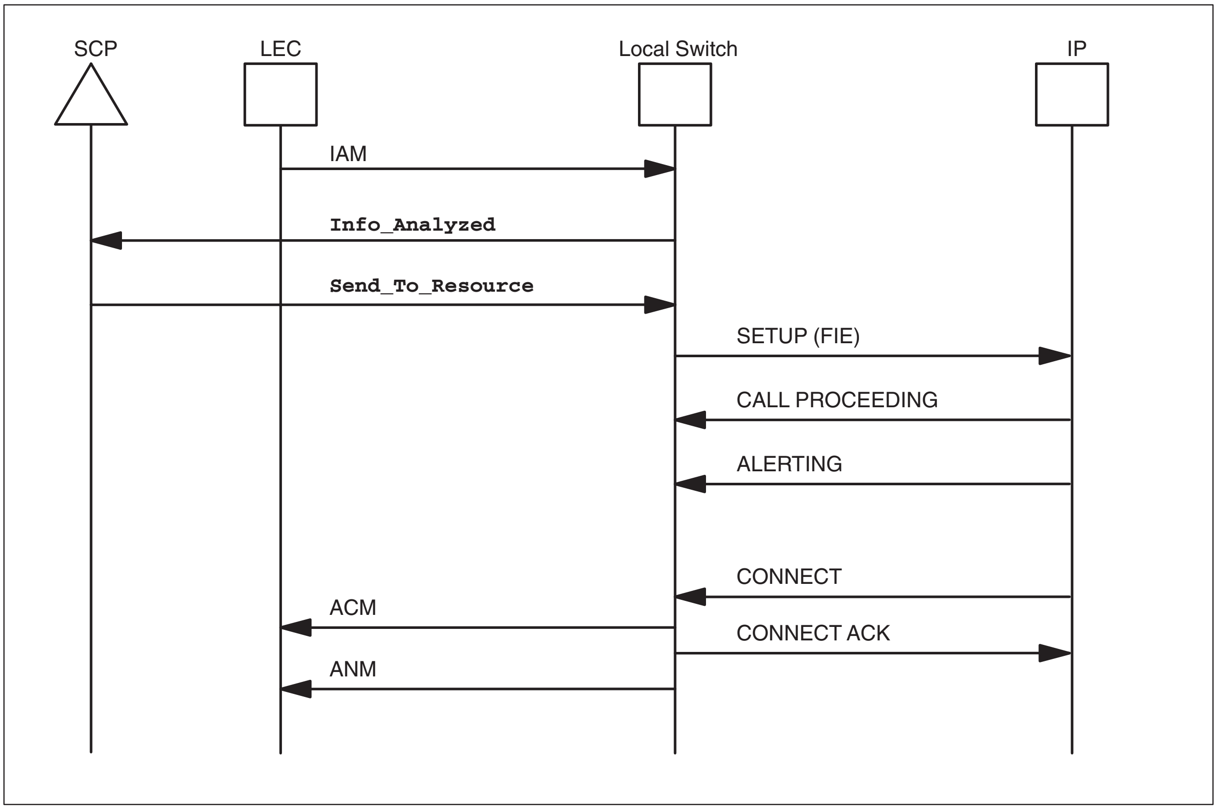
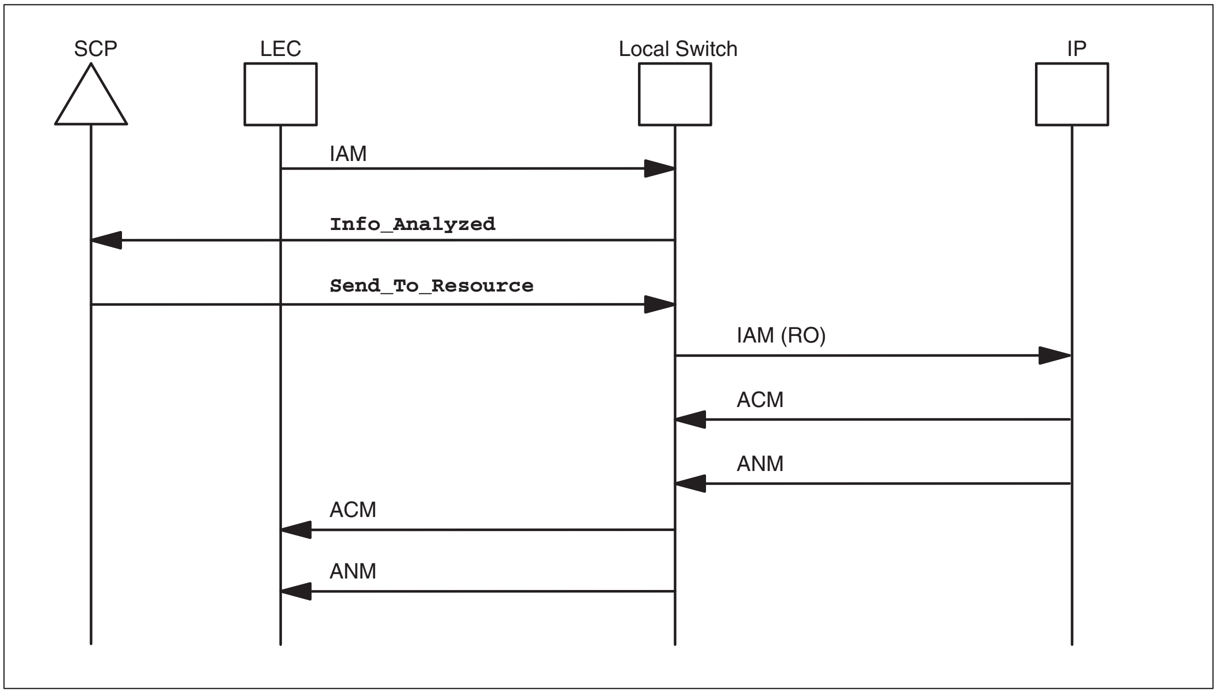


Figure 2-10
SS7 call establishment to a local IP with Answer Indication, using SS7 signaling



Send_To_Resource without AnswerIndicator parameter for PRI originations

- If no message beyond CALL PROCEEDING has been sent to the originating switch, a PROGRESS message with the progress indicator set to "inband information or appropriate pattern now available" is sent to the originating switch.
- If a CONNECT message has already been sent to the originating switch, no message is sent.

Figures 2-11 and 2-12 provide examples of call establishment to a local IP without answer indication.

Figure 2-11
PRI call establishment to a local IP without Answer Indication, using ISDN signaling

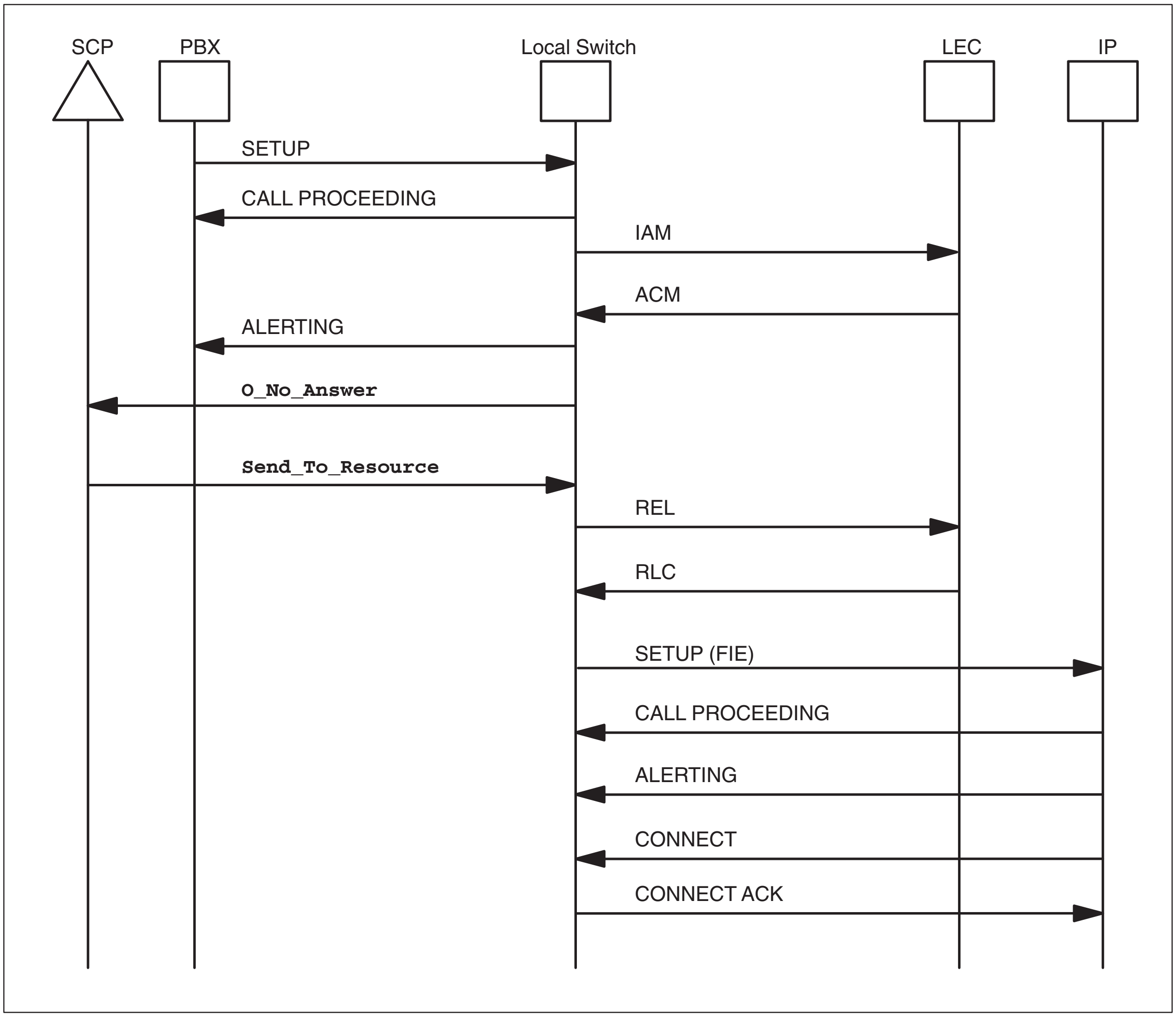
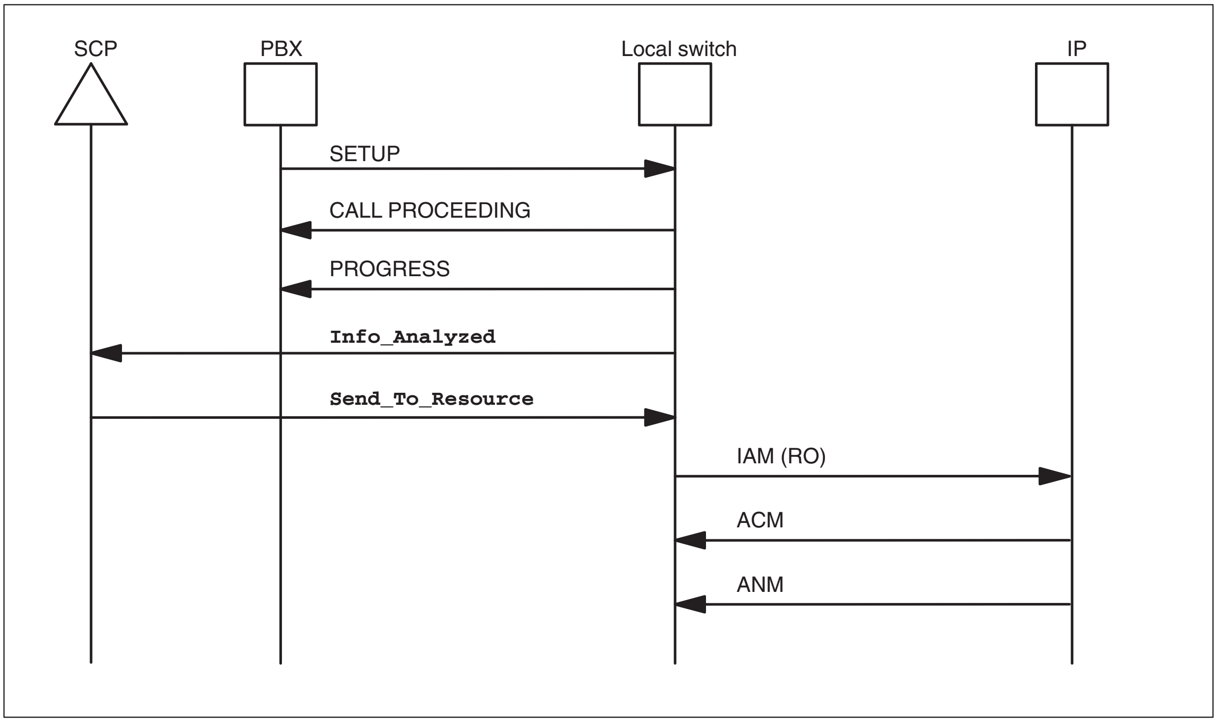


Figure 2-12
PRI call establishment to a local IP without Answer Indication, using ISDN signaling



Send_To_Resource with AnswerIndicator parameter for PRI originations

- A CONNECT message is sent to the originating switch, unless one has previously been sent to the originating switch.

Figures 2-13 and 2-14 provide examples of call establishment to a local IP with answer indication.

Figure 2-13
PRI call establishment to a local IP with Answer Indication, using ISDN signaling

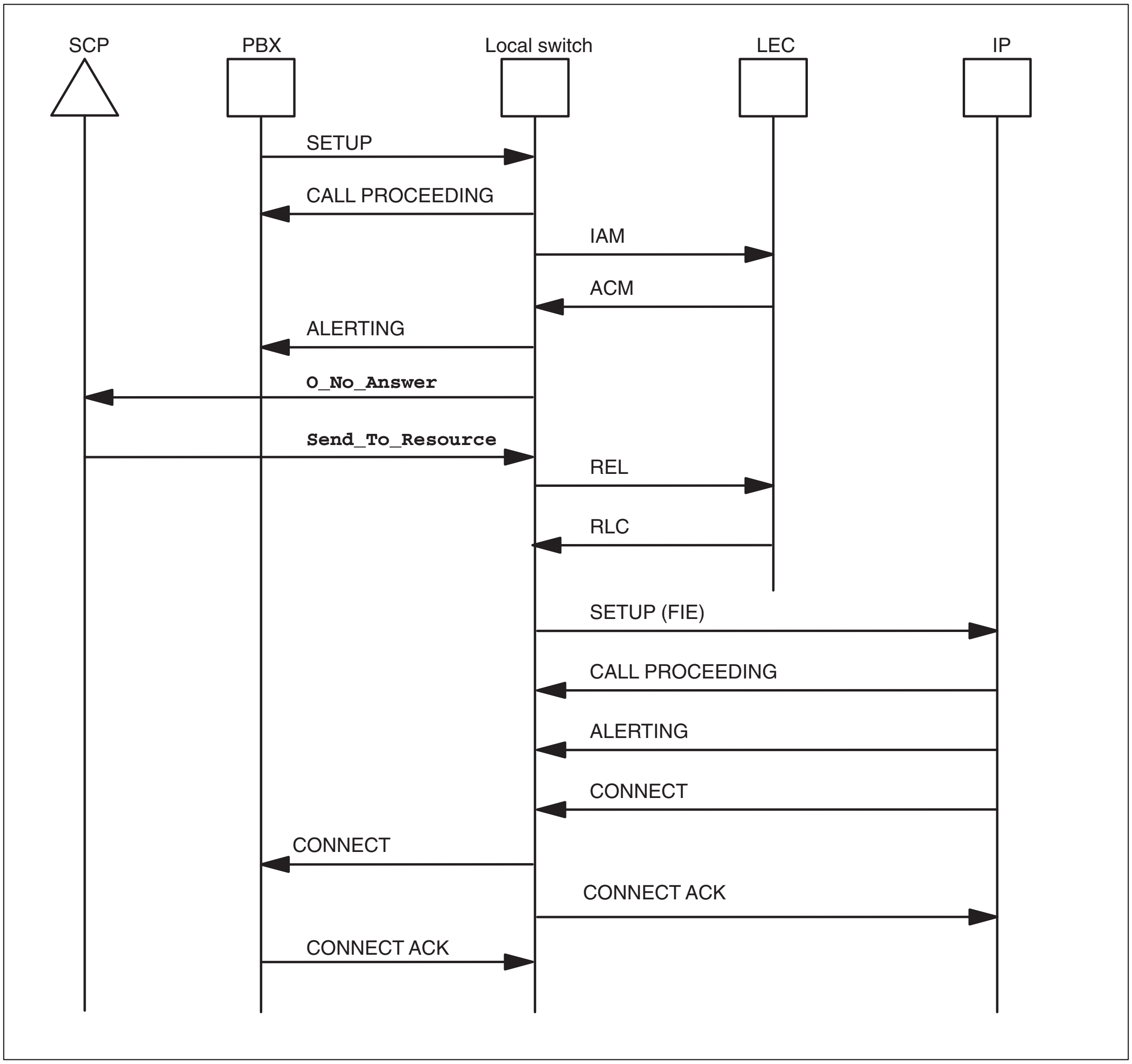
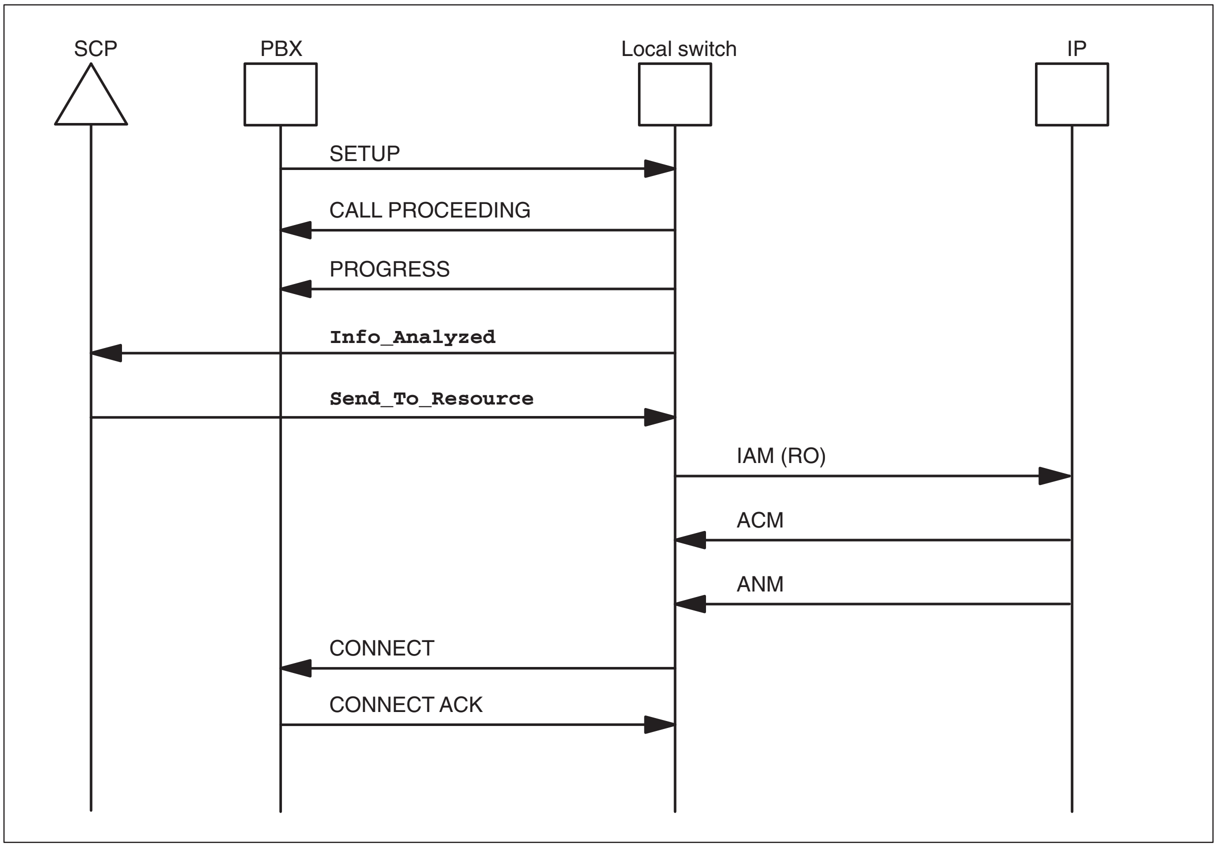


Figure 2-14
PRI call establishment to a local IP with Answer Indication, using SS7 signaling



Signaling for CONNECT_ONLY during an active connection to a local IP

A CONNECT_ONLY STR-Connection is considered active when an answer indication is receive from the IP. While the STR-Connection is active to a local IP, the local switch may receive an ISDN FACILITY or SS7 FAR message from the local IP requesting a supplemental service. As stated earlier in this document, the CONNECT_ONLY IPI does not support an exchange of data between the SCP and IP during an active connection. However, release link trunk (RLT) and billing information supplemental services are currently supported on the switch.

If the local switch receives a FACILITY or SS7 FAR message from the local IP during an active CONNECT_ONLY STR-Connection, a **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `suppServiceInvoked`.

Signaling for CONNECT_1129_STYLE during an active connection to a local IP

A CONNECT_1129_STYLE STR-Connection is considered active when an answer indication is received from the IP. When a STR-Connection with ISDN signaling is established to a local IP, the local switch may receive an ISDN FACILITY message containing an FIE with a Return Result component. When a STR-Connection with SS7 signaling is established to a local IP, the local switch may receive an SS7 FAR message containing an RO parameter with a Return Result component. These messages allow the SCP and IP to exchange intermediate information during a CONNECT_1129_STYLE STR-Connection.

If there is already an outstanding **`Call_Info_To_Resource`** message for the last **`Call_Info_From_Resource`** message, the switch initiates call clearing toward the IP by sending an ISDN DISCONNECT or SS7 REL message. A **`Resource_Clear`** message in a conversational package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.

Upon receipt of a valid ISDN FACILITY or SS7 FAR message, the switch builds a **`Call_Info_From_Resource`** message containing the ***`IPReturnBlock`*** parameter when it is present in the Return Result component. The message is sent to the SCP and the T1 timer is started.

The switch initiates call clearing toward the IP if the switch detects any of the following conditions:

- the SCP responds with an **`Application_Error`** or **`Failure_Report`** message.
- the T1 timer expires.
- the switch detects a fatal application error.

The switch performs the following actions when the switch detects one of the conditions in the preceding paragraph:

- sends an ISDN DISCONNECT message or SS7 REL message to the IP.
- routes call to AIN Final (AINF) treatment.
- cancels the TSTRC timer.

The expected response to the **`Call_Info_From_Resource`** message is the **`Call_Info_To_Resource`** message. The message is processed differently depending upon the presence or absence of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters.

Call_Info_To_Resource without ResourceType and StrParameterBlock parameters

- In this scenario the SCP has determined that the IP is no longer needed for the call. The switch initiates call clearing toward the IP by sending and ISDN DISCONNECT or SS7 REL message.
- A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`.

Figures 2-15 and 2-16 provide examples of **`Call_Info_To_Resource`** messages received without ***`ResourceType`*** and ***`StrParameterBlock`*** parameters.

Figure 2-15
Call_Info_To_Resource message without parameters using ISDN signaling

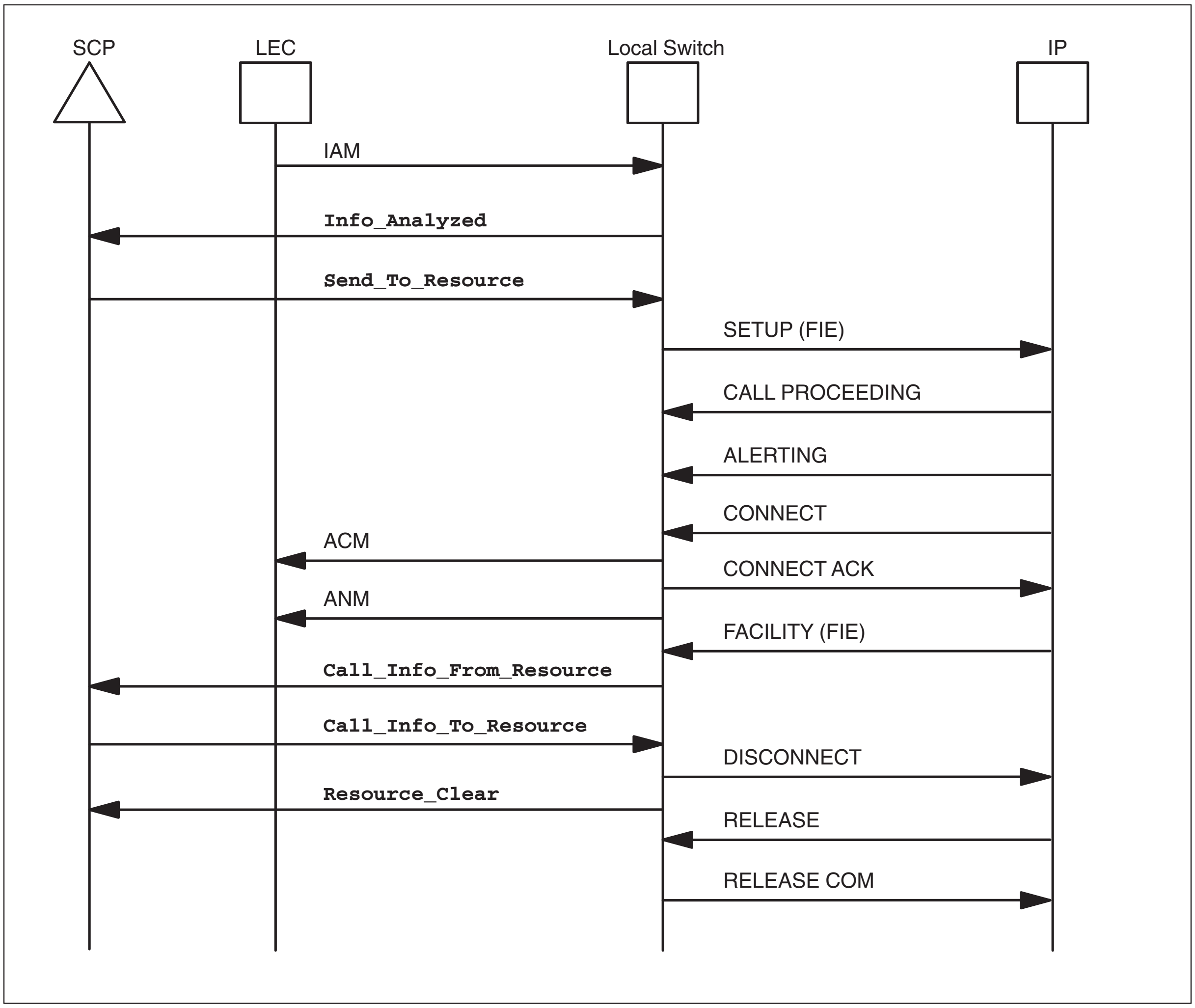
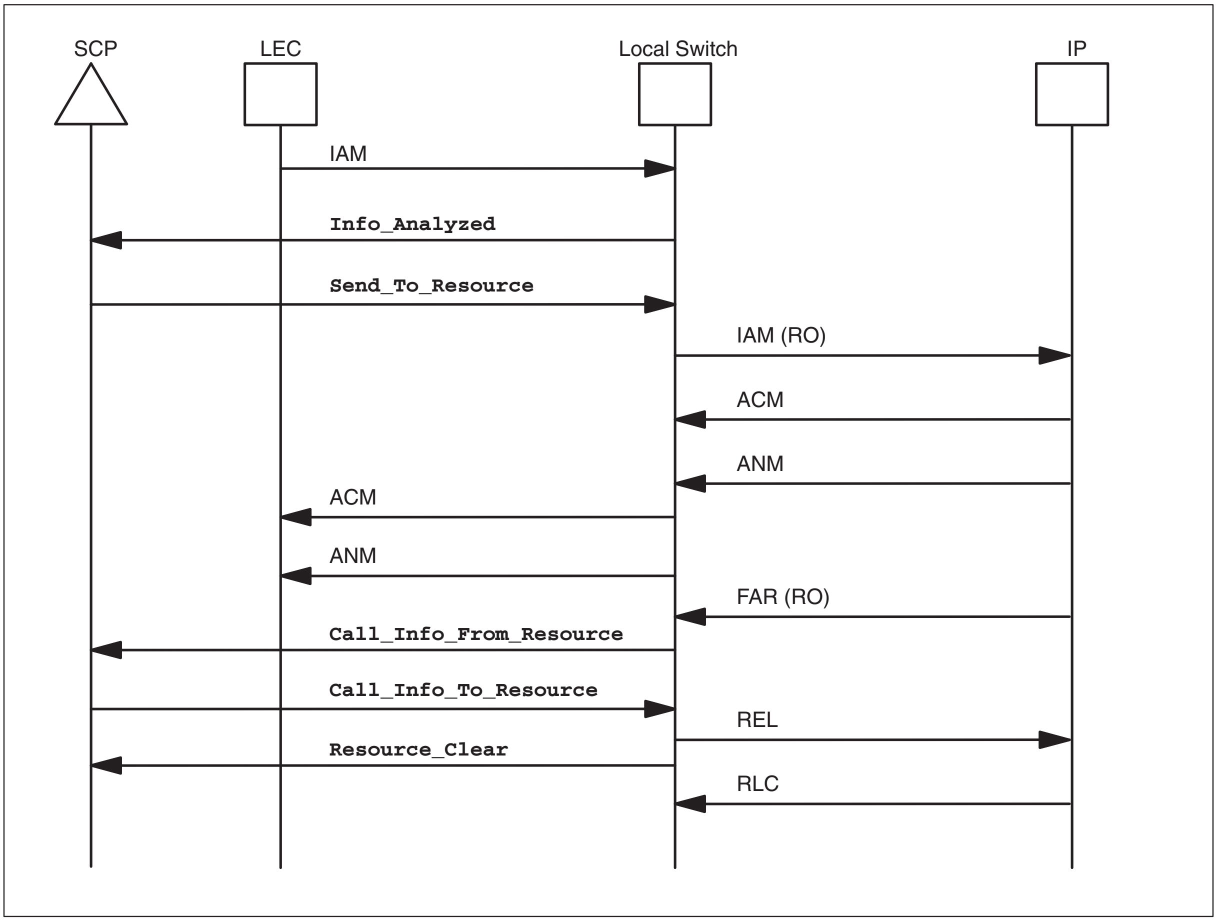


Figure 2-16
Call_Info_To_Resource message without parameters using SS7 signaling



Call_Info_To_Resource with ResourceType and StrParameterBlock parameters

- In this scenario the SCP has determined that additional information needs to be passed to the IP. An ISDN FACILITY message with an FIE or an SS7 FAR message with an RO parameter is sent to the IP. The FIE or RO contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the switch.

Figures 2-17 and 2-18 provide examples of **Call_Info_To_Resource** messages received with ***ResourceType*** and ***StrParameterBlock*** parameters.

Figure 2-17
Call_Info_To_Resource message with parameters using ISDN signaling

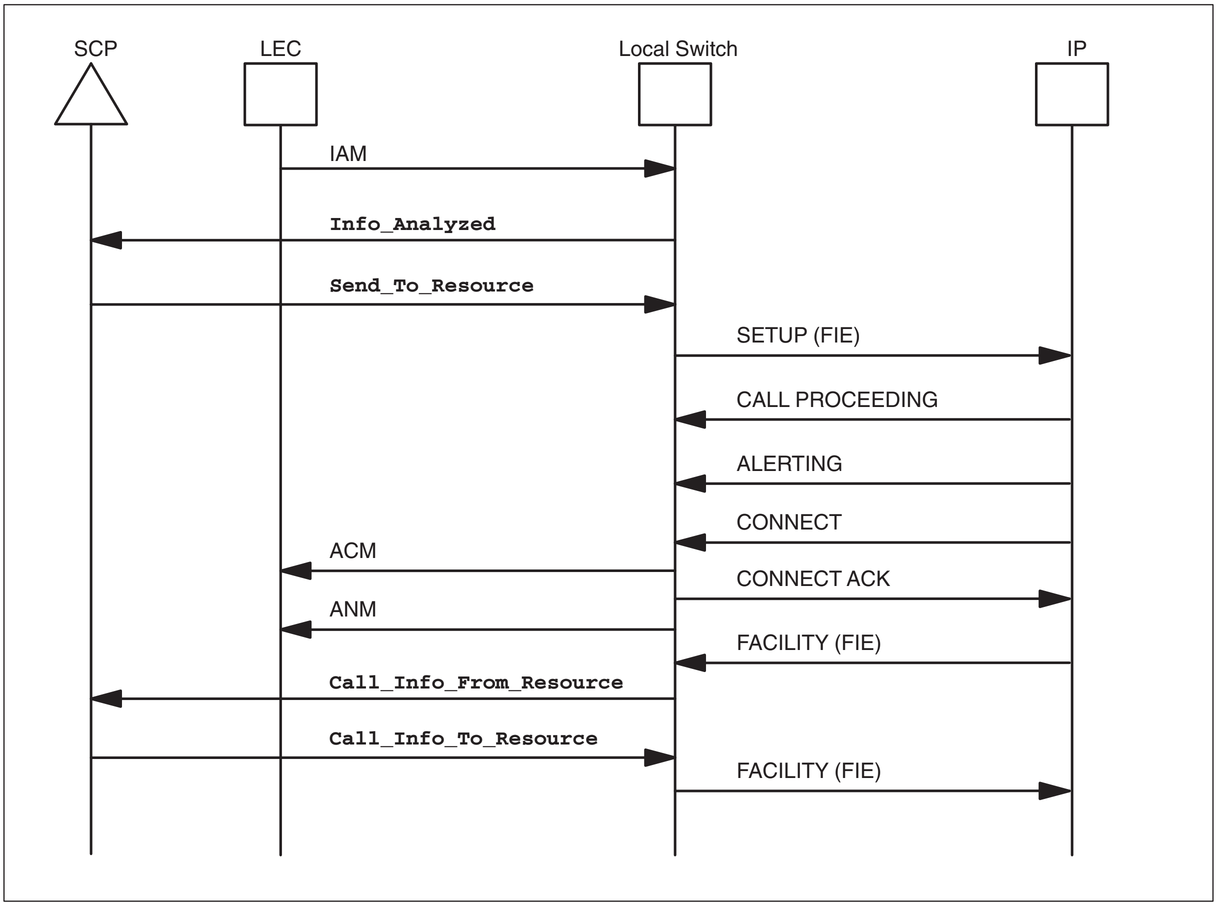
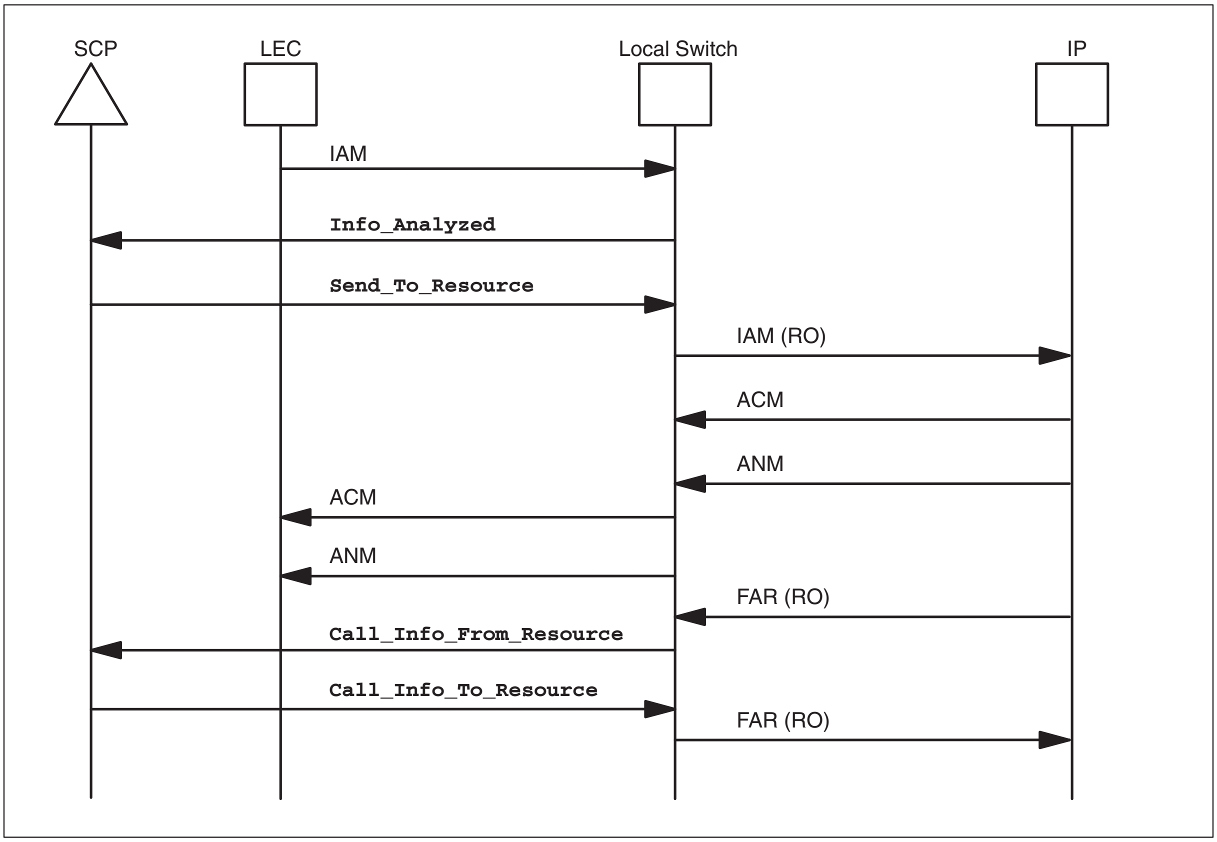


Figure 2-18
Call_Info_To_Resource message with parameters using SS7 signaling



During an active CONNECT_1129_STYLE STR-Connection the switch may receive an ISDN FACILITY message with an FIE or an SS7 FAR message with a facility indicator indicating a service. This occurs when the IP is requesting a supplemental service on the switch. Release link trunk (RLT) and billing information supplemental services are currently supported on the switch.

Release link trunk (RLT)

The RLT supplemental service allows an IP to connect to a second subscriber and then bridge the second subscriber to the calling subscriber. Once the call is bridged, the two subscribers are connected and the trunks between the IP and switch are released. The RLT supplemental service also allows an IP to redirect (for SS7 RLT calls only) or transfer the call. In the case of SS7 RLT, reorigination information can also be updated through supplemental services.

The STR-Connection is terminated when the RLT service is invoked by the IP. This is necessary since the RLT service allows the IP to control the routing of the call instead of the SCP. The switch sends a `Resource_Clear` message to the SCP in a response package with a ***ClearCause*** value of `suppServiceInvoked`. The RLT service processes the incoming FACILITY or FAR message.

Billing Information

The Billing Information supplemental service allows an IP, using ISDN signaling, to provide billing information to update the CDR fields BILLNUM, ACCTCD, and PINDIGS. This supplemental service does not affect the STR-Connection to the IP. However, it is important to note that the IP-provided billing information may be overwritten later in the call by SCP-provided billing information.

IP-initiated clearing of a CONNECT_ONLY STR-Connection to a local IP

Once the local switch sends the ISDN SETUP message or SS7 IAM message to the local IP, the local IP may respond with a message that clears the STR-Connection. The following information describes the ISDN RELEASE, RELEASE COMPLETE and DISCONNECT messages and SS7 REL and RLC messages which clear the STR-Connection.

The exchange of data using the DISCONNECT or REL message is not supported for the CONNECT_ONLY IPI. When either message is received a **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `normal`. The Return Result component is ignored if it is present in the incoming DISCONNECT or REL message. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

When either an ISDN RELEASE COMPLETE or SS7 REL message is received, a **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `abort`. The Reject component is ignored if it is present in the incoming RELEASE COMPLETE or REL message. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

The exchange of data using the ISDN FACILITY or SS7 FAR message is not supported for the CONNECT_ONLY IPI.

Figures 2-19 and 2-20 provide examples of normal call clearing for the CONNECT_ONLY IPI.

Figure 2-19
Normal IP-initiated call clearing with the DISCONNECT message using ISDN signaling

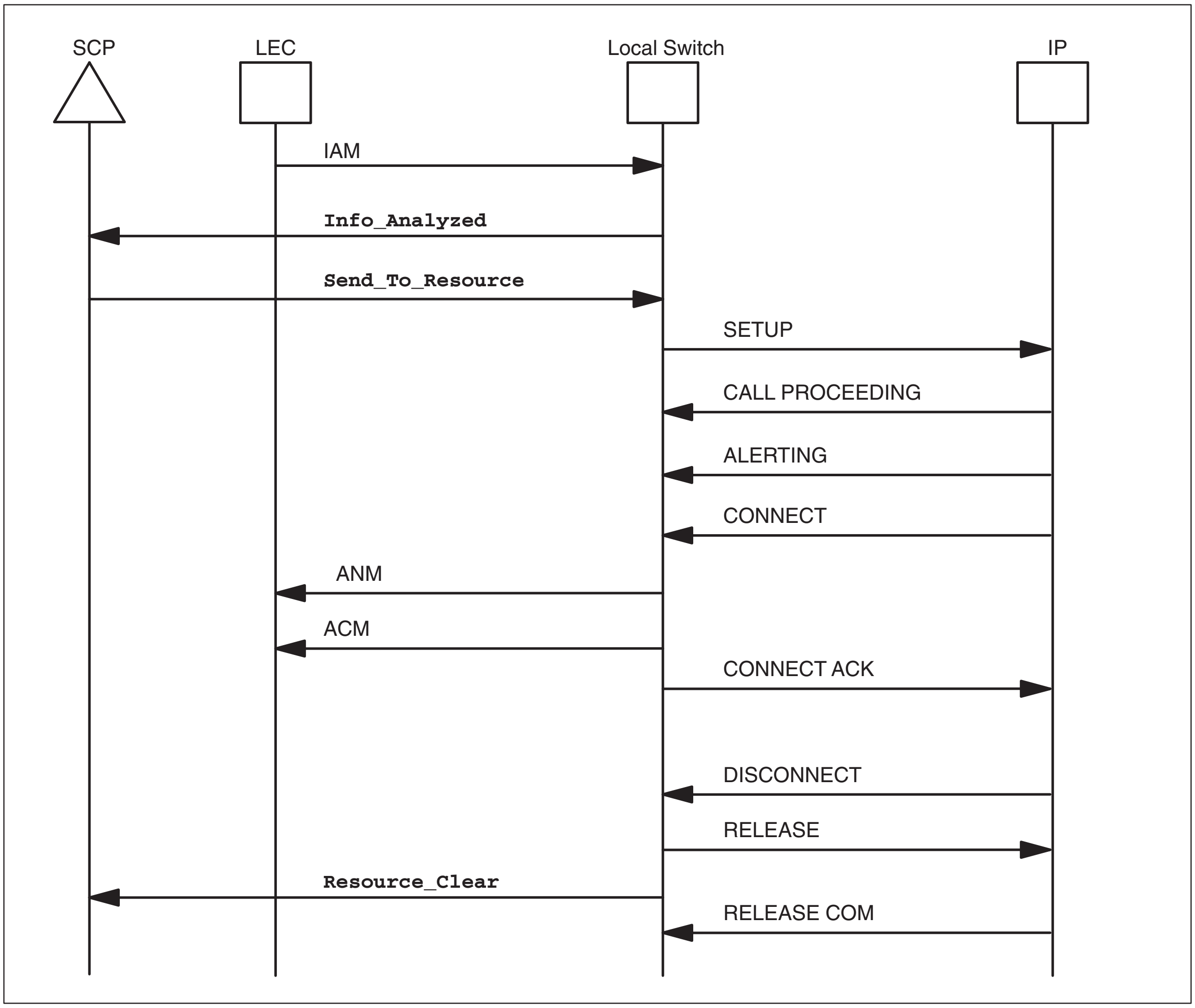
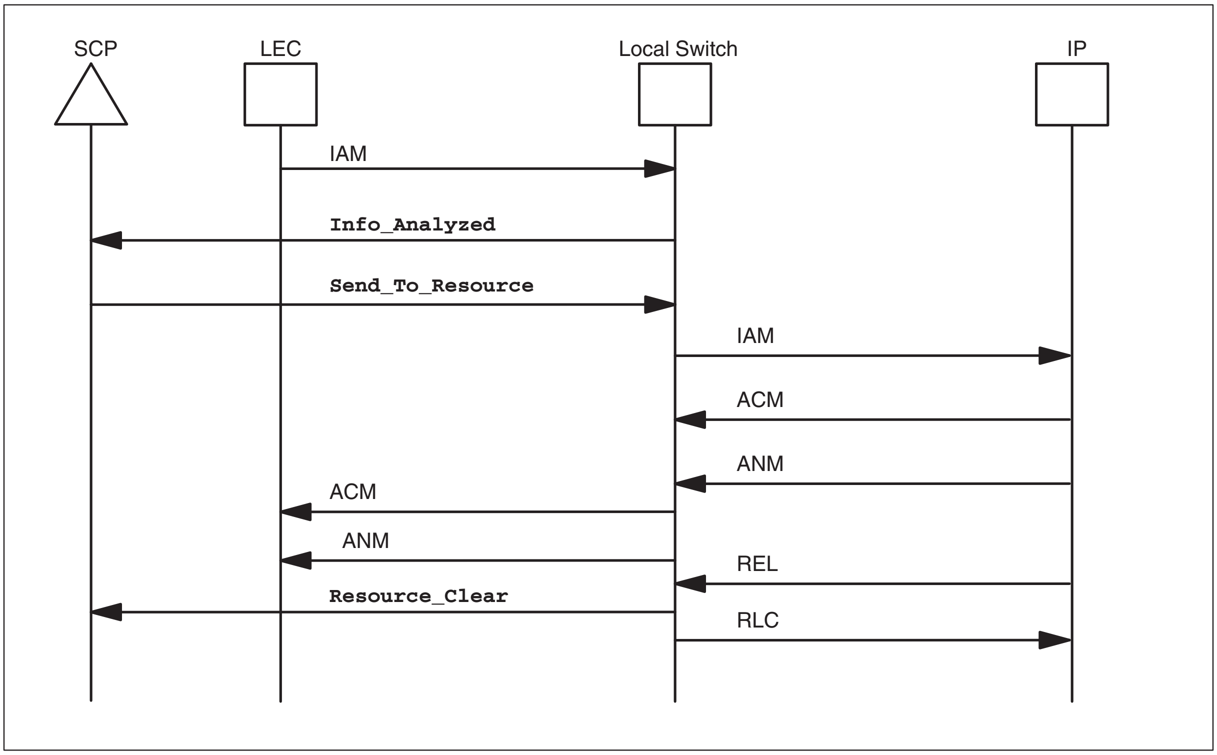


Figure 2-20
Normal IP-initiated call clearing with the REL message using SS7 signaling



The caller may abandon the call during a STR-Connection to the local IP. When this occurs, the switch sends a **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.

Note: When the caller abandon occurs during an active STR-Connection (as in Figure 2-21), Bellcore's *GR-1129-CORE* states that a FACILITY message should be sent to the local IP indicating that the connection needs to be cancelled. As stated earlier in this document, the CONNECT_ONLY IPI does not support the exchange of data during an active STR-Connection. Therefore, a DISCONNECT message is sent to the local IP indicating that call clearing is required.

Figure 2-21 provides an example of CONNECT_ONLY call clearing due to a caller abandon. Tables 2-4 and 2-5 provide a list of ISDN and SS7 clearing messages that local switch receives the the local IP.

Figure 2-21
Switch-initiated call clearing for a caller abandon

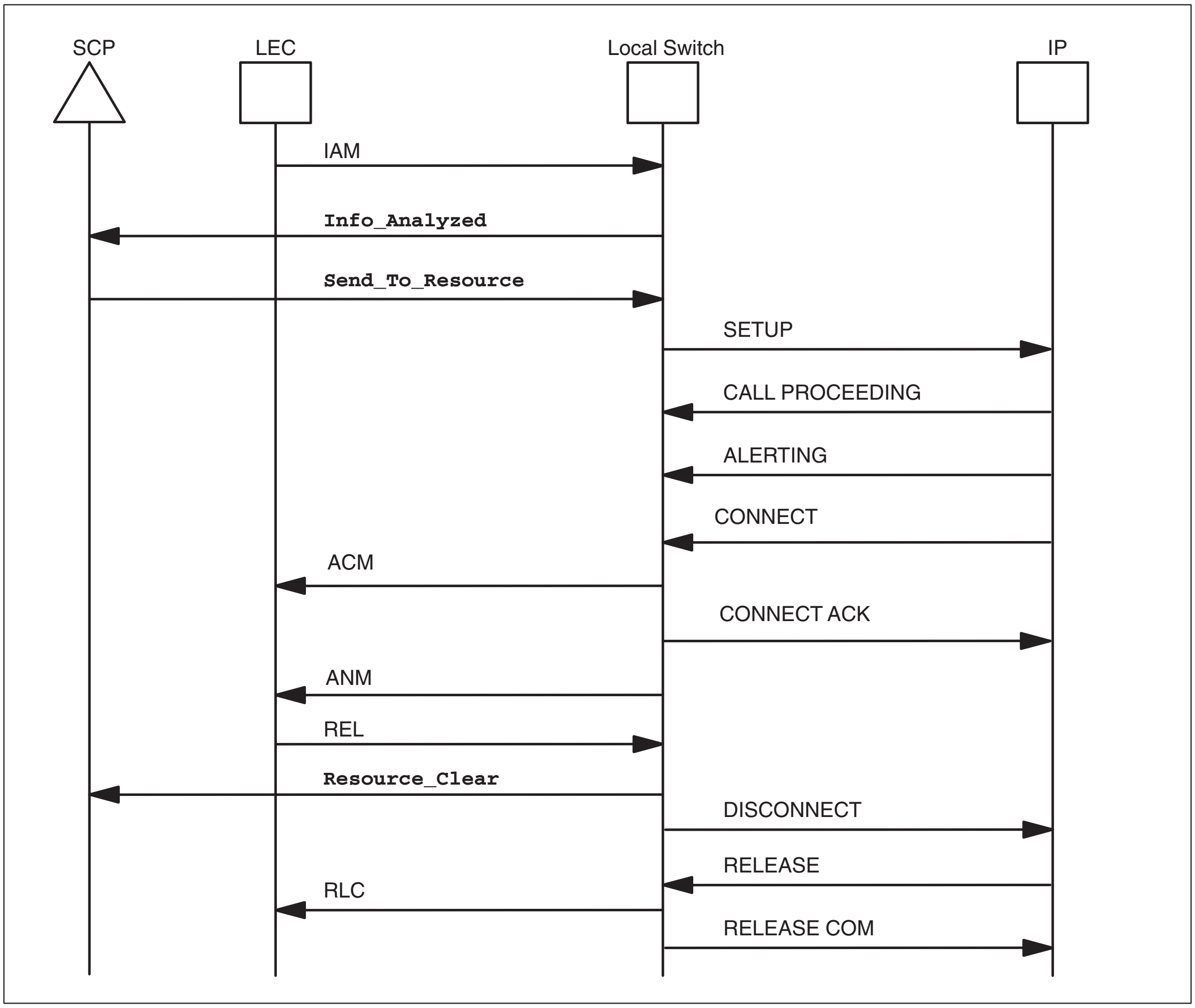


Table 2-4
Local switch processing of ISDN clearing messages received from the local IP

Message	Description
RELEASE or RELEASE COMPLETE	During abnormal conditions, the local IP sends a RELEASE message to initiate call clearing. Examples of abnormal conditions include: • protocol violations • protocol time-outs • B-channel not available • B-channel glare • rejection of the call by the local IP When the local switch receives a RELEASE or RELEASE COMPLETE message, a **`Resource_Clear`** message is sent in a conversation package with a ***`ClearCause`*** of `abort`. The RELEASE or RELEASE COMPLETE message is not passed to the originating switch.
DISCONNECT	This message initiates normal call clearing. When the local switch receives a DISCONNECT, a **`Resource_Clear`** is sent in a conversation package with a ***`ClearCause`*** of `normal`. The DISCONNECT message is not passed to the originating switch.

Table 2-5
Local switch processing of SS7 clearing messages received from the local IP

Message	Description
REL	This message initiates normal call clearing. When the local switch receives an REL, a **`Resource_Clear`** is sent in a conversation package with a ***`ClearCause`*** of `normal`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component. The REL message is not passed to the originating switch.

IP-initiated clearing of a CONNECT_1129_STYLE STR-Connection to a local IP

Once the local switch sends the ISDN SETUP message or SS7 IAM message to the local IP, the local IP may respond with a message that clears the STR-Connection. The following information describes the ISDN RELEASE, RELEASE COMPLETE and DISCONNECT messages and SS7 REL and RLC messages which clear the STR-Connection.

When initiating normal call clearing the local IP sends a DISCONNECT or REL message with a Cause Indicator indicating normal clearing. When the

DISCONNECT or REL message contains a Return Result component, a **Resource_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of normal. The ***IPReturnBlock*** parameter is included in the message when it is present in the Return Result component. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

Figures 2-22 and 2-23 provide examples of normal call clearing for the CONNECT_1129_STYLE IPI.

Figure 2-22
Normal IP-initiated call clearing with a Return Result component using ISDN signaling

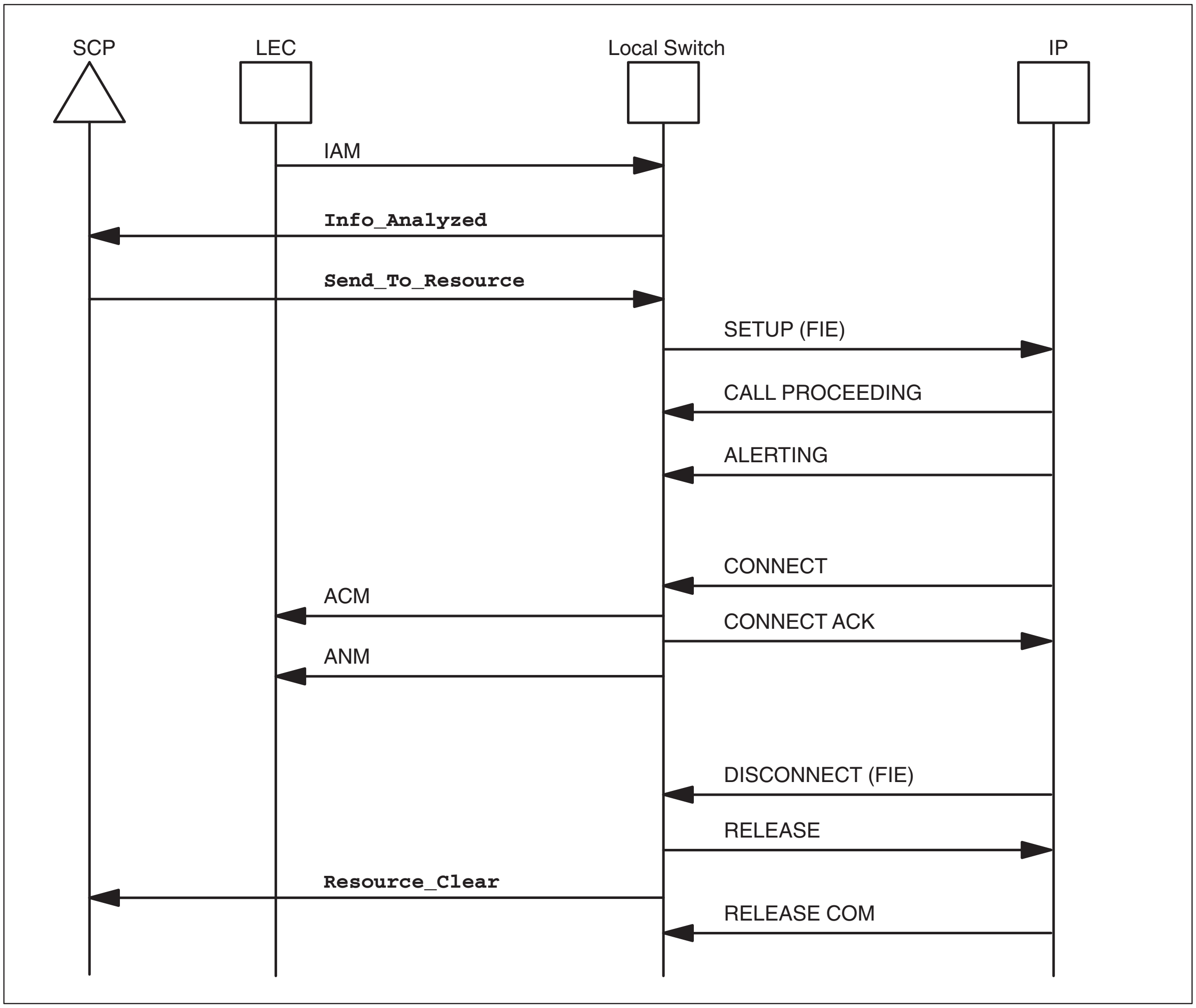
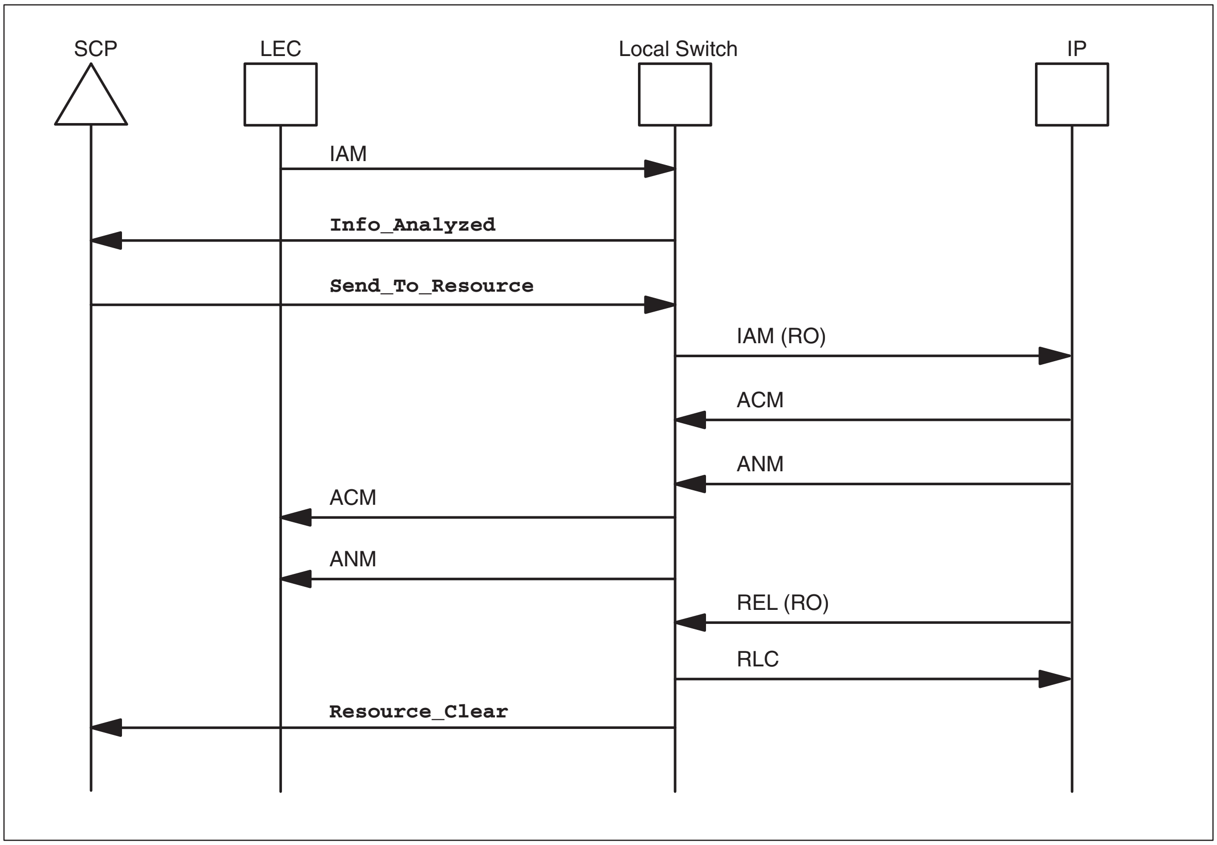


Figure 2-23
Normal IP-initiated call clearing with a Return Result component using SS7 signaling



When the ISDN DISCONNECT or SS7 REL message contains a Return Error component, a **Resource_Clear** message in a conversation package is sent to the SCP. The value of the ***ClearCause*** parameter is based upon the contents of the Error Value field of the Return Error component. The ***ClearCauseData*** parameter is included in the **Resource_Clear** message when the error parameter is present in the Return Error component. (The ***ClearCauseData*** parameter can only be present when the Error Value is `taskRefused`.) The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

When the DISCONNECT or REL message contains a Reject component, a **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `protocolError`. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

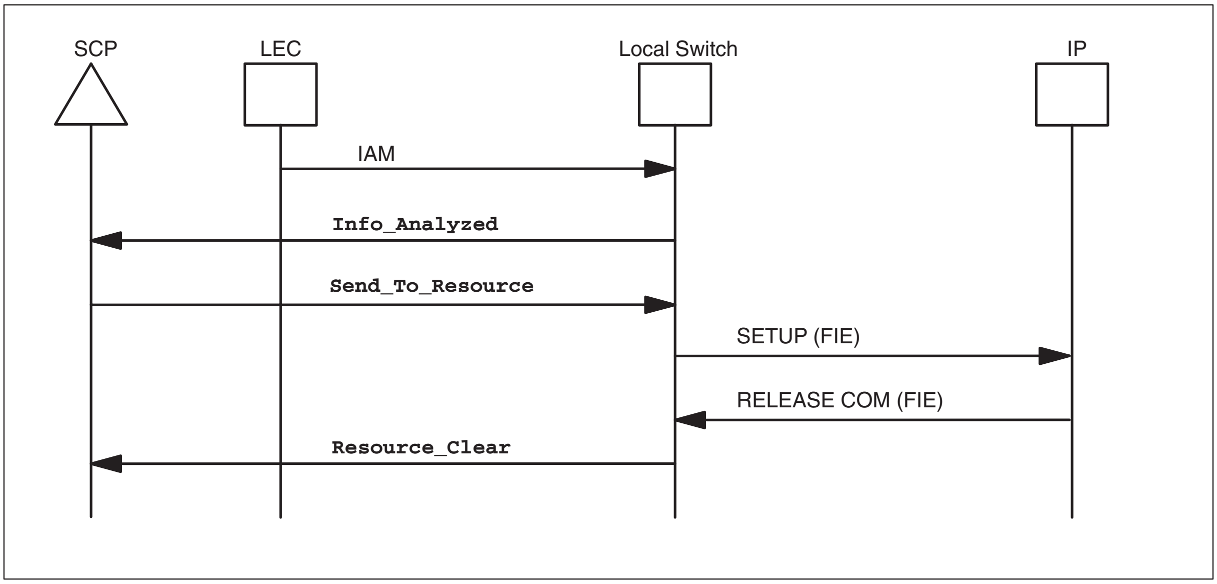
If the switch receives a DISCONNECT or REL message without a component, a **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `abort`. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

Note: As stated in the preceding text, for the CONNECT_ONLY IPI, a DISCONNECT or REL message without a component is expected for this IPI. A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`.

When initiating abnormal call clearing, the local IP may send an ISDN RELEASE COMPLETE or SS7 REL message with a Cause Indicator indicating abnormal clearing. When the message contains a Reject component, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

Figure 2-24 provides an example of abnormal call clearing for the CONNECT_1129_STYLE IPI.

Figure 2-24
Abnormal IP-initiated call clearing with a Reject component using ISDN signaling



If the ISDN RELEASE COMPLETE or SS7 REL message does not contain a component, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

When initiating abnormal call clearing during an active STR-Connection, the local IP may send an ISDN FACILITY or SS7 FAR message. When the FACILITY or FAR message contains a Return Error component, a **`Resource_Clear`** message in a conversation package is sent to the SCP. The value of the ***`ClearCause`*** parameter is based upon the contents of the Error Value field of the Return Error component. The ***`ClearCauseData`*** parameter is included in the **`Resource_Clear`** message when the error parameter is present in the Return Error component. (The ***`ClearCauseData`*** parameter can only be present when the Error Value is `taskRefused`.) The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

When the FACILITY or FAR message contains a Reject component a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`. The STR-Connection is cleared and the TSTRC Timer is canceled. The call is not cleared toward the calling user.

Switch-initiated clearing of a STR-Connection to a local IP

The local switch may end a CONNECT_ONLY STR-Connection when

- a caller abandon occurs.
- the SCP sends a **`Cancel_Resource_Event`** to the local switch.
- the TSTRC Timer expires.

Caller abandon

The caller may abandon the call during a STR-Connection to the local IP. When this occurs before the STR-Connection is active, a **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.

Figures 2-25 and 2-26 provide examples of caller abandon before an active STR-Connection is established.

Figure 2-25
Caller abandon before an active STR-Connecting using ISDN signaling

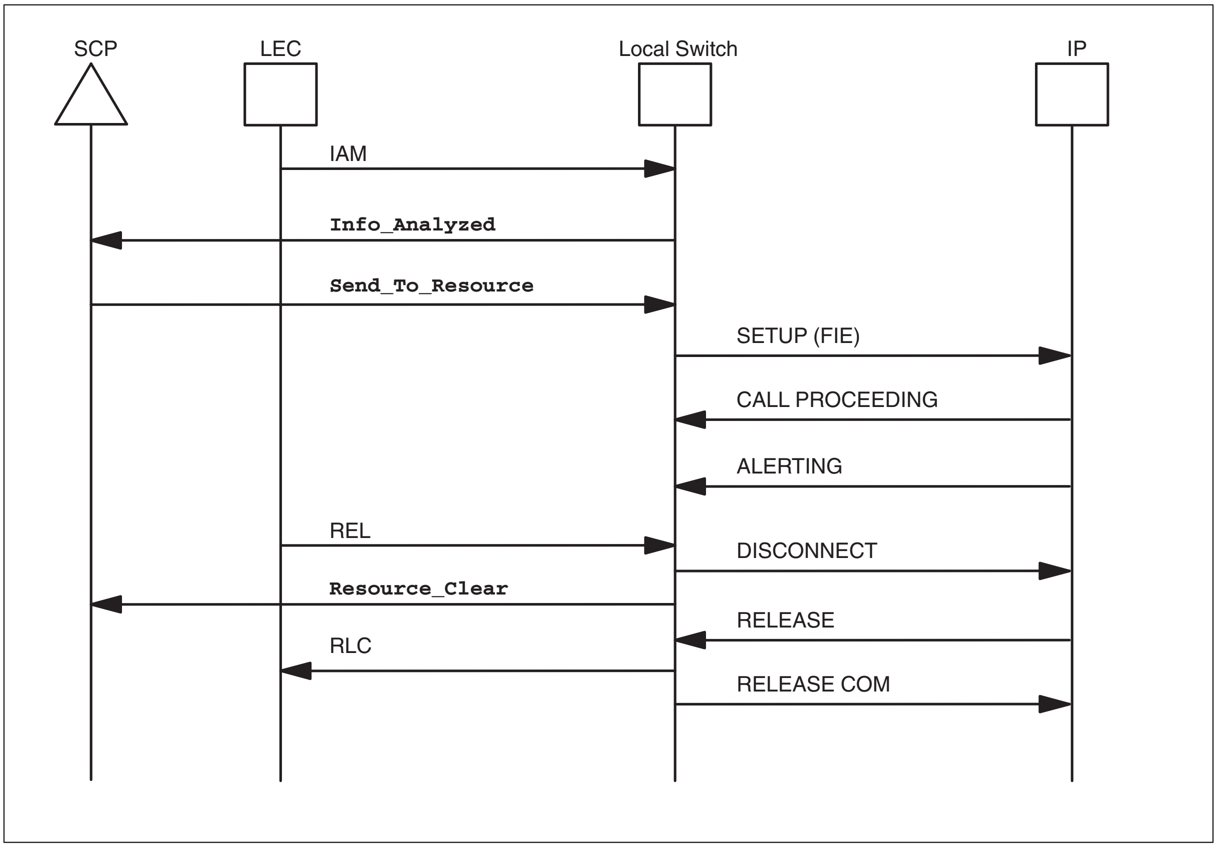
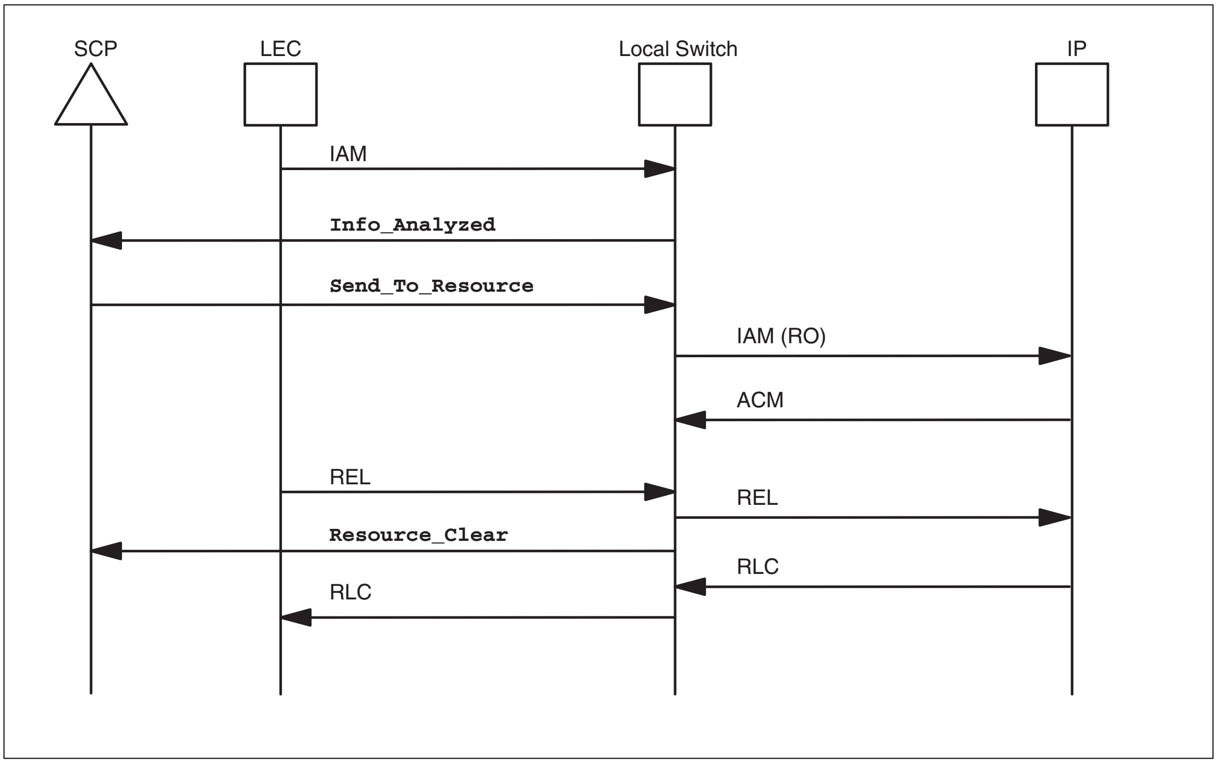


Figure 2-26
Caller abandon before an active STR-Connection using SS7 signaling



When the calling user abandons during an active STR-Connection, a FACILITY or FAR message containing an Invoke component with an operation of `cancelIPResource` is sent to the IP. The IP Disconnect Timer (TDISC) is started. This timer specifies the maximum time in seconds in which an IP must respond to a FACILITY or FAR message with the `cancelIPResource` operation. The TSTRC Timer is canceled.

It is possible that the user abandoned while the switch was waiting for a **`Call_Info_To_Resource`** message. If the message is received after the FACILITY or FAR message with the `cancelIPResource` operation was sent to the IP, the T1 timer is canceled and the **`Call_Info_To_Resource`** message is discarded.

The IP is expected to respond to the `cancelIPResource` operation with a DISCONNECT or REL message normally containing a Return Result component.

- If the T1 timer is not running, a **`Resource_Clear`** message is sent to the SCP in a response package with a ***`ClearCause`*** parameter value of `userAbandon`. The ***`IPReturnBlock`*** is included in the **`Resource_Clear`** message when it is present in the Return Result component. The STR-Connection is cleared and the TDISC Timer is canceled.

- If the T1 timer is running, the STR-Connection is cleared and the TDISC Timer is canceled. The local switch awaits the receipt of the **`Call_Info_To_Resource`** message from the SCP. A **`Resource_Clear`** message is sent to the SCP in a response package with a ***`ClearCause`*** parameter value of `userAbandon`. The **`Call_Info_To_Resource`** message is discarded.
- If the T1 timer expires, a fatal application error is detected.

Figures 2-27 and 2-28 provide examples of caller abandon during an active STR-Connection.

Figure 2-27
Caller abandon during an active STR-Connection using SS7 signaling

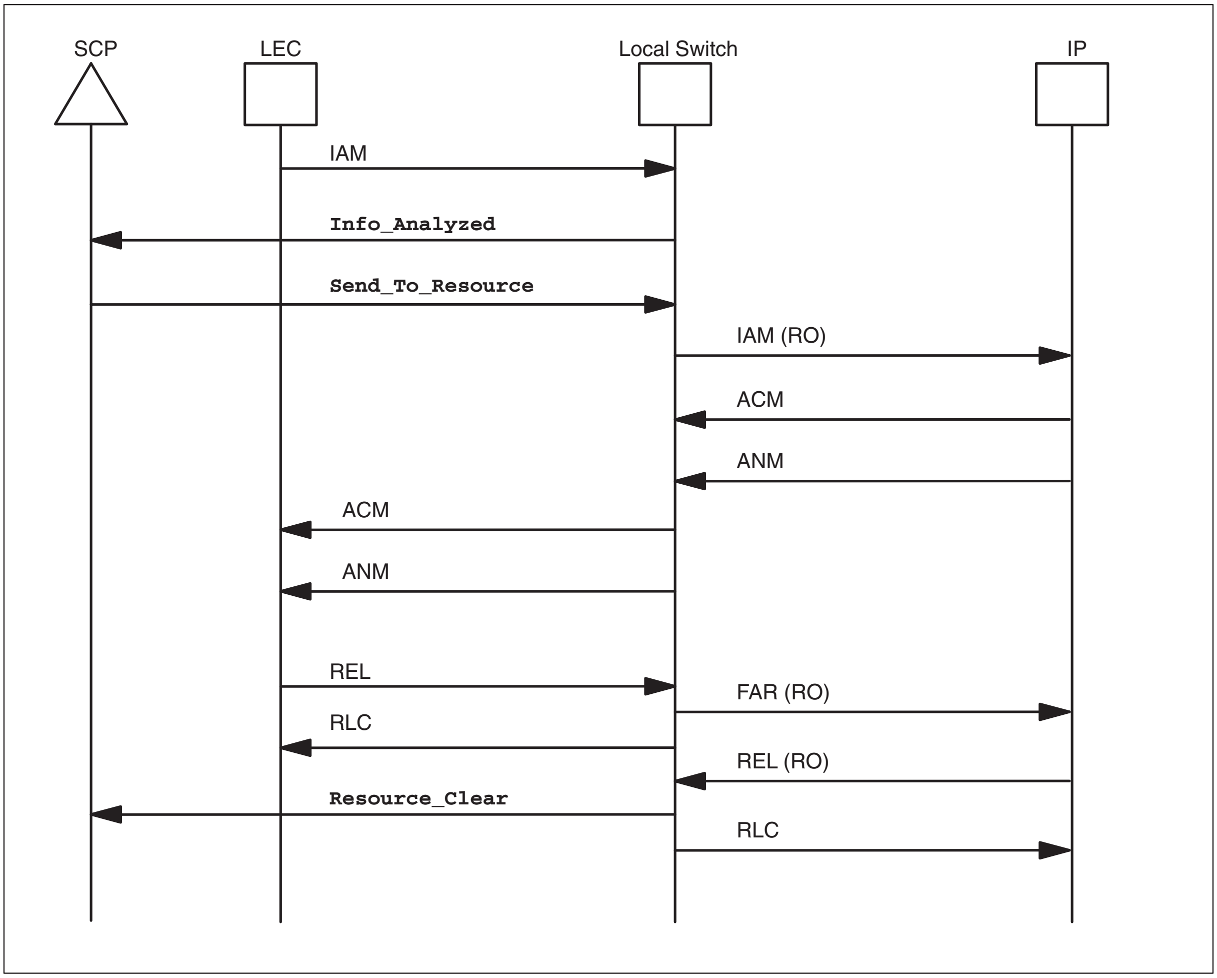
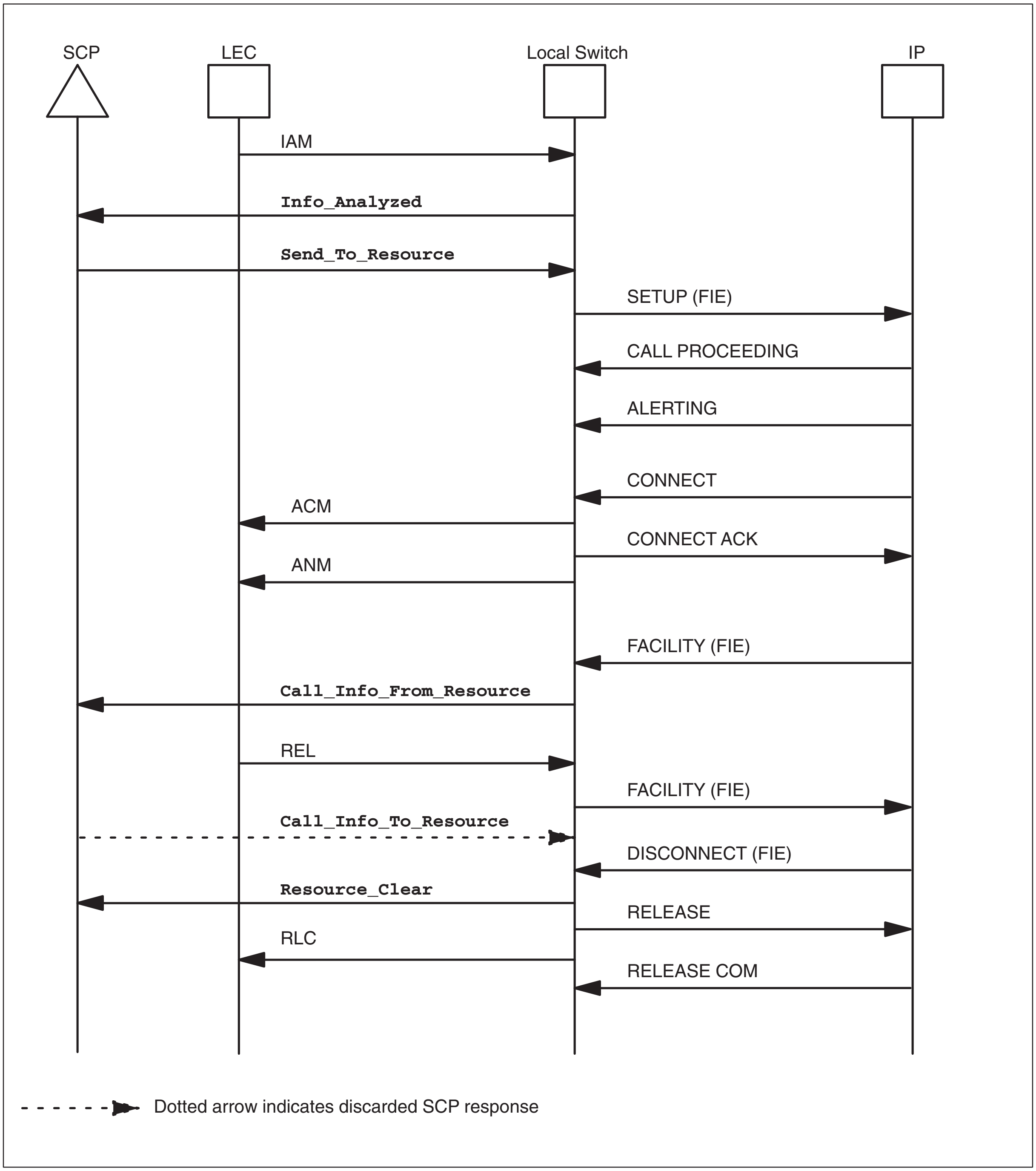


Figure 2-28
Caller abandon during an active STR-Connection with T1 timer running using ISDN signaling



If the TDISC timer expires before the IP responds to the FACILITY or FAR message with the cancelIPResource operation, the following actions are performed:

- If the T1 timer is not running, a **Resource_Clear** message in a response package is sent to the SCP with a ***ClearCause*** parameter value of ipTimeout. The switch clears the STR-Connection.
- If the T1 timer is running, the switch clears the STR-Connection. The local switch awaits the receipt of the **Call_Info_To_Resource** message from the SCP. When the message is received, it is discarded. A **Resource_Clear** message in a response package is sent to the SCP with the ***ClearCause*** parameter value of ipTimeout.
- If the T1 timer expires, a fatal application error is detected.

Figures 2-29 and 2-30 provide examples of caller abandon during an active STR-Connection.

Figure 2-29
TDISC timer expires during caller abandon, with T1 timer not running, using ISDN signaling

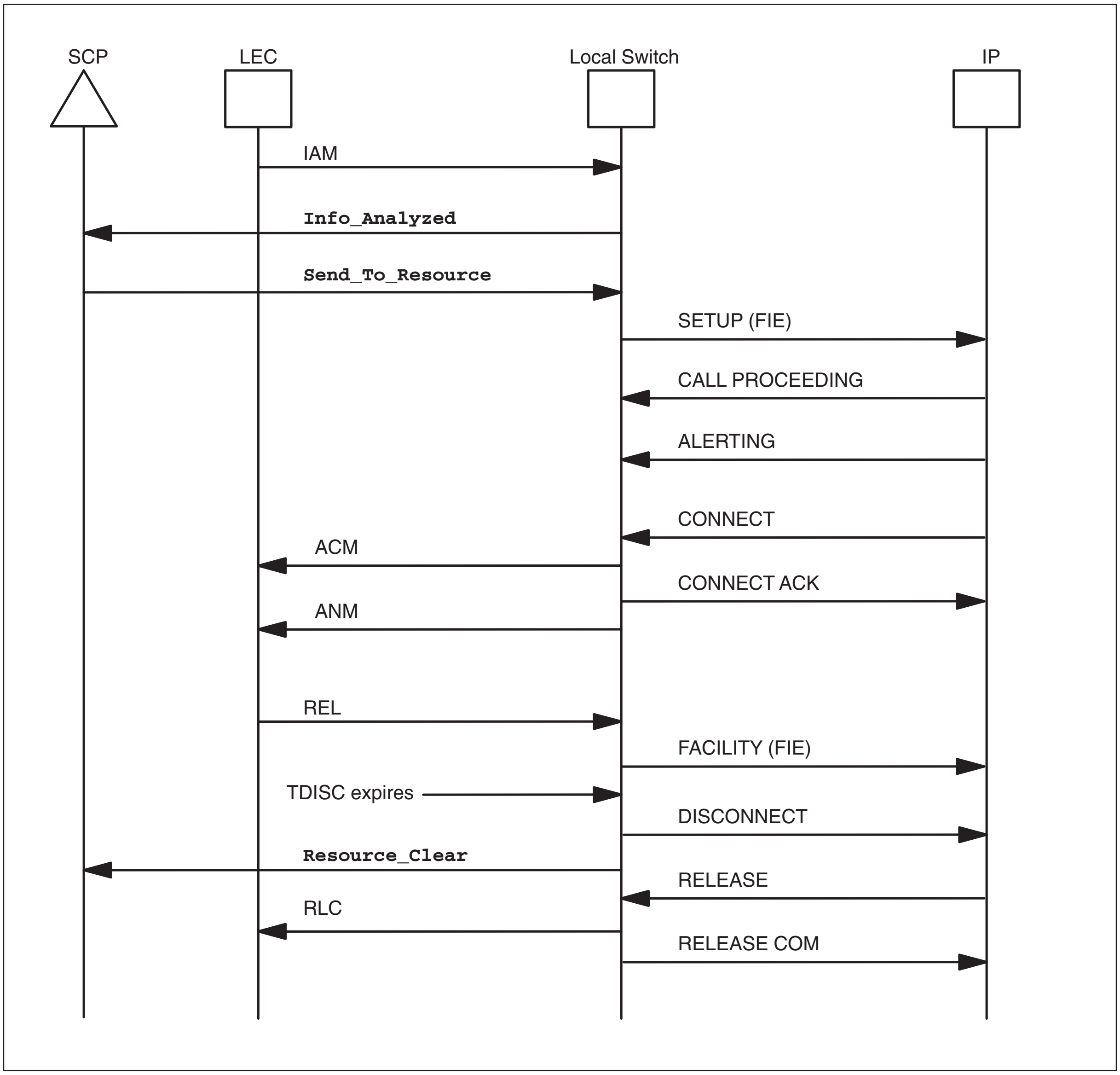
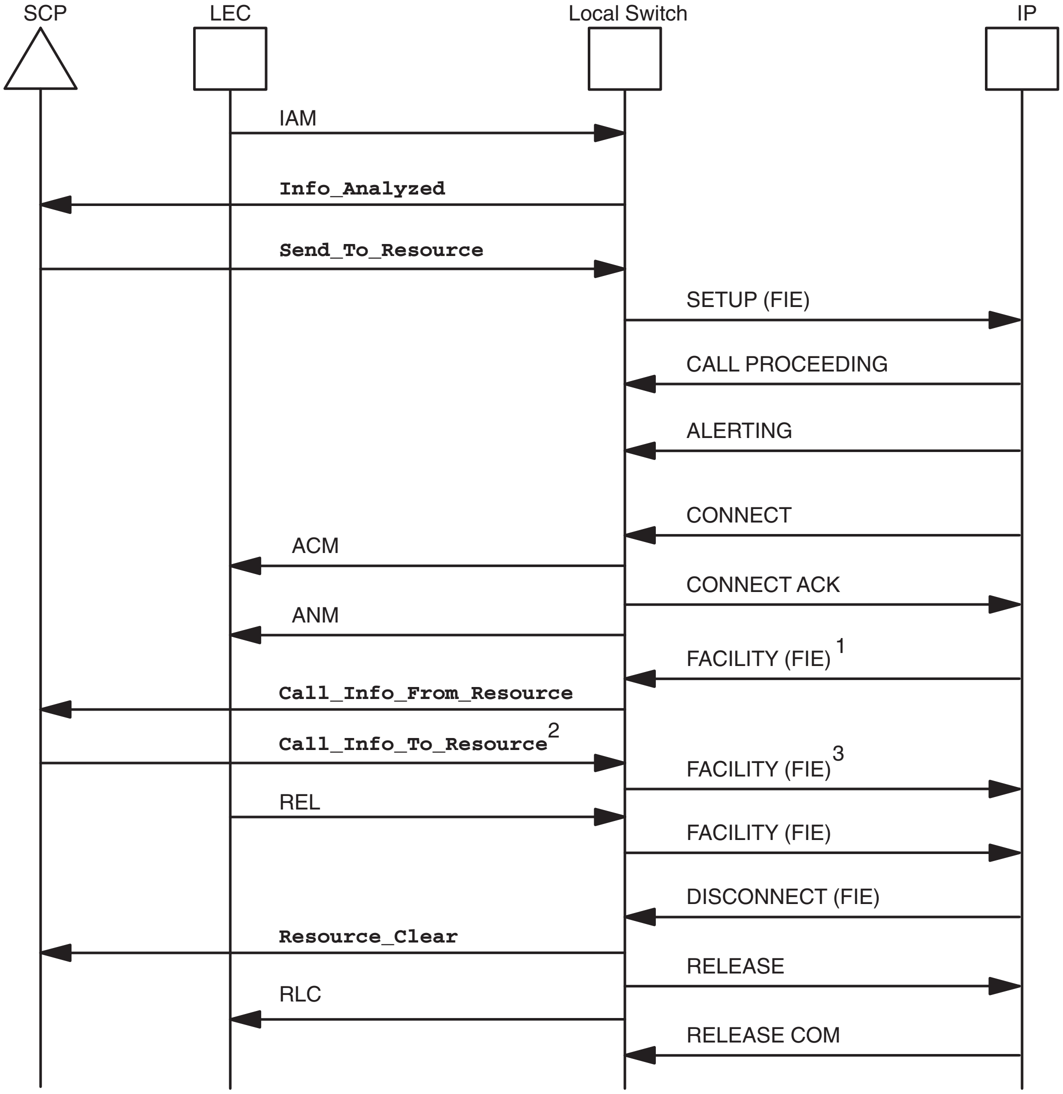


Figure 2-30 provides an example of a service based on 1129-Style IP interactions which bills the subscriber even though the subscriber may have abandoned the call before completion of the entire service, “Early billing”. For example, a subscriber may subscribe to a stock quoting service where it would be natural for the subscriber to abandon the call when a stock of interest has been quoted rather than waiting for the entire message to play. Figure 2-30 illustrates the following scenario:

- The call triggers at the ***Info_Analyzed*** TDP and the SCP determines that the call should be routed to an IP for one of several IP-based services.

- The IP determines the subscriber's selected service (refer to item 1 in Figure 2-30). In this example, the subscriber has selected a stock-quoting service.
- This service selection is propagated back to the SCP through the FACILITY and **`Call_Info_From_Resource`** messages.
- Based on the service selected, the SCP sends a **`Call_Info_To_Resource`** message to the switch, including the `amaDigits` extension parameter (refer to item 2 in Figure 2-30). This parameter contains the billing information which the switch updates into its billing records (CDR). This message also includes the ***`StrParameterBlock`*** which contains information on how the IP is to deliver the stock quoting service.
- The switch delivers the ***`StrParameterBlock`*** to the IP through the FACILITY message (refer to item 3 in Figure 2-30). It is only at this point the IP finally delivers the stock quoting service to the subscriber. Since the subscriber billing information has already been delivered to the switch, the subscriber is still billed even though they abandon the call before completion of the service.

Figure 2-30
Early billing during caller abandon using ISDN signaling



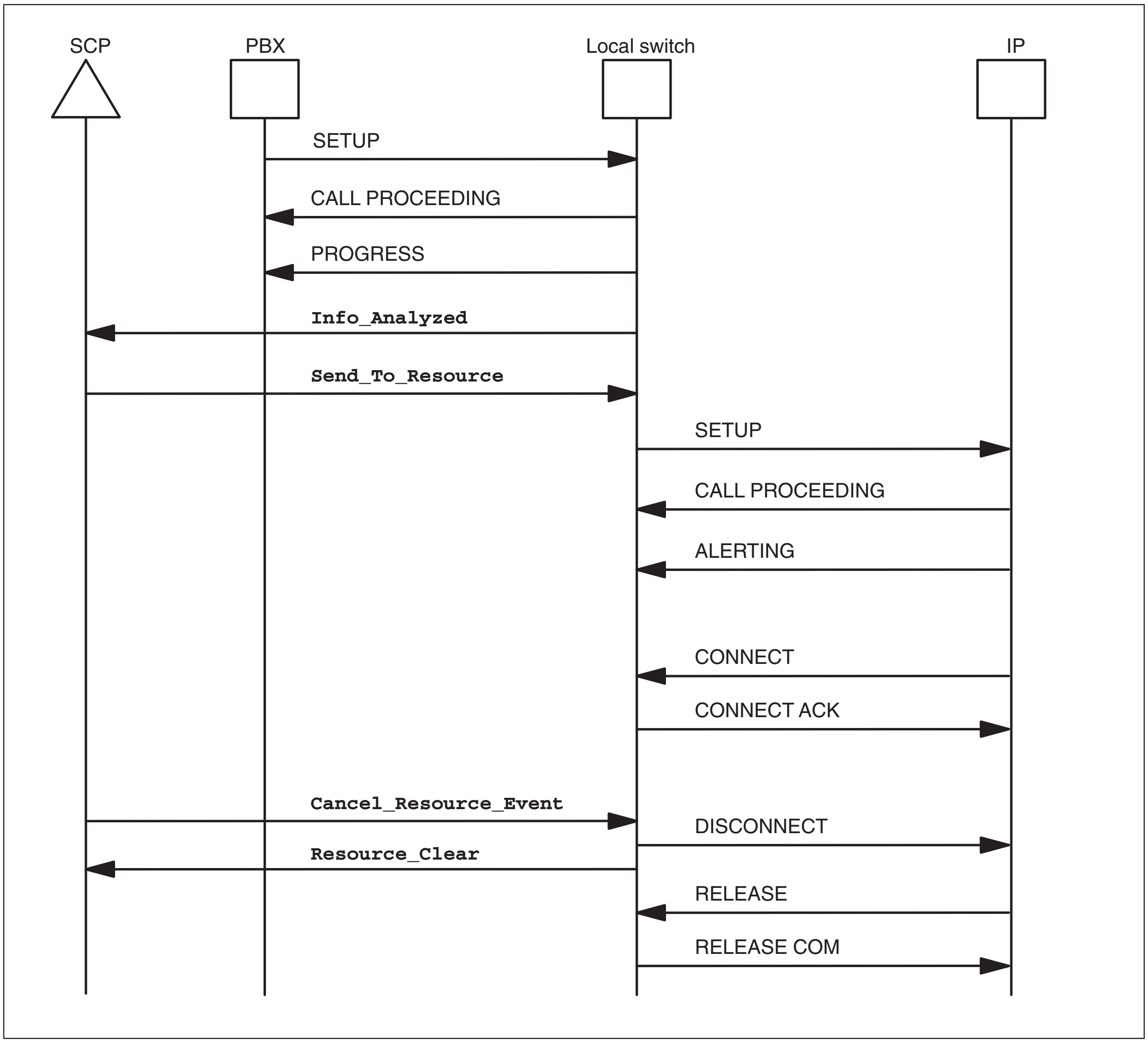
1. Subscriber has selected the menu option (for example, 1 for stock quotes). This FACILITY message contains a Return Result component with the menu choice so the SCP can determine how to bill the call.
2. This Call_Info_To_Resource message includes billing information (amaDigits extension parameter) to update switch billing.
3. This FACILITY message contains the FlexParameterBlock to instruct the IP to deliver the service (stock quotes).

Cancel_Resource_Event for CONNECT_ONLY

During a STR-Connection, the SCP may send a **Cancel_Resource_Event** message to the local switch requesting termination of the STR-Connection. Since the exchange of data is not supported for the CONNECT_ONLY IPI, when this occurs, a **Resource_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of `resourceCanceled`. The STR-Connection is cleared and the TSTRC Timer is canceled when necessary. The call is not cleared toward the calling user.

Figure 2-31 provides an example of the call clearing performed for this CONNECT_ONLY IPI scenario.

Figure 2-31
Switch-initiated call clearing for a Cancel_Resource_Event message using ISDN signaling



Cancel_Resource_Event for CONNECT_1129_STYLE

During a STR-Connection, the SCP may send a **Cancel_Resource_Event** message to the local switch requesting termination of the STR-Connection.

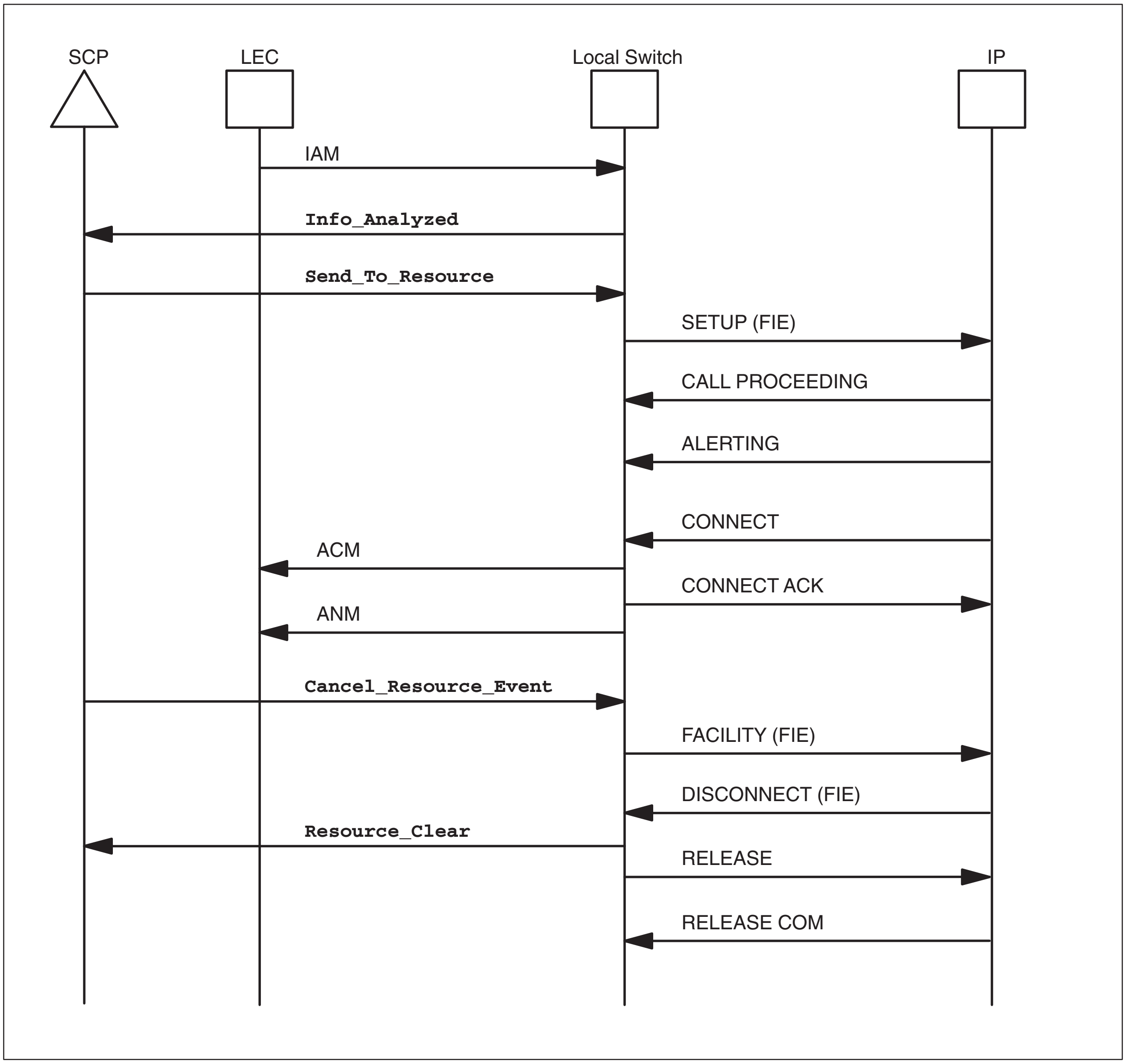
When this occurs, call clearing toward the local IP is initiated, and a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `resourceCanceled`. The STR-Connection is cleared. The call is not cleared toward the calling user.

When the **`Cancel_Resource_Event`** message is received during an active STR-Connection, an ISDN FACILITY or SS7 FAR message containing an Invoke component with a `cancelIPResource` operation is sent to the IP. The TDISC timer is started, and the TSTRC timer is canceled.

The IP is expected to respond to the `cancelIPResource` operation with a DISCONNECT or REL message containing a Return Result component. A **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `resourceCanceled`. The ***`IPReturnBlock`*** is included in the **`Resource_Clear`** message when it is present in the Return Result component. The STR-Connection is cleared and the TDISC timer is canceled. The call is not cleared toward the calling user.

Figure 2-32 provides an example of the call clearing performed with the **`Cancel_Resource_Event`**.

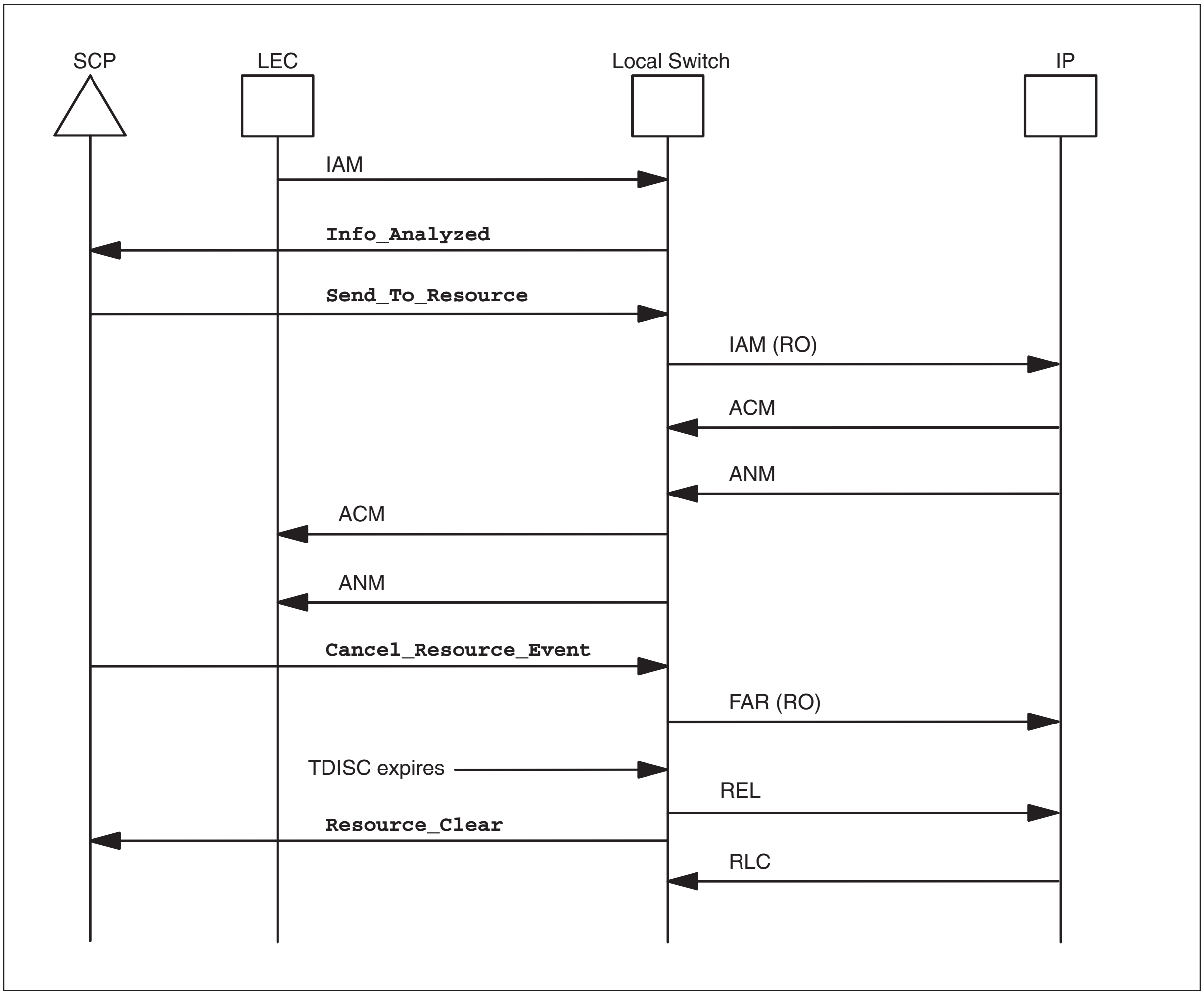
Figure 2-32
STR-Connection clearing with the Cancel_Resource_Event using ISDN signaling



If the TDISC timer expires before the IP responds to the FACILITY message with the `cancelIPResource` operation, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The switch clears the STR-Connection. The call is not cleared toward the calling user.

Figure 2-33 provides an example of the TDISC timer expiring during call clearing with the **`Cancel_Resource_Event`**.

Figure 2-33
TDISC timer expires during clearing with the Cancel_Resource_Event using SS7 signaling



The **`Cancel_Resource_Event`** message is ignored when it is received as a response to a **`Resource_Clear`** message. If the **`Cancel_Resource_Event`** message is received at any other time when it is not expected, an unxpected message fatal application error is detected. A CAIN200 log is generated and AINF treatment is provided. An **`Application_Error`** message is reported to the SCP with the ***`ErrorCause`*** parameter set to `unexpectedMessage`.

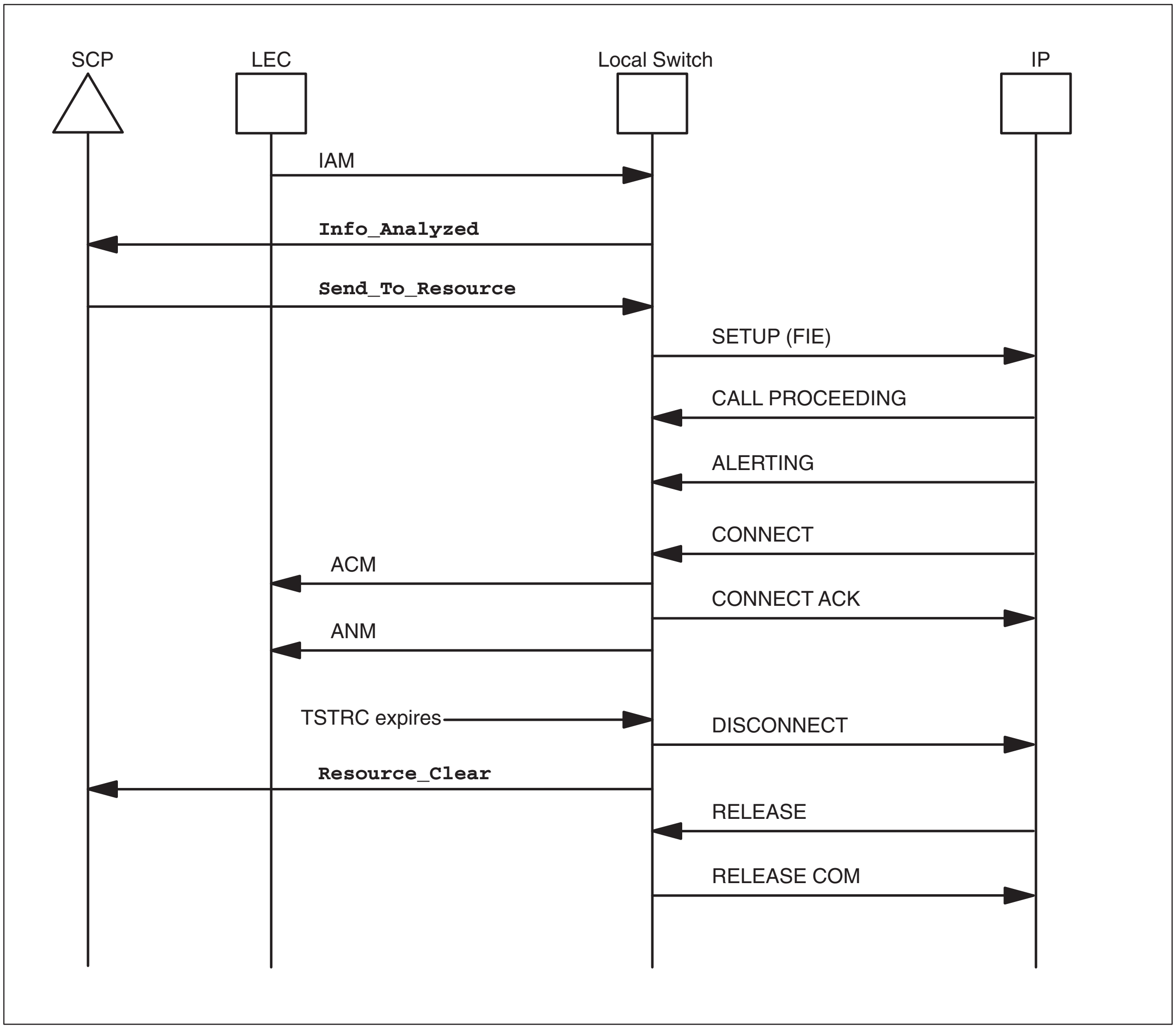
TSTRC timer

The TSTRC timer (STR-Connection Timer) provides a maximum time limit for a STR-Connection to an IP. It is started when the IP answers and is canceled when the STR-Connection is cleared, either by the switch or the IP. If the TSTRC timer expires, the following actions are performed:

- When the T1 timer is not running, a **`Resource_Clear`** mesage in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The call is not cleared toward the calling user.
- When the T1 timer is running, the switch awaits the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The call is not cleared toward the calling user.
- If the T1 timer expires before the SCP response is received, a fatal application error is detected.

Figure 2-34 provides an example of the TSTRC timer expiring during an STR-Connection without the T1 timer running.

Figure 2-34
TSTRC timer expires during a STR-Connection, T1 timer not running, using ISDN signaling



Establishing a STR-Connection to a remote IP

As with local IP connections, triggering in the terminating call model is not supported during remote IP connections. The originating and terminating call models are disabled at both the intermediate and remote switches while a STR-Connection is active.

CONNECT_ONLY call establishment at the local switch

If the IPI is CONNECT_ONLY when the switch receives a **Send_To_Resource** message (with a ***DestinationAddress*** parameter) from the SCP, switch call processing translates the ***DestinationAddress*** and determines a routing list to the appropriate IP.

Once the local switch identifies a route index, an attempt to establish a connection to the remote IP is made. The local switch selects a trunk group from the route list and attempts to locate an idle trunk member within the group. If no idle trunk members are present, the local switch route advances to the next available trunk group in the route list.

On the local switch, a **`Resource_Clear`** message is sent in to the SCP in a response package with a ***`ClearCause`*** value of `abort` if the PRI or SS7 IMT selected is not provisioned with the IPTRUNK option.

If the local switch is unable to locate an idle trunk member within the route list, a **`Resource_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** value of `channelsBusy`.

Once an idle trunk member is located, the local switch constructs an Initial Address Message (IAM) and sends it to the remote switch. The local switch may be connected directly to the remote switch, or there may be one or more intermediate switches between the two switches.

The Called Party parameter is built using the address contained in the ***`DestinationAddress`*** parameter. As stated earlier, the exchange of data between the SCP and remote IP thru the switch is not supported for the CONNECT_ONLY IPI. Therefore, the contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameter is discarded and the RO parameter is not included in the outgoing IAM.

It is important to note that the IAM is built using the existing UCS DMS-250 call processing logic. Therefore, the contents of the IAM may be influenced by switch software such as table RTEATTR or non-standard routing.

When the long call duration timer is enabled for the terminating SS7 IMT agent, the timer is started using the value provisioned in table TRKGRP1. If the timer expires before an Answer Message (ANM) is received from the remote switch, a **`Resource_Clear`** is sent in to the SCP in a response package with a ***`ClearCause`*** value of `strCancelled`. The call is then sent to treatment by the long call duration feature.

CONNECT_1129_STYLE call establishment at the local switch

A **`Send_To_Resource`** message containing the ***`DestinationAddress`*** parameter is used to initiate a STR-Connection. The message is processed according to the rules previously described in this document.

Once the message is processed and a route index is identified, the local switch attempts to establish a connection to the remote switch. It is important to note that existing UCS DMS-250 software is used to establish the connection. Therefore, inswitch features may interact with the STR-Connection. Several examples are listed below:

- During the **Authorize_Termination** PIC, the UCS DMS-250 switch performs bearer capability screening on the terminating trunk using table BCCOMPAT. When bearer capability screening fails, the switch route advances to the next available trunk group in the route list.
- During the **Present_Call** PIC, the UCS DMS-250 switch constructs the IAM to be sent to the remote switch. Delivery of the Charge Number, Originating Line Information (OLI), and Calling Party Number parameters is controlled using the CPIXFER option of table TRKGRP.

Using the route list identified by the ***DestinationAddress*** parameter, the local switch selects a trunk group from the route list and attempts to locate an idle trunk member within the group. If no idle trunk members are present, the local switch route advances to the next available trunk group in the route list.

If the local switch is unable to locate an idle trunk member within the route list, a **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of channelsBusy.

If the selected trunk group is not a PRI or SS7 IMT trunk, or if it is a PRI trunk without the IPTRUNK option provisioned, a **Resource_Clear** message is sent to the SCP in a conversation package with a ***ClearCause*** parameter value of abort.

Once an idle trunk member is located, the local switch constructs an IAM and sends it to the remote switch. The following parameters are built using the data provided in the **Send_To_Resource** message:

- Called Party Number – This parameter is built using the address contained in the ***DestinationAddress*** parameter.
- Remote Operations – This parameter shall contain an Invoke component with an operation of sendToIPResource. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the local switch.

Once the IAM is sent, the local switch waits for a response message from the remote switch.

When the long call duration timer is enabled for the terminating IMT agent, the timer is started using the value provisioned in table TRKGRP1. The long call duration timer is provided by an inswitch feature, and is similar in behavior to the T_No_Answer timer. If the long call duration timer expires before an Answer Message (ANM) is received by the local switch, the following actions are performed:

- A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `strCanceled`.
- The long call duration feature is activated in order to handle the expired timer.

Call establishment at the intermediate switch

Existing UCS DMS-250 software establishes the call at the intermediate switch. The intermediate switch:

- processes the incoming IAM containing the RO parameter

  Note: For the CONNECT_ONLY IPI, the incoming IAM message does not contain an RO parameter.

- translates the Called Party parameter and identifies a route index
- selects a terminating trunk group and identifies an idle trunk member
- detects an invalid termination (an SS7 IMT without the IPTRUNK option), this results in a **`Resource_Clear`** message being sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`
- constructs an IAM containing the unmodified RO parameter and sends it to the Remote switch

  Note: For the CONNECT_ONLY IPI the RO parameter is not present

- can potentially trigger again

Call establishment at the remote switch

The remote switch uses the existing UCS DMS-250 software to:

- process the incoming IAM containing the RO parameter

  Note: For the CONNECT_ONLY IPI, the incoming IAM message does not contain an RO parameter.

- translate the Called Party parameter and identify a route index
- select a terminating trunk group and identify an idle trunk member
- construct a SETUP or IAM message and send it to the Remote IP

— For ISDN signaling a SETUP message is built using the data provided in the incoming IAM. It includes a Called Party Information element built using the address contained in the Called Party Number parameter of the incoming IAM and a Facility Information element (FIE) constructed using the information contained in the RO parameter of the incoming IAM. The FIE is composed of an Invoke component with an operation of `sendToIPResource`. The contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are placed into the Invoke component without modification by the remote switch.

Note: For the CONNECT_ONLY IPI the RO parameter is not present in the incoming IAM message received by the intermediate switch. Therefore, with ISDN signaling an FIE is not added to the outgoing SETUP message, and with SS7 signaling an RO parameter is not added to the outgoing IAM message.

— For SS7 signaling an IAM message is built by copying the RO parameter contained in the incoming IAM message and sending it in the outgoing IAM message to the remote IP without modification.

- Once the SETUP or IAM message is sent, the remote switch waits for a response message from the remote IP.

Figure 2-35 provides a CONNECT_ONLY IPI example of call establishment by an SS7 FGD originator, using ISDN signaling to a remote IP.

Figure 2-35
SS7 call establishment to a remote IP using ISDN signaling

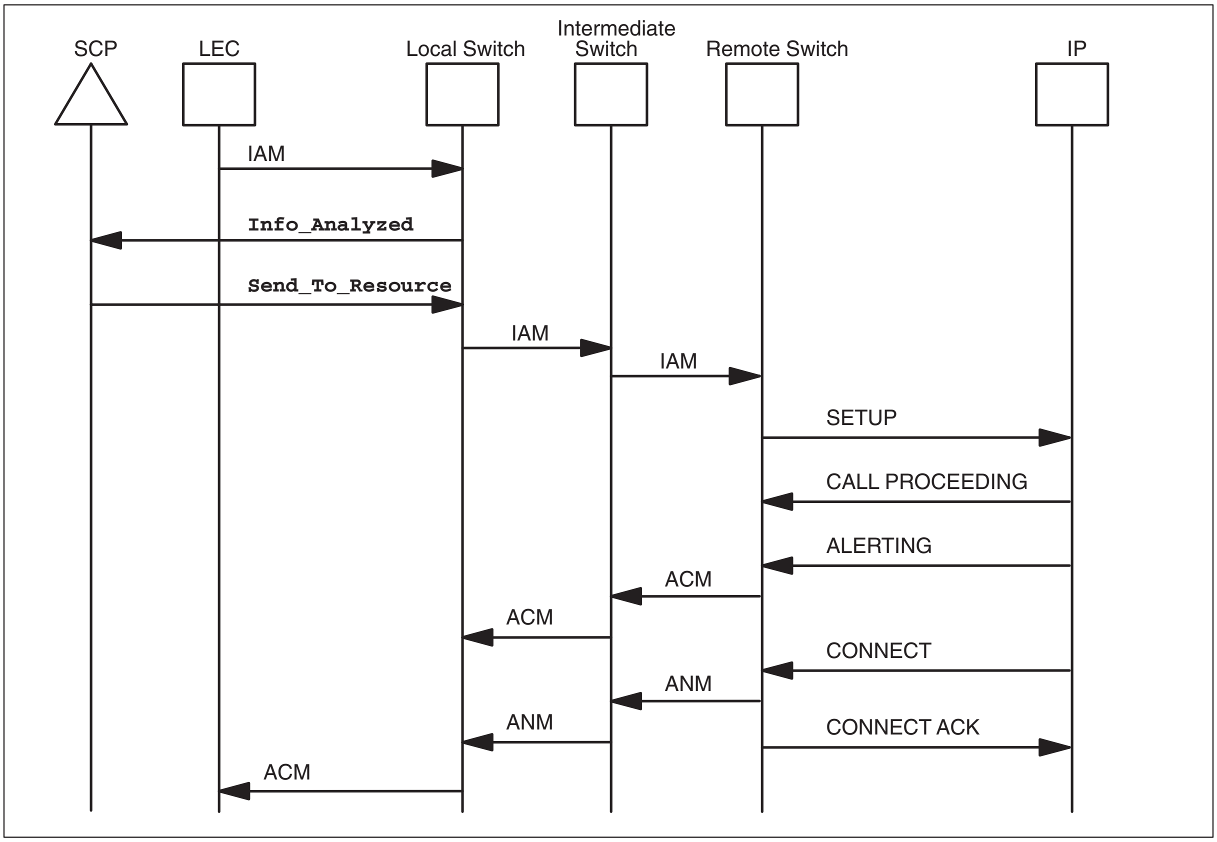
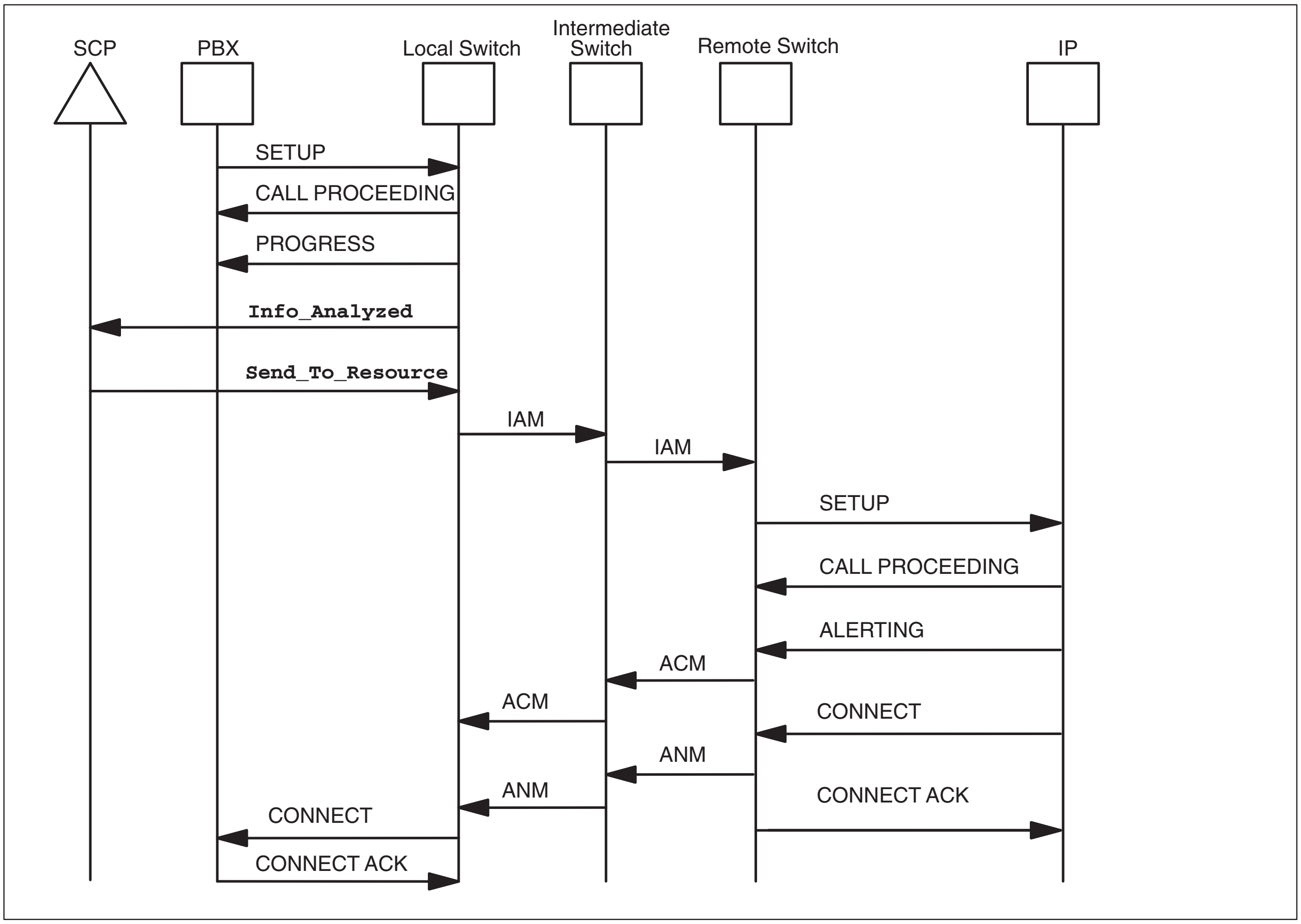


Figure 2-36 provides a CONNECT_ONLY IPI example of call establishment by a PRI originator, using ISDN signaling to a remote IP.

Figure 2-36
PRI call establishment to a remote IP using ISDN signaling



Figures 2-37 and 2-38 provide CONNECT_1129_STYLE IPI examples of call establishment with answer indication, using ISDN signaling to a remote IP.

Figure 2-37
PRI call establishment to a remote IP with Answer Indication, using ISDN signaling

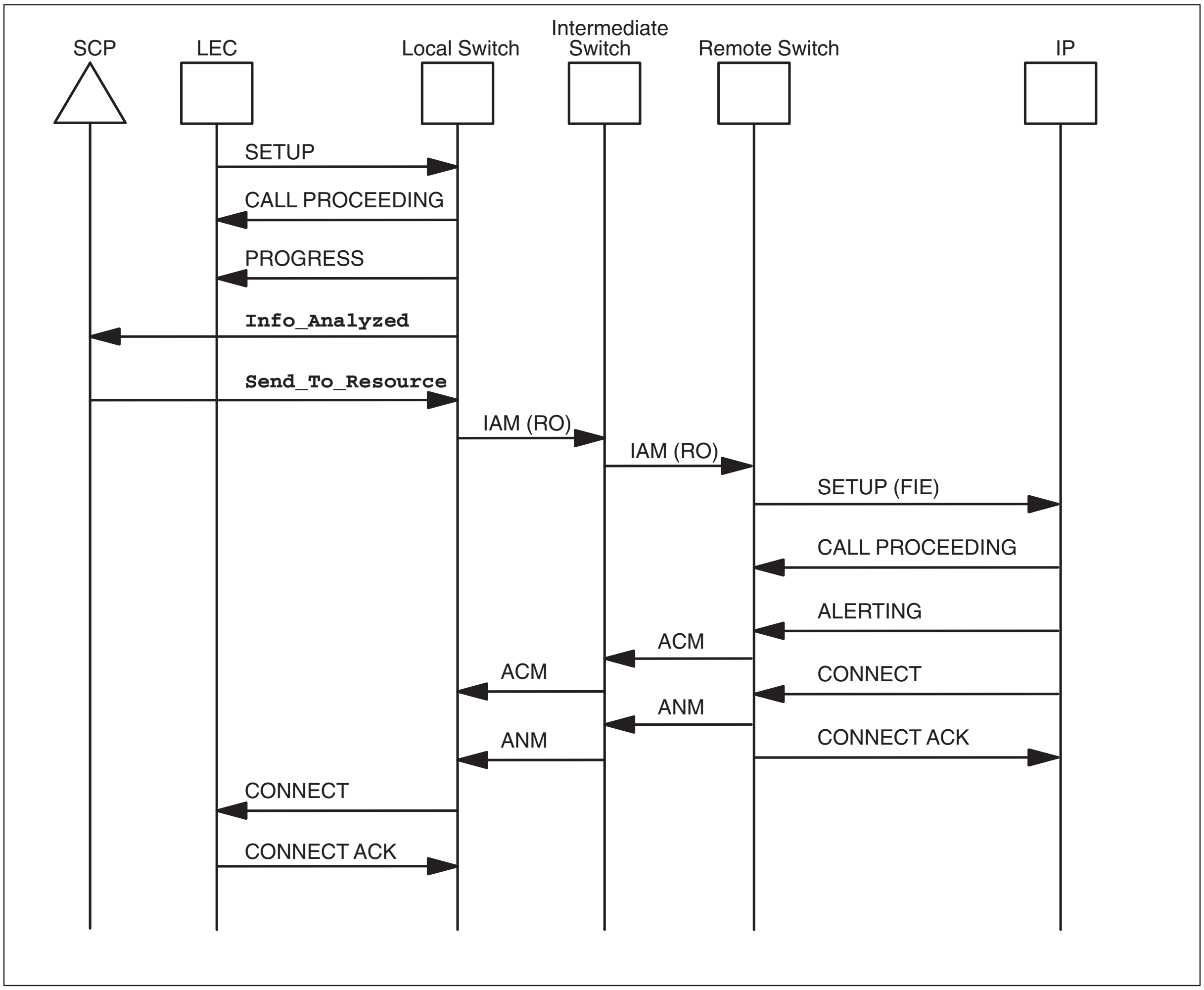
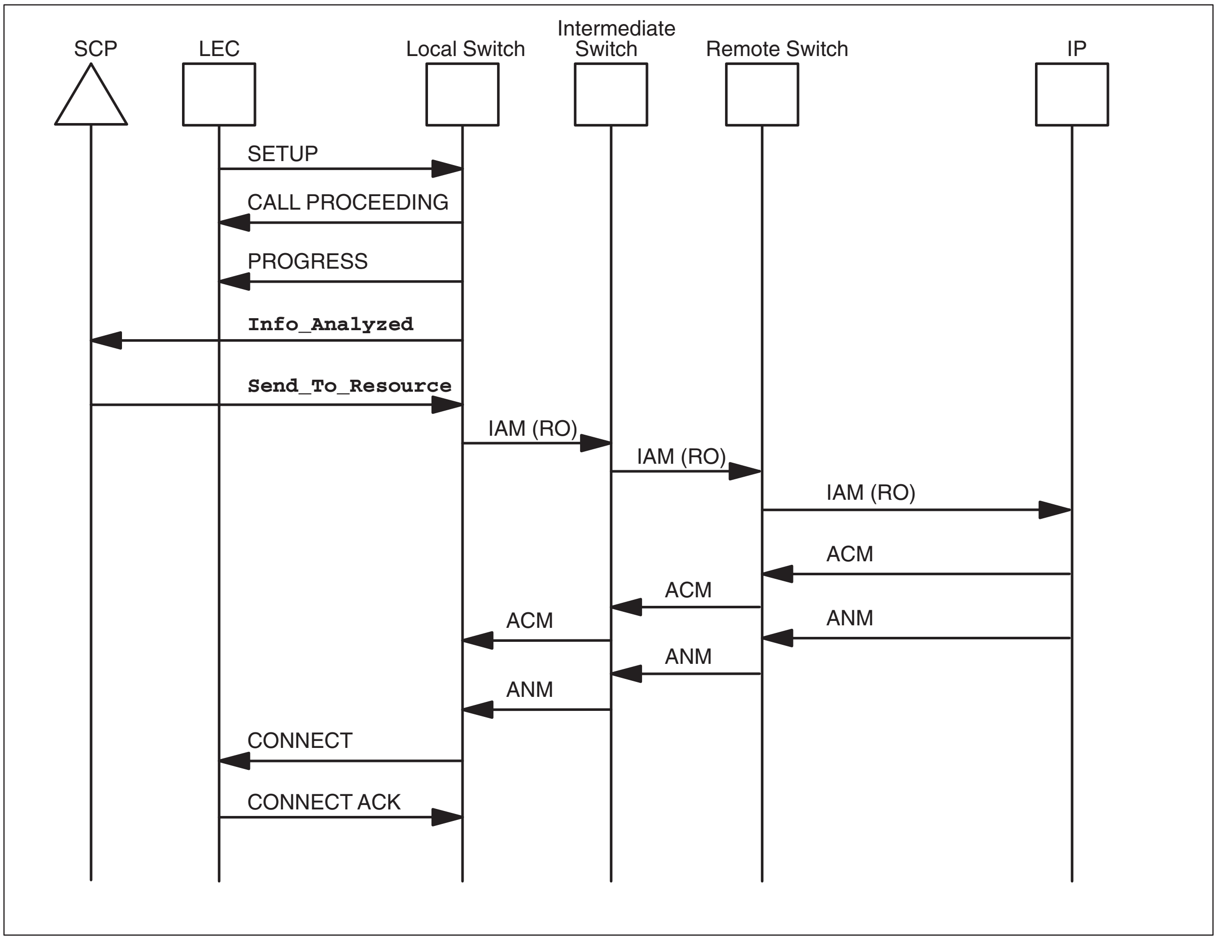


Figure 2-38
PRI call establishment to a remote IP with Answer Indication, using SS7 signaling



Call Proceeding Message

The CALL PROCEEDING message indicates that the remote IP using ISDN signaling has initiated the call establishment requested by the remote switch. This message is NOT passed to the intermediate or local switch.

Alerting and Address Complete Message (ACM)

The ISDN ALERTING message indicates that the remote IP using ISDN signaling has been alerted to the call. When this message is received by the remote switch, an SS7 ACM is constructed and sent to the intermediate and local switches.

Note: This differs from Bellcore’s *GR-1129-CORE* requirement which states that no message should be sent toward the user as a result of the ALERTING message.

The SS7 ACM message indicates that the remote IP using SS7 signaling has been alerted to the call. When this message is received by the remote switch, it is passed without modification to the intermediate and local switches.

When the intermediate switch receives an ACM, the message is passed along to the local switch.

When the local switch receives an ACM, no message is sent to the originating switch.

CONNECT and Answer Message (ANM)

The ISDN CONNECT or SS7 ANM message indicates that the remote IP has answered the call. When this message is received by the remote switch, an ANM is constructed and sent to the intermediate switch, then passed to the local switch. Timer TSTRC is started for CONNECT_1129_STYLE IPI connections.

Note: The timers (TSTRC and TDISC) used during a CONNECT_1129_STYLE STR-Connection are only started at the switch which is directly connected to the IP.

When the intermediate switch receives an ANM, the message is passed along to the local switch.

At the local switch, the action taken upon receipt of an ANM depends upon both the originating agent and the presence or absence of the SS7 FGD and SS7 IMT ***`AnswerIndicator`*** parameter in the **`Send_To_Resource`** message.

For SS7 FGD and SS7 IMT originating agents, the following actions are performed when the **`Send_To_Resource`** message is received without the ***`AnswerIndicator`*** parameter:

- If an ACM has not already been sent to the originating switch, an ACM is sent with an optional backward call indicator parameter indicating “user-network interaction”.
- If an ACM has already been sent, but it did not indicate “user-network interaction”, a Call Progress message (CPG) is sent with an optional backward call indicator parameter indicating “user-network interaction”.
- No action is taken if an ACM or CPG has already been sent with an optional backward call indicator parameter indicating “user-network interaction”.

Tables 2-6 and 2-7 provide a description of these messages.

Table 2-6
Remote switch processing of ISDN setup messages received from the remote IP

Message	Description
CALL PROCEEDING	Indicates the remote IP has initiated the call establishment requested by the local switch. ***Note:*** This message is not passed to the intermediate or local switch.
ALERTING	Indicates that the remote IP has been alerted to the call. When the remote switch receives the ALERTING message, an ACM is built and sent to the intermediate and then passed on to the local switch. For SS7 originations, the incoming ACM is sent to the originating switch. For PRI originations, the contents of the incoming ACM are used to construct an ALERTING message that is sent to the originating switch.
CONNECT	Indicates that the remote IP has answered the call. When the remote switch receives the CONNECT message, an ANM is constructed and sent to the intermediate switch and then passed on to the local switch. For SS7 originations at the local switch, the ANM is not passed to the originating switch. For PRI originations at the local switch, the CONNECT message is not passed to the originating switch.

Table 2-7
Remote switch processing of SS7 setup messages received from the remote IP

Message	Description
ACM	Indicates the remote IP has initiated the call establishment requested by the local switch. This message is passed without modification to the intermediate and local switches.
ANM	Indicates that the remote IP has answered the call. When the remote switch receives the ANM message, an ANM is constructed and sent to the intermediate switch and then passed on to the local switch. The TSTRC timer is started for CONNECT_1129_STYLE IPI scenarios.

Figures 2-39 and 2-40 provide CONNECT_1129_STYLE IPI examples of call establishment to a remote IP, without answer indication.

Figure 2-39
SS7 call establishment to a remote IP without Answer Indication, using ISDN signaling

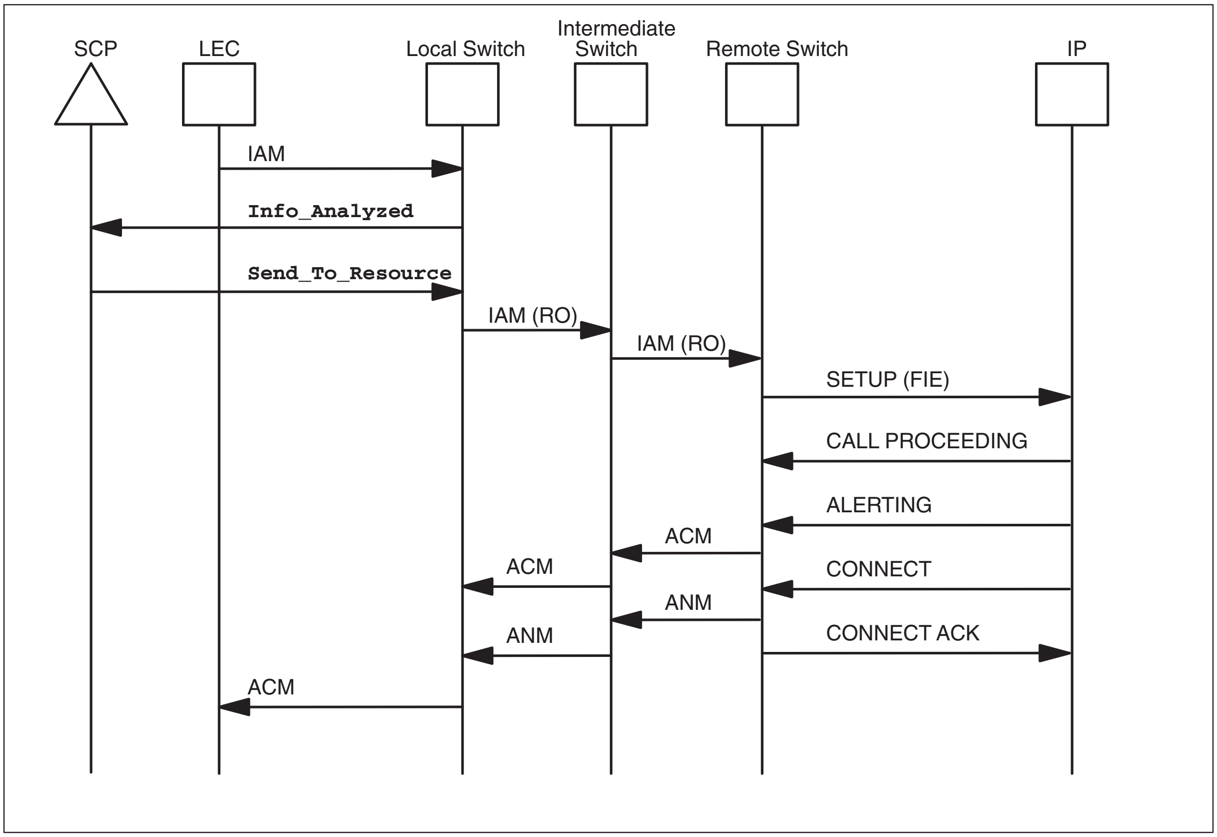
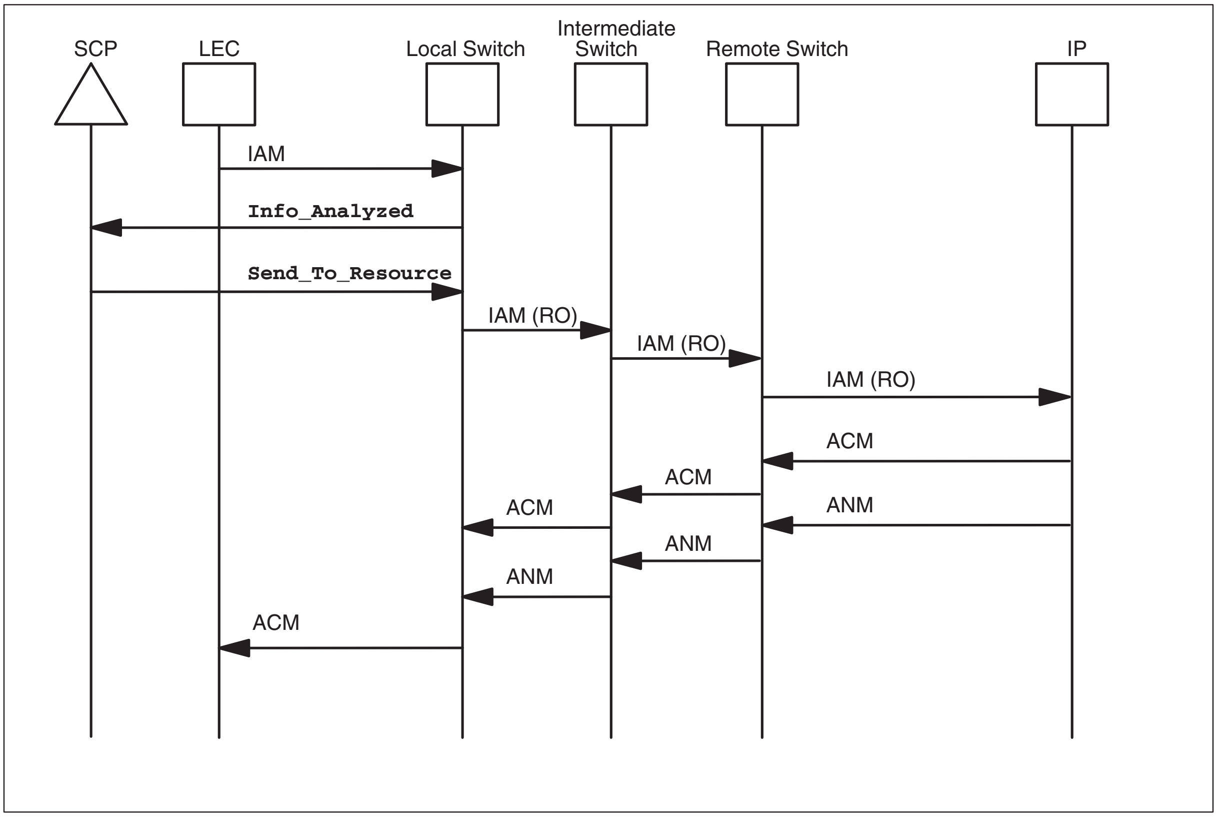


Figure 2-40
SS7 call establishment to a remote IP without Answer Indication, using SS7 signaling

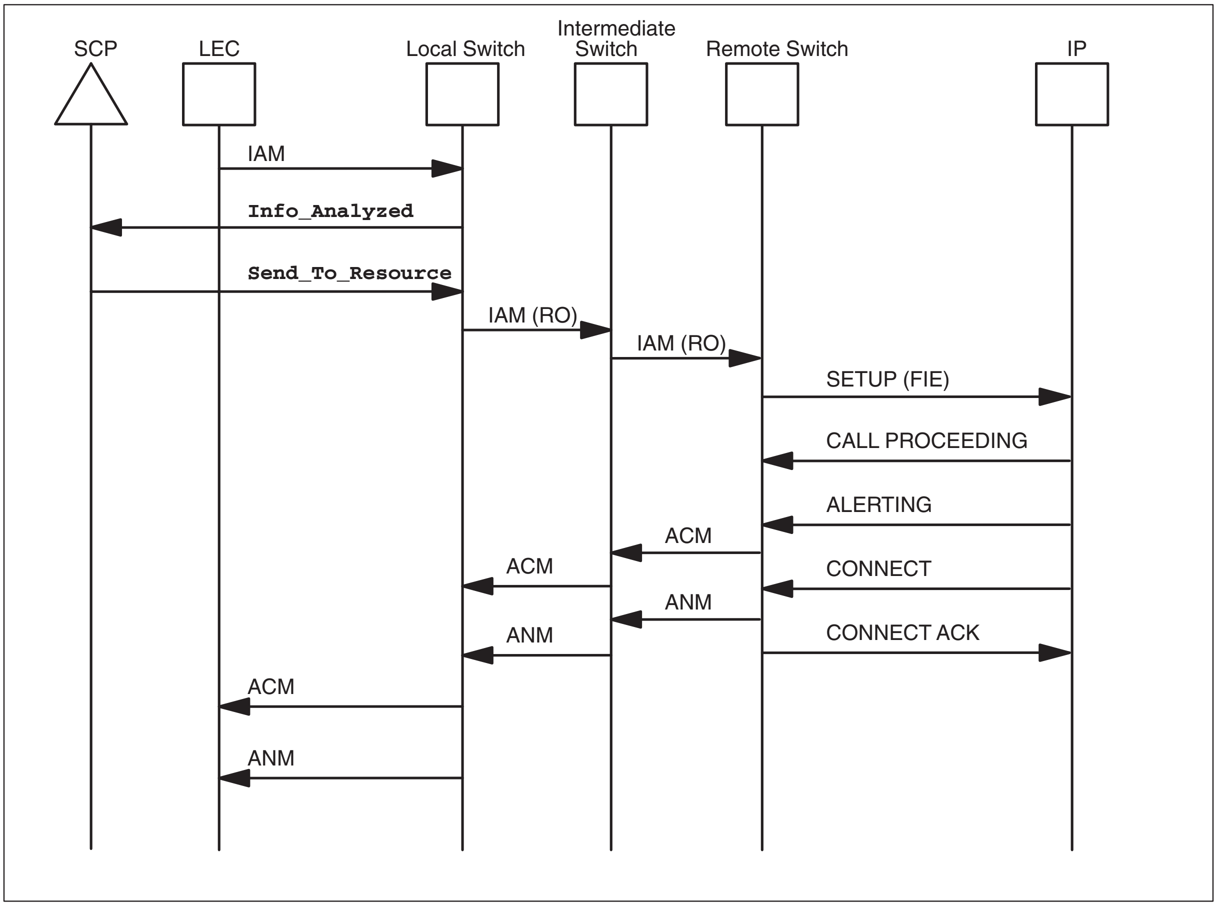


For SS7 FGD and SS7 IMT originating agents, the following actions are performed when the **`Send_To_Resource`** message is received with the ***`AnswerIndicator`*** parameter:

- If an ACM has not already been sent to the originating switch, an ACM is sent with an optional backward call indicator parameter indicating “user-network interaction”.
- An ANM is sent to the originating switch, if one has not previously been sent earlier in the call.

Figure 2-41 provides a CONNECT_1129_STYLE IPI example of call establishment by a SS7 originator, with answer indication, using ISDN signaling to a remote IP.

Figure 2-41
SS7 call establishment to a remote IP with Answer Indication, using ISDN signaling

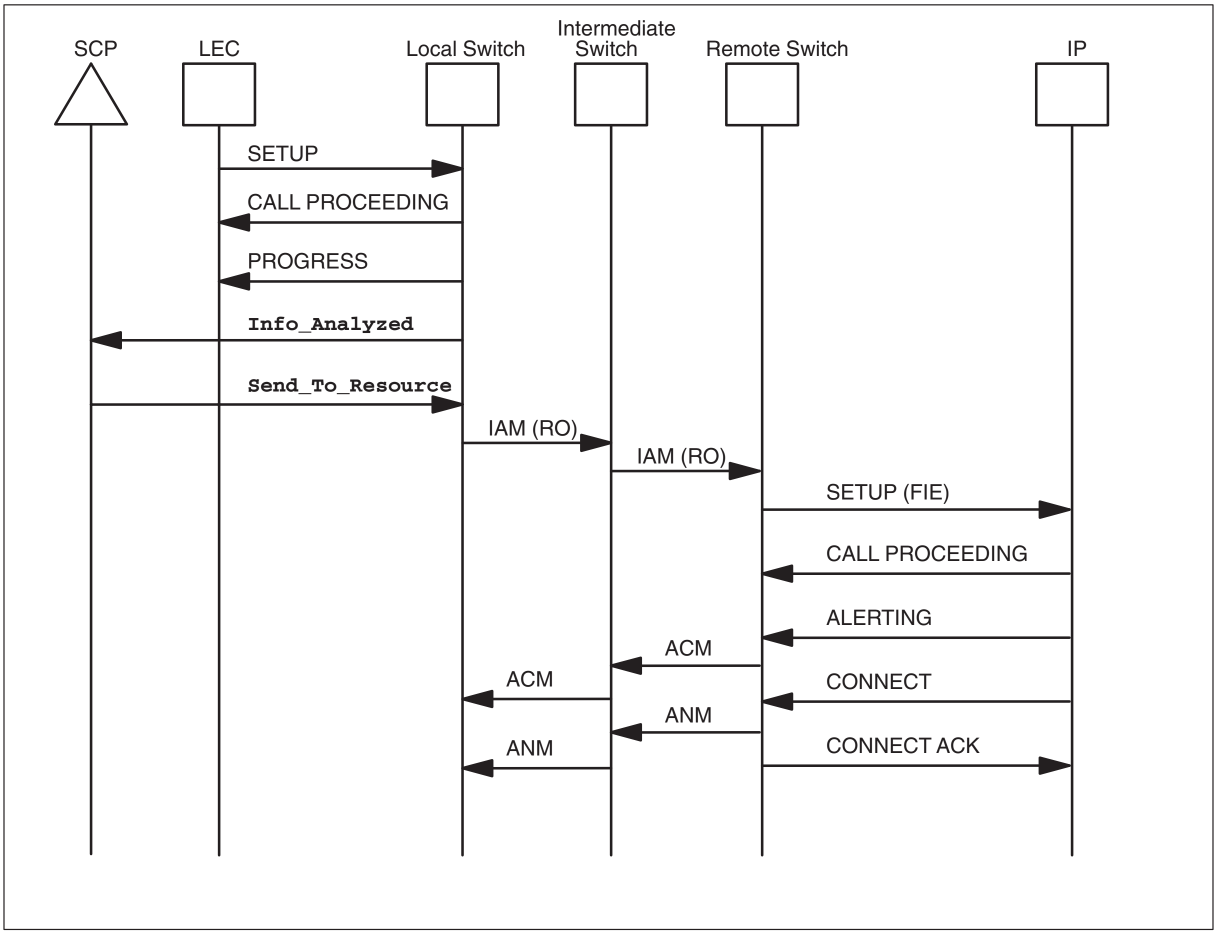


For PRI originating agencies, the following actions are performed when the **`Send_To_Resource`** message is received without the ***`AnswerIndicator`*** parameter:

- If no message beyond CALL PROCEEDING has been sent to the originating switch, a PROGRESS message with the progress indicator set to “inband information or appropriate pattern now available” is sent to the originating switch.
- If a CONNECT message has already been sent to the originating switch, no message is sent.

Figure 2-42 provides a CONNECT_1129_STYLE IPI example of call establishment without answer indication, using ISDN signaling to a remote IP.

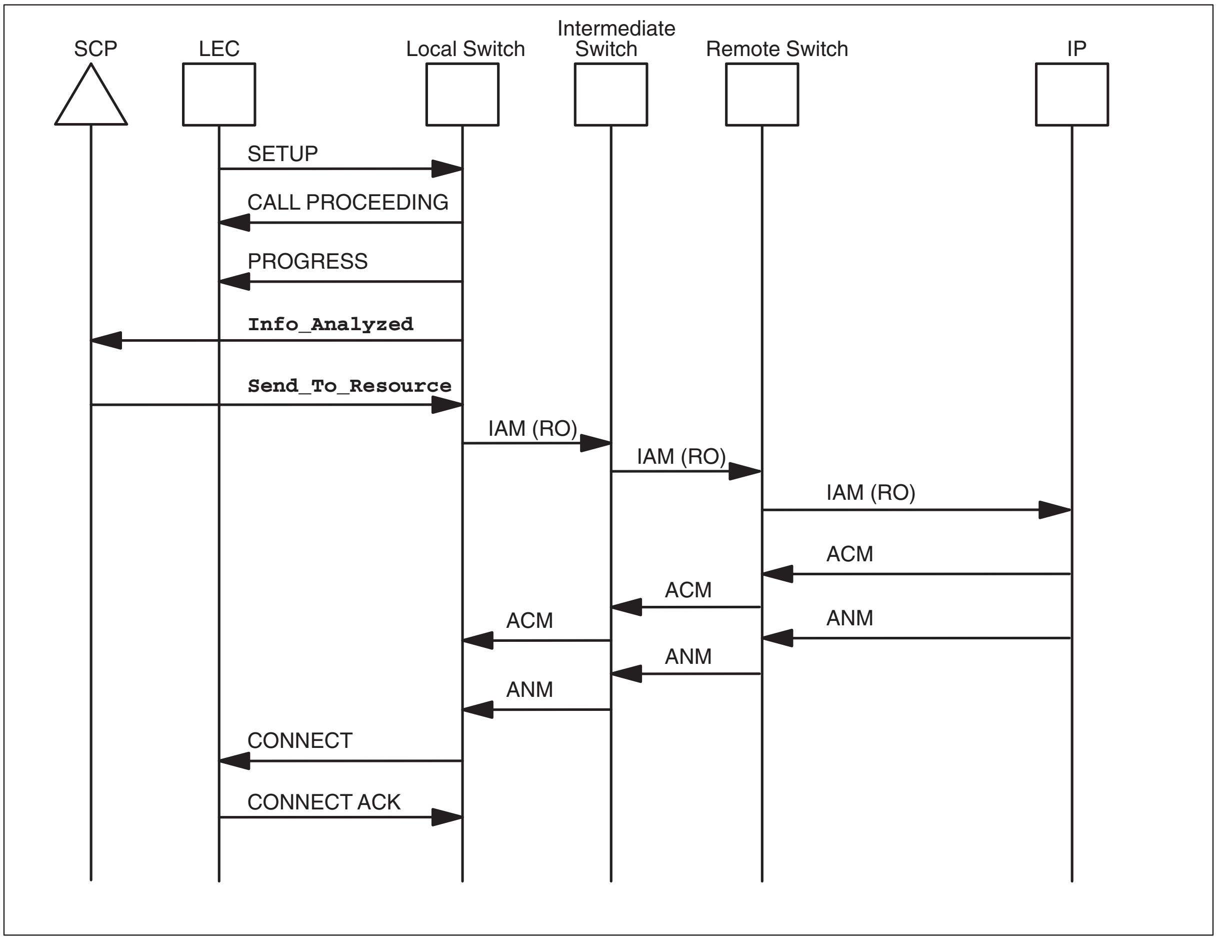
Figure 2-42
PRI call establishment to a remote IP without Answer Indication, using ISDN signaling



For PRI originating agencies, when the **`Send_To_Resource`** message is received with the ***`AnswerIndicator`*** parameter, a CONNECT message is sent to the originating switch, unless one has previously been sent to the originating switch.

Figure 2-43 provides a CONNECT_1129_STYLE IPI example of call establishment with answer indication, using SS7 signaling to a remote IP.

Figure 2-43
PRI call establishment to a remote IP with Answer Indication, using SS7 signaling



Signaling during an active STR-Connection to a local IP

A STR-Connection is considered active when an answer indication is received from the remote IP.

Signaling to exchange intermediate information during an active CONNECT_ONLY IPI STR-Connection is not supported. If the remote switch receives an ISDN FACILITY or SS7 FAR message, it is assumed to be a supplemental service and it is processed by the existing UCS DMS-250 software.

Since the exchange of information is not supported for CONNECT_ONLY, the remote switch is unable to notify the local switch that a supplemental service has been invoked. Therefore, the STR-Connection remains active at the local swirch which may result in undesired interactions with the supplemental service. The CONNECT_ONLY IPI supports the RLT and Billing Information supplemental services described below.

During an active CONNECT_1129_STYLE STR-Connection, the FACILITY and FAR messages allow the SCP and IP to exchange intermediate information.

While the STR-Connection is active to an IP with ISDN signaling, the remote switch may receive a FACILITY message containing an FIE with a Return Result component. The remote switch uses the information contained in the FIE to construct a FAR message containing a RO parameter, which is then sent to the intermediate switch and then to the local switch.

While the STR-Connection is active to a IP with SS7 signaling, the remote switch may receive a FAR message containing an RO parameter with a Return Result component. The remote switch passes the FAR message to the intermediate switch and then to the local switch without modification.

Note: Bellcore's *GR-1129-CORE* specifies that the SS7 Facility (FAC) message should be used to transport intermediate information to the local switch. Since the FAC message is not currently supported by the UCS DMS-250 switch, the proprietary FAR message is used to provide similar functionality.

When the intermediate switch receives a FAR message containing a RO parameter, the message is passed along to the local switch without modification.

When the local switch receives a FAR message containing a RO parameter with a Return Result component, the following actions are performed:

- A **`Call_Info_From_Resource`** message is sent to the SCP. The ***`IPReturnBlock`*** parameter is included when it is present in the incoming Return Result component.
- Upon sending the message to the SCP, the local switch shall start the T1 timer and wait for further information from the SCP.

If there is already an outstanding **`Call_Info_To_Resource`** message for the last **`Call_Info_From_Resource`** message, the following actions are performed:

- The switch initiates call clearing toward the remote IP by sending a REL message.
- A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.

If the SCP responds with a **Application_Error** or **Failure_Report** message, the T1 timer expires, or a fatal application error is detected, and the following actions are performed:

- The switch initiates call clearing toward the remote IP by sending a REL message.
- AIN Final Treatment (AINF) is provided.
- Timer TSTRC is canceled.

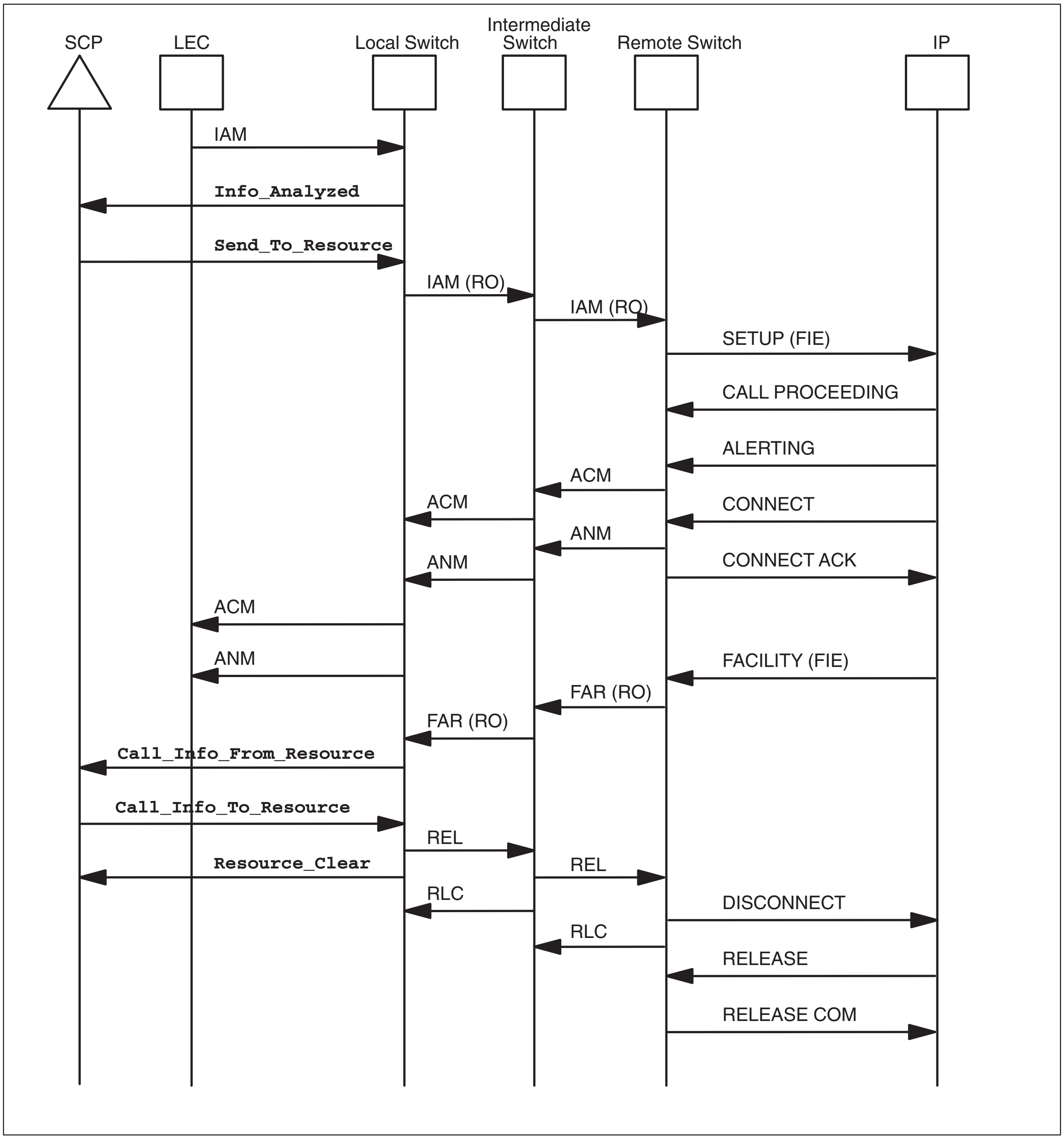
The expected response to the **Call_Info_From_Resource** message is the **Call_Info_To_Resource** message. The message is processed differently depending upon the presence or absence of the ***ResourceType*** and ***StrParameterBlock*** parameters.

When the **Call_Info_To_Resource** message is received without the ***ResourceType*** and ***StrParameterBlock*** parameters, the SCP has determined that the IP is no longer needed for the call. The following actions are performed at the local switch:

- The switch initiates call clearing toward the remote IP by sending a REL message.
- A **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `normal`.
- The call is not cleared toward the calling party.

Figure 2-44 provides a CONNECT_1129_STYLE IPI example of intermediate information processing for an SS7 originator, when the ***ResourceType*** and ***StrParameterBlock*** parameters are not present. This example shows ISDN signaling to a remote IP.

Figure 2-44
Call_Info_To_Resource without parameters, using ISDN signaling to the remote IP

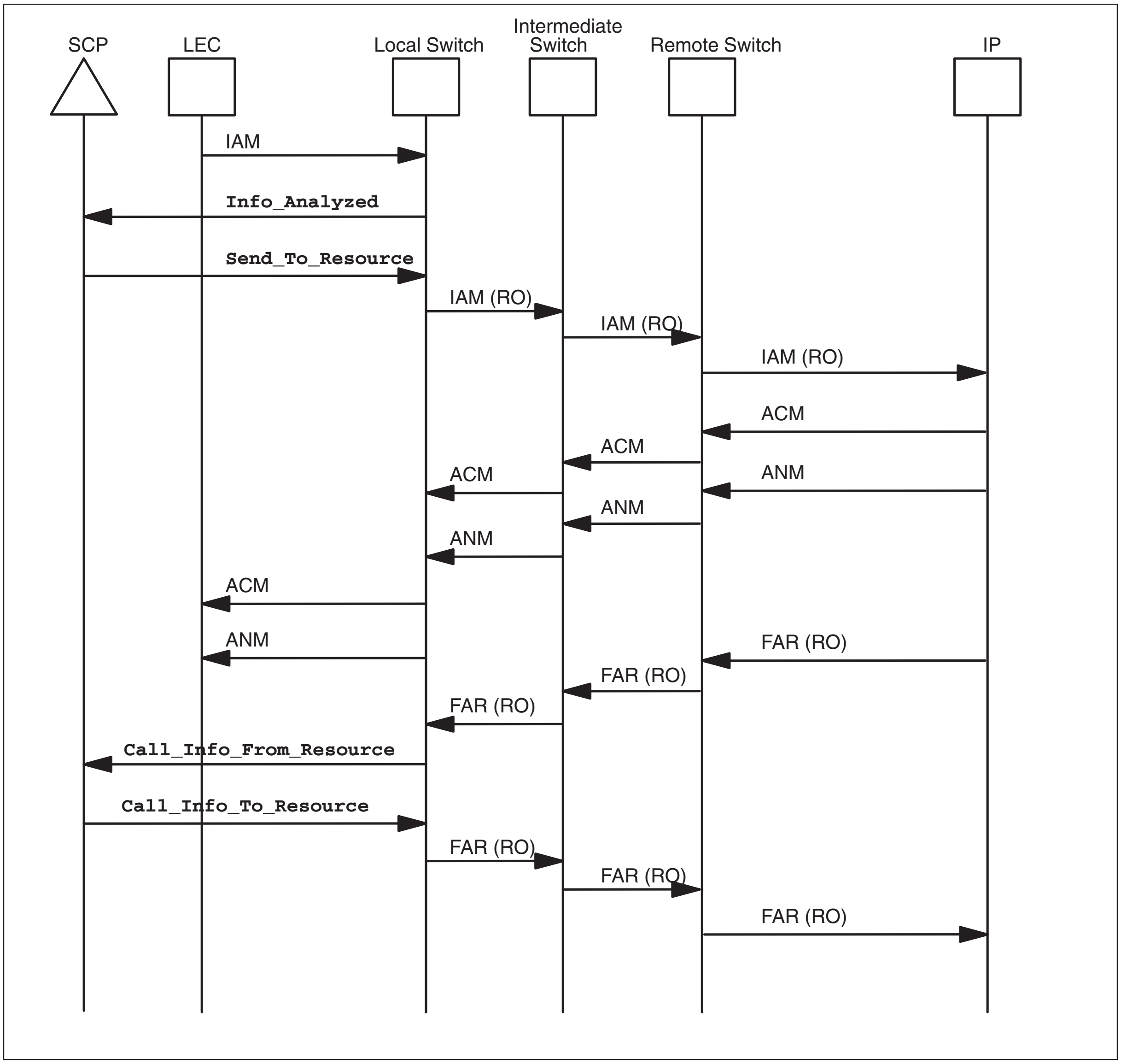


When the **`Call_Info_To_Resource`** message is received with the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters, the SCP has determined that additional information needs to be passed to the IP. The following actions are performed at the local switch:

- A FAR message with an RO parameter is sent to the intermediate and remote switches. The RO parameter contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are placed into the Invoke component without modification by the switch.
- At the intermediate switch, the FAR message containing the RO parameter is passed without modification to the remote switch.
- At the remote switch, the contents of the RO parameter are used to construct a FACILITY or FAR message which is then sent to the remote IP.

Figure 2-45 provides a CONNECT_1129_STYLE IPI example of intermediate information processing for an SS7 originator, when the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are present. This example shows SS7 signaling to a remote IP.

Figure 2-45
Call_Info_To_Resource with parameters, using SS7 signaling to the remote IP



Once the STR-Connection to the remote IP is established, the remote switch may receive a FACILITY or FAR message from the remote IP requesting a supplemental service. RLT and Billing Information are supported supplemental services.

Release link trunk (RLT)

The RLT supplemental service allows a remote IP to connect to a second subscriber, and then bridge the second subscriber to the calling subscriber. Once the call is bridged, the two subscribers are connected and the trunks between the remote IP and the remote switch are released. The RLT supplemental service also allows an IP to redirect (SS7 RLT calls only) or transfer the call.

The STR-Connection is terminated when the RLT service is invoked by the IP. This is necessary since the RLT service allows the remote IP to control the routing of the call, instead of the SCP. In order to terminate the STR-Connection, the local switch must be notified that a supplemental service has been invoked.

ATTENTION

Since the CONNECT_ONLY IPI does not support the exchange of information during an active connection, the remote switch is unable to notify the local switch that the RLT service was invoked. Therefore the STR-Connection remains active at the local switch, even though the calling user has been connected to another party. This may result in undesired interactions with the supplemental service.

When the supplemental service is invoked before an active STR-Connection (such as, before the IP answers), the following actions are performed:

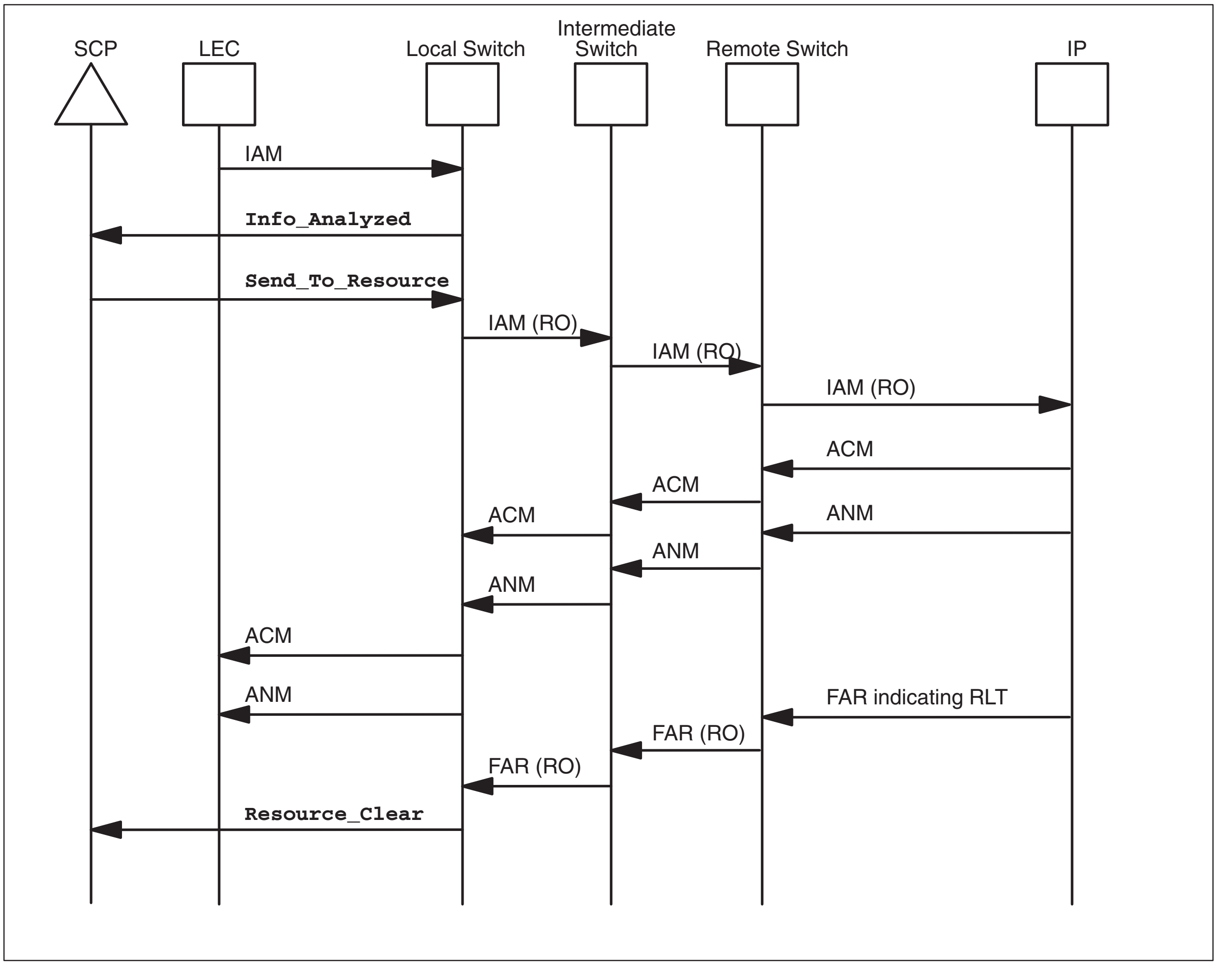
- The remote switch shall send a Call Progress (CPG) message containing a RO parameter to the intermediate and local switch. The RO parameter shall contain a Return Error component with an Error Code of `suppServiceInvoked`. The remote switch then processes the incoming FACILITY or FAR message and performs the requested supplemental service.
- When the intermediate switch receives the CPG message containing the RO parameter, it will pass the message to the local switch without modification.
- When the local switch receives the CPG message containing the RO parameter with the Return Error component, the following actions are performed:
    - — A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `suppServiceInvoked`.
    - — The STR-Connection is cleared.
    - — The call is not cleared toward the calling party.

When the supplemental service is invoked during an active STR-Connection (after the IP answers), the following actions are performed:

- The remote switch shall send a FAR message containing a RO parameter to the intermediate and local switch. The RO parameter contains a Return Error component with an Error Code of `suppServiceInvoked`. The remote switch then processes the incoming FACILITY or FAR message and performs the requested supplemental service.
- When the intermediate switch receives the FAR message containing the RO parameter, it will pass the message to the local switch without modification.
- When the local switch receives the FAR message containing the RO parameter with the Return Error component, the following actions are performed:
  - — A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `suppServiceInvoked`.
  - — The STRConnection is cleared and the TSTRC timer is canceled.
  - — The call is not cleared toward the calling party.

Figure 2-46 provides an example of RLT supplemental service invoked by a remote IP, with an SS7 originator and SS7 signaling to a remote IP.

Figure 2-46
RLT supplemental service invoked by a remote IP, using SS7 signaling



Billing information

The Billing Information supplemental service allows a remote IP to provide billing information to update the CDR fields BILLNUM, ACCTCD, and PINDIGS. This supplemental service does not affect the STR-Connection to the remote IP.

IP-provided billing information is only used to update the CDR at the remote switch. Billing at the local switch and intermediate switch(s) is unaffected.

Remote IP-initiated clearing of a STR-Connection

Once the remote switch sends the ISDN SETUP or SS7 IAM message, the remote IP may respond with a message that clears the STR-Connection. The following sections describe the messages which clear the STR-Connection.

DISCONNECT and REL messages

Since the CONNECT_ONLY IPI does not support the exchange of data, the DISCONNECT or REL message is not supported. When either message is received, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and local switches.
- At the intermediate switch, the REL message is passed to the local switch.
- At the local switch, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`. The STR-Connection is cleared.

Tables 2-8 and 2-9 provide descriptions of these ISDN and SS7 clearing messages.

Table 2-8
Remote switch processing of ISDN clearing messages received from the remote IP

Message	Description
RELEASE or RELEASE COMPLETE	During abnormal conditions, the local IP sends a RELEASE message to initiate call clearing. Examples of abnormal conditions include: • protocol violations • protocol time-outs • B-channel not available • B-channel glare • rejection of the call by the local IP When the remote switch receives a RELEASE or RELEASE COMPLETE message, an SS7 REL message (with the Cause Indicator parameter specifying the type of abnormal call clearing) is constructed and sent to the intermediate switch. The intermediate switch then passes the REL message to the local switch. When the local switch receives a REL indicating abnomal release, a **`Resource_Clear`** message is sent in a conversation package with a ***`ClearCause`*** of `abort`. The REL message is not passed to the originating switch. For SS7 originations at the local switch, a RELEASE message is not passed to the originating switch. For PRI originations at the local switch, a RELEASE (or RELEASE COMPLETE) message is not passed to the originating switch.
DISCONNECT	This message initiates normal call clearing. When the remote switch receives a DISCONNECT, an SS7 REL message (with the Cause Indicator specifying normal call clearing) is constructed and sent to the intermediate switch. The intermediate switch then passes the REL message to the local switch. When the local switch receives a normal REL, a **`Resource_Clear`** is sent in a conversation package with a ***`ClearCause`*** of `normal`. The REL message is not passed to the originating switch. For SS7 FGD originations at the local switch, a REL message is not passed to the originating switch. For PRI originations at the local switch, a DISCONNECT message is not passed to the originating switch.

Table 2-9
Remote switch processing of SS7 clearing messages received from the remote IP

Message	Description
REL	During abnormal conditions, the local IP sends a REL message to initiate call clearing. When the local switch receives a REL indicating abnomal release, a **`Resource_Clear`** message is sent in a conversation package with a ***`ClearCause`*** of `abort`. The REL message is not passed to the originating switch. For SS7 originations at the local switch, a REL message is not passed to the originating switch. For PRI originations at the local switch, a REL message is not passed to the originating switch.
RLC	During abnormal conditions, the local switch sends a RLC message to clear the STR-Connection.

When initiating normal call clearing, the remote IP sends a DISCONNECT or REL message with a Cause Indicator indicating normal clearing. When the DISCONNECT or REL message contains a Return Result component, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains the information that was present in the Return Result component received from the remote IP. The TSTRC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.

Figure 2-47 provides an example of normal CONNECT_ONLY call clearing initiated by the remote IP.

Figure 2-47
IP initiated call clearing using ISDN signaling

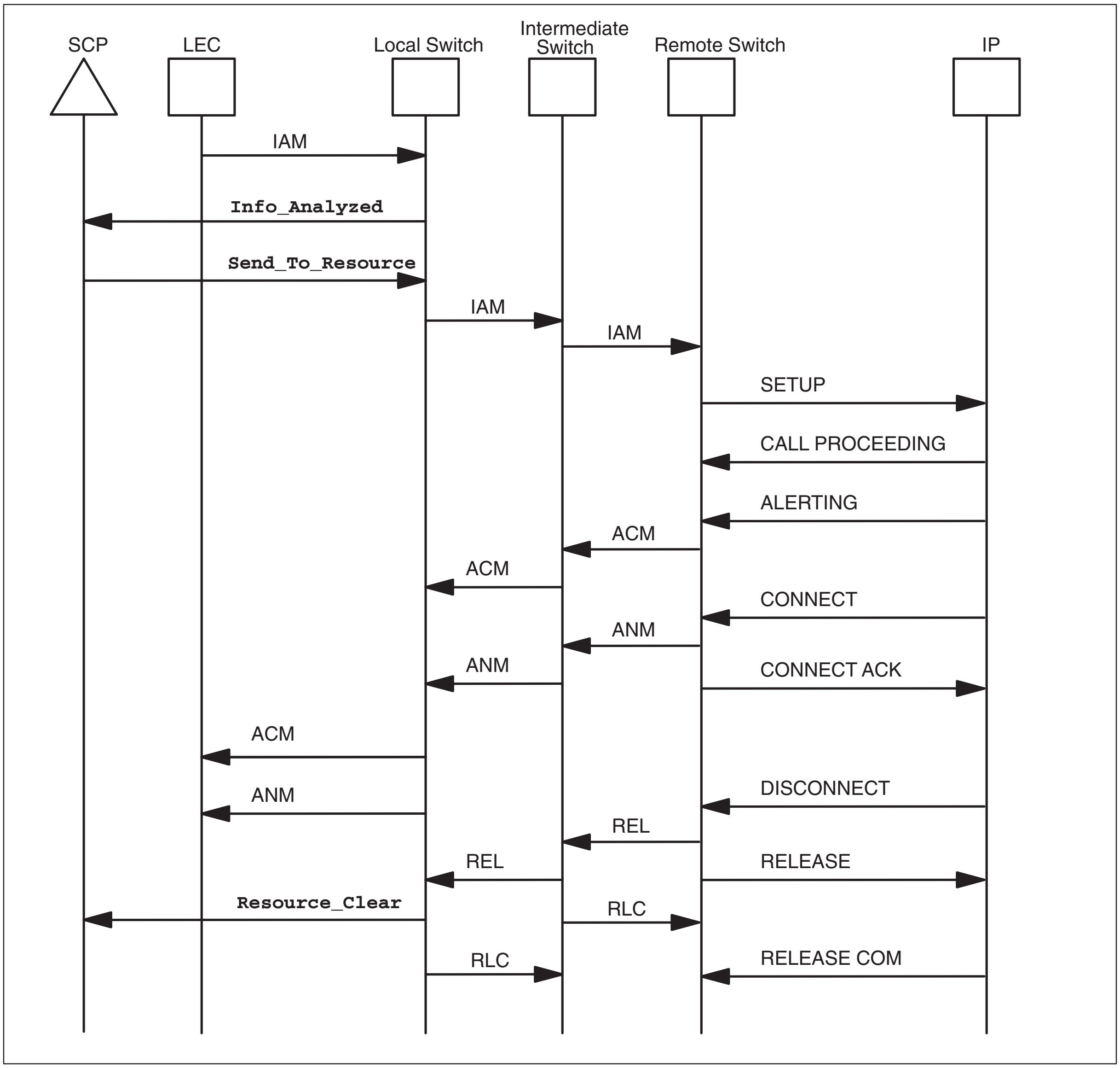
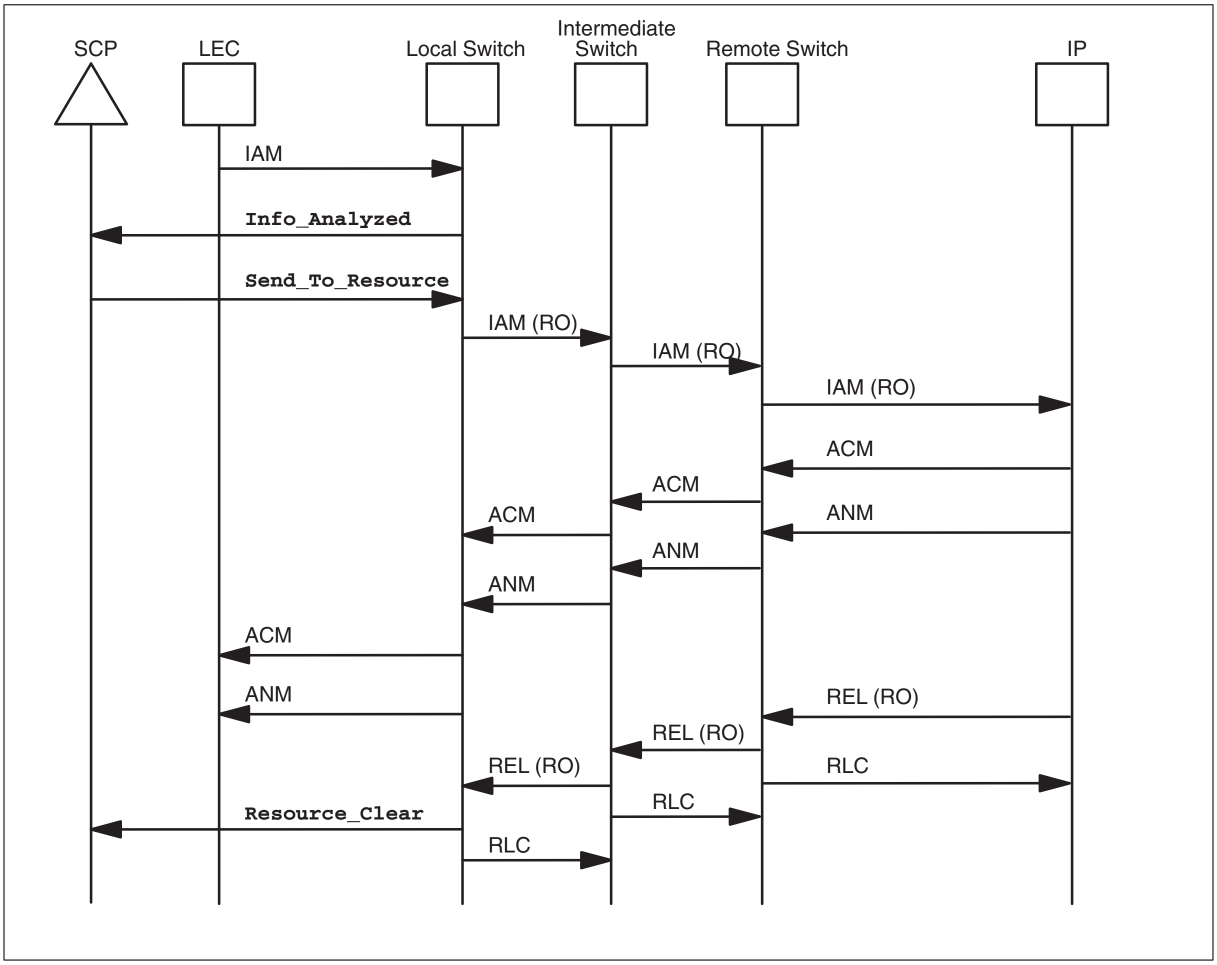


Figure 2-48 provides an example of normal CONNECT_1129_STYLE call clearing initiated by the remote IP with a Return Result component.

Figure 2-48
IP initiated call clearing with a Return Result component



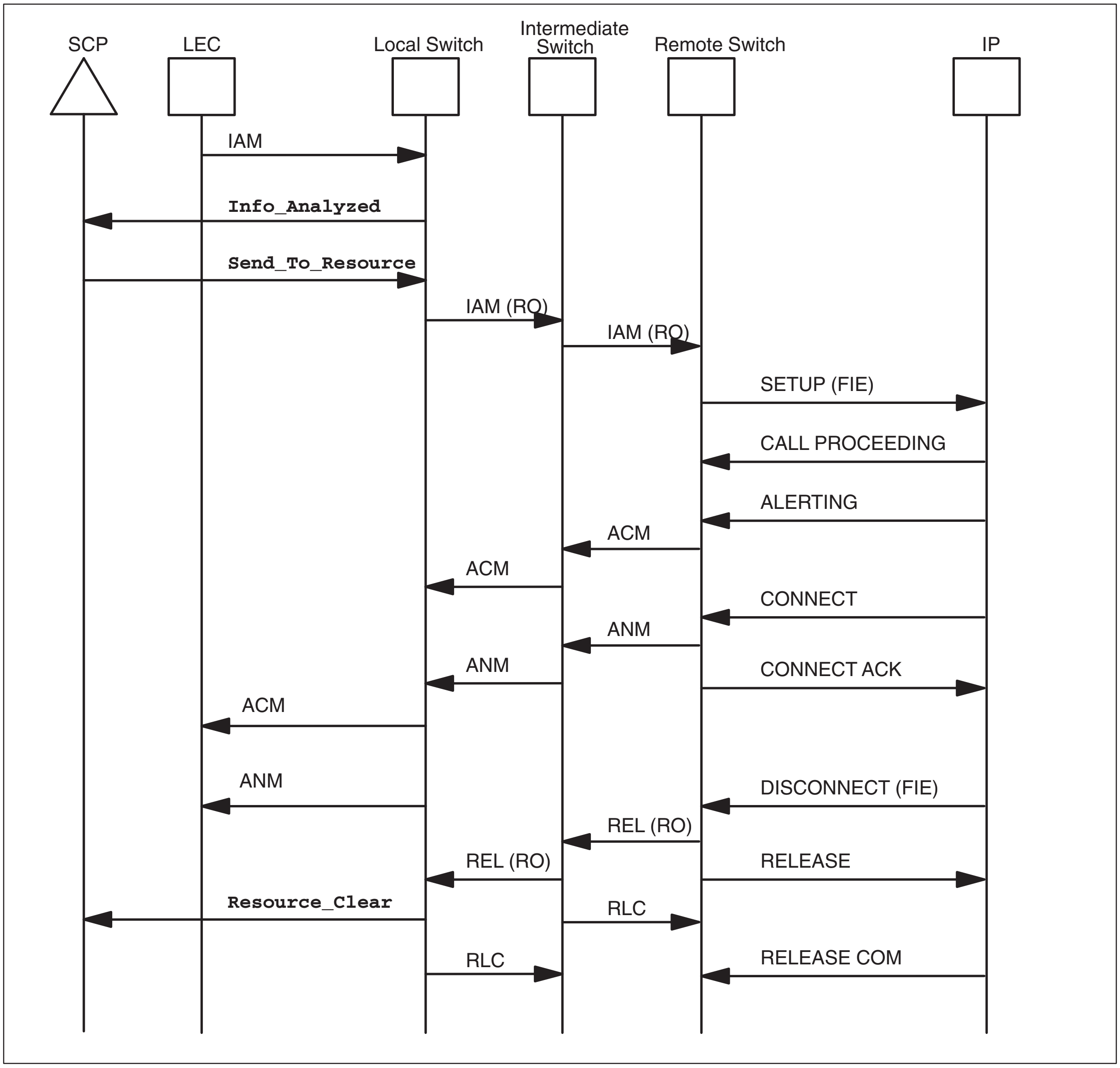
When the DISCONNECT or REL message contains a Return Error component, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains the information that was present in the Return Error component received from the remote IP. The TSTRC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:

- — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value that is based upon the contents of the Error Value field of the Return Error component.
- — The STR-Connection is cleared.
- — The call is not cleared toward the calling user.

Figure 2-49 provides an example of normal CONNECT_1129_STYLE call clearing initiated by the remote IP with a Return Error component.

Figure 2-49
IP initiated call clearing with a Return Error component



When the DISCONNECT or REL message contains a Reject component, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The TSTRC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.

For the CONNECT_1129_STYLE IPI, when the remote switch receives a DISCONNECT or REL message without a component, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and the local switch. The TSTRC timer is canceled.
- At the intermediate switch, the REL message is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.

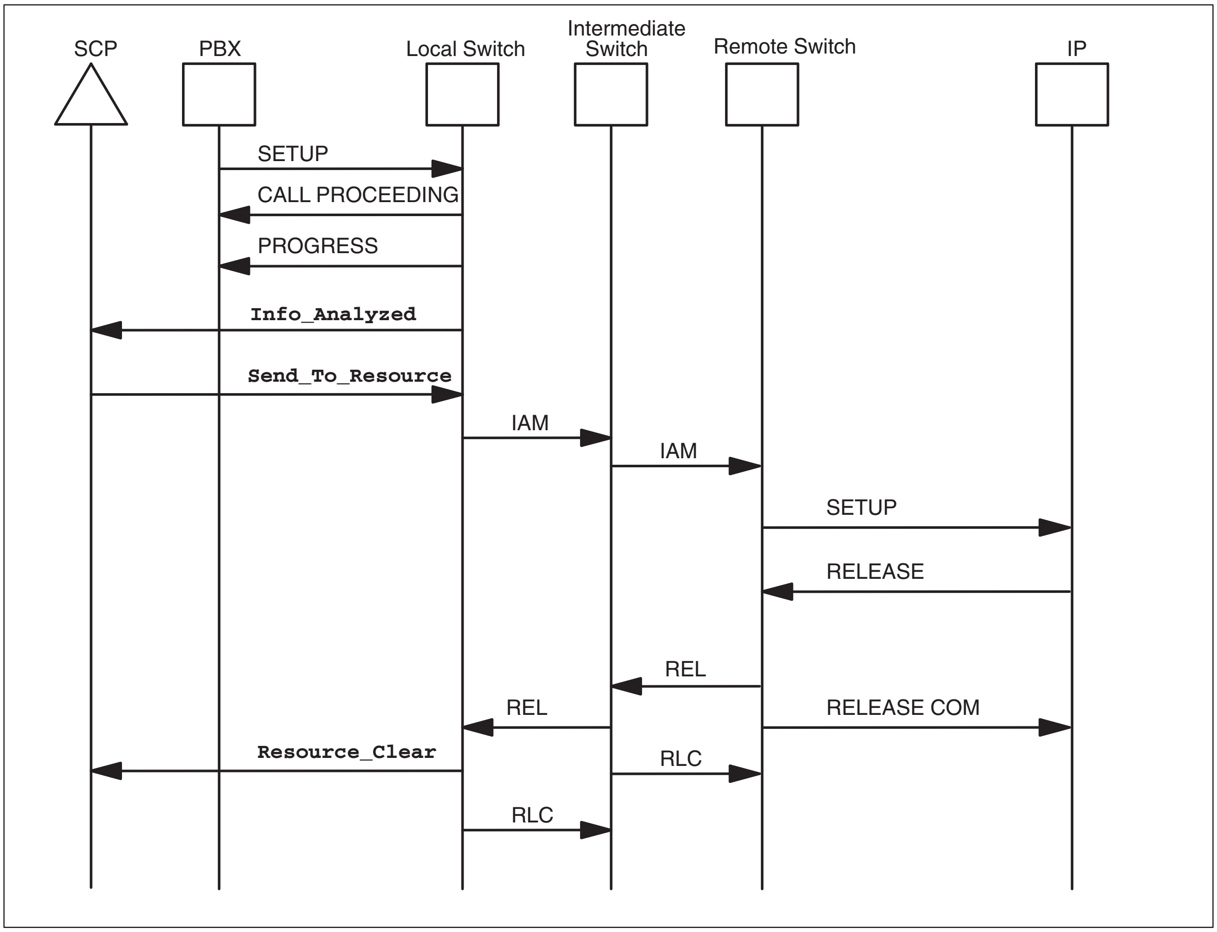
For the CONNECT_ONLY IPI, the remote switch expects to receive a DISCONNECT or REL message without a component. A **`Resource_Clear`** message in a conversation package is sent by the local switch to the SCP with a ***`ClearCause`*** parameter value of `normal`.

RELEASE COMPLETE and Release (REL) messages

For the CONNECT_ONLY IPI, once the remote switch sends the SETUP message, the remote IP may respond with a message that clears the STR-Connection. When initiating abnormal call clearing, the remote IP may send an ISDN RELEASE message or SS7 REL message with a Cause Indicator indicating abnormal clearing. A **`Resource_Clear`** message in a conversation package is sent by the local switch to the SCP with a ***`ClearCause`*** parameter value of `normal`.

Figure 2-50 provides an example of abnormal call clearing initiated by the remote IP using the CONNECT_ONLY IPI and ISDN signaling to the remote IP.

Figure 2-50
Abnormal call clearing initiated by the remote IP



For the CONNECT_1129_STYLE IPI, when initiating abnormal call clearing, the remote IP sends an ISDN RELEASE COMPLETE message or SS7 REL message with a Cause Indicator indicating abnormal clearing. When the message contains a Reject component, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The TSTRC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **Resource_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of protocolError.
  - — The STR-Connection is cleared.

— The call is not cleared toward the calling user.

If the RELEASE COMPLETE or REL message does not contain a component, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and the local switch. The TSTRC timer is canceled.
- At the intermediate switch, the REL message is passed without modification to the local switch.
- At the local switch, the following actions are performed:

  — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.

  — The STR-Connection is cleared.

  — The call is not cleared toward the calling user.

FACILITY and Facility Request (FAR) messages

For the CONNECT_ONLY IPI, the exchange of data using the ISDN FACILITY or SS7 FAR message is not supported.

For the CONNECT_1129_STYLE IPI, when initiating abnormal call clearing during an active STR-Connection, the remote IP may send an FACILITY or FAR message. When the FACILITY or FAR message contains a Return Error component, the following actions are performed:

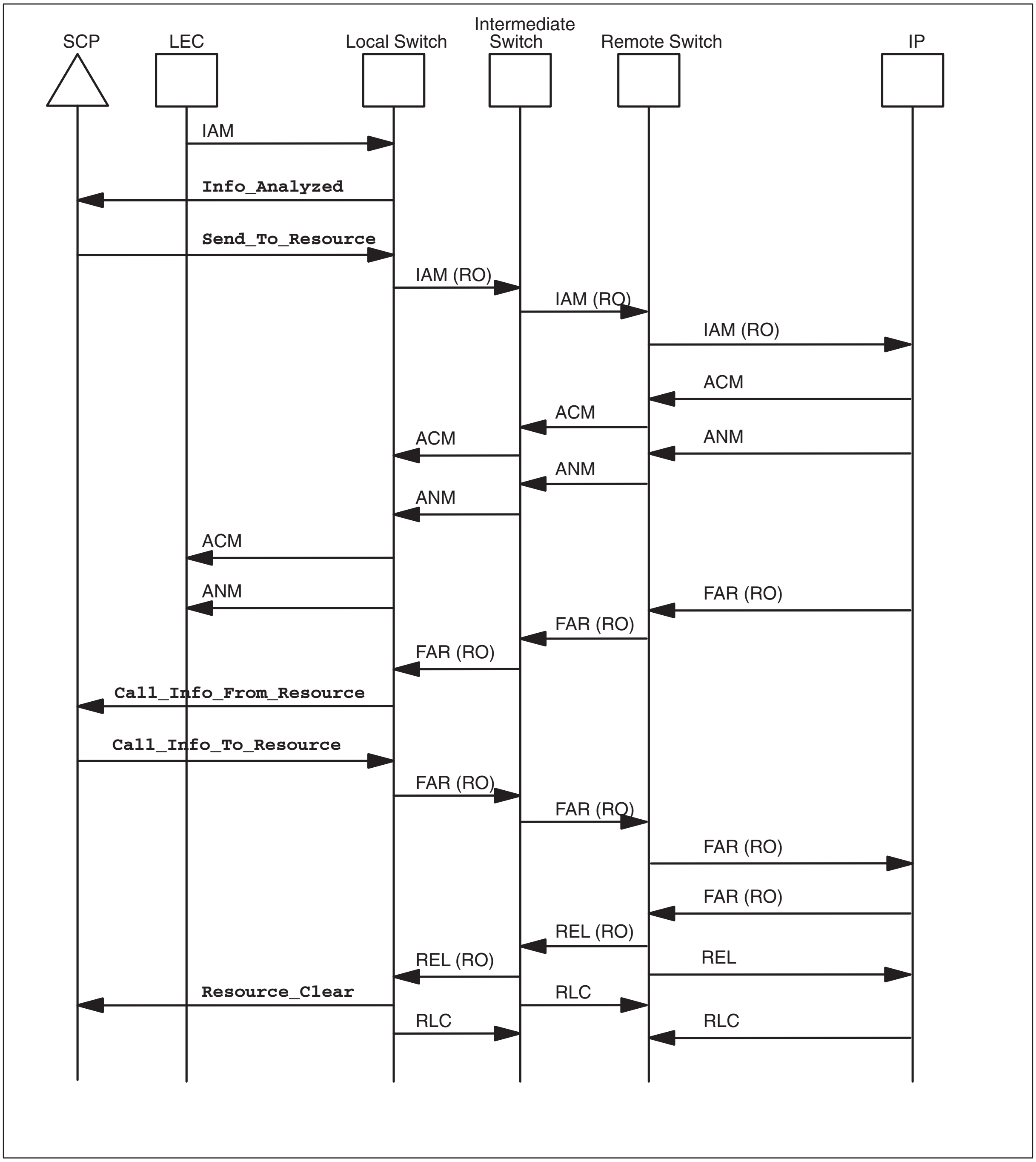
- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains the information that was present in the Return Error component received from the remote IP. The TSTRC timer is canceled and the STR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:

  — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value that is based upon the contents of the Error Value field of the Return Error component.

  — The call is not cleared toward the calling user.

Figure 2-51 provides an example of abnormal call clearing initiated by the remote IP using the CONNECT_1129_STYLE IPI and SS7 signaling to the remote IP.

Figure 2-51
IP initiated call clearing with the FAR message



When the FACILITY or FAR message contains a Reject component, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The TSTRC timer is canceled and the STR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
    - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.
    - — The call is not cleared toward the calling user.

Remote switch-initiated clearing of a STR-Connection

The STR-Connection to the remote IP is cleared by the switch when the calling user abandons, the SCP responds with the **`Cancel_Resource_Event`** message, or when the TSTRC timer expires. These scenarios are described in the following sections.

Caller Abandon

Since the exchange of data is not supported for the CONNECT_ONLY IPI, the following actions are performed by the local switch upon user abandon during an active STR-Connection:

- A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
- The STR-Connection is cleared.

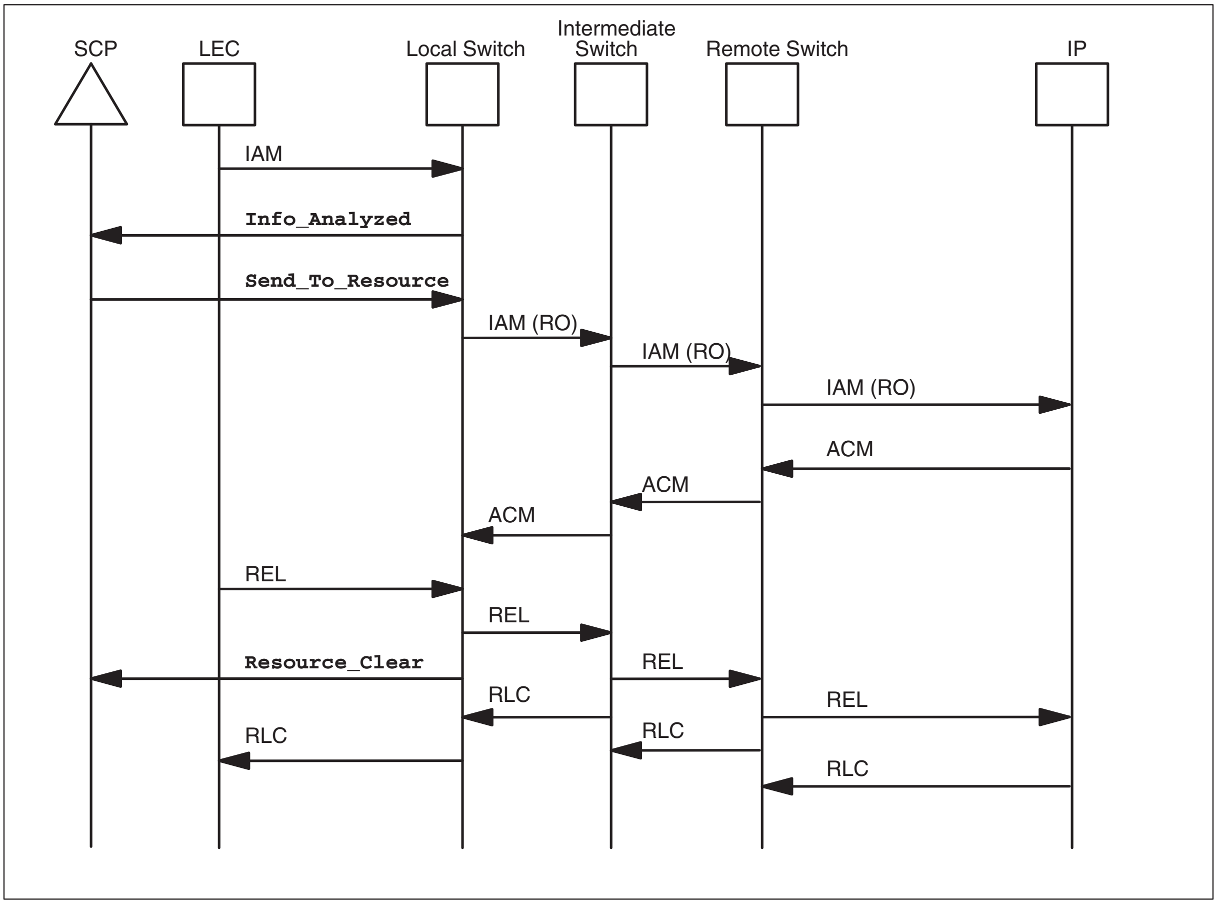
With the CONNECT_1129_STYLE IPI, the calling user may decide to abandon the call during a STR-Connection to the remote IP. When this occurs before the STR-Connection is active (before the IP has answered), the following actions are performed by the local switch:

- A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
- The STR-Connection is cleared.

Note: This behavior differs from Bellcore's *GR-1129-CORE* which states that a FAC message should be sent to the remote switch, which then initiates call clearing if the IP hasn't answered. NetworkBuilder will allow the local switch to initiate call clearing immediately.

Figure 2-52 provides an example of a caller abandon scenario before an active STR-Connection is established.

Figure 2-52
Caller abandon before an active STR-Connection



When the calling user abandons during an active STR-Connection, the following actions are performed:

- At the local switch, a FAR message containing an Invoke component with an operation of `cancelIPResource` is sent to the remote switch.
- At the intermediate switch, the SS7 FAR message is passed without modification to the remote switch.
- At the remote switch, the following actions are performed:
  - — An ISDN FACILITY or SS7 FAR message is sent to the remote IP containing the component information received in the incoming FAR message.
  - — The TSTRC timer is canceled.

- — The TDISC timer is started. This timer specifies the maximum time in seconds in which an IP must respond to a FACILITY or FAR message with the `cancelIPResource` operation.

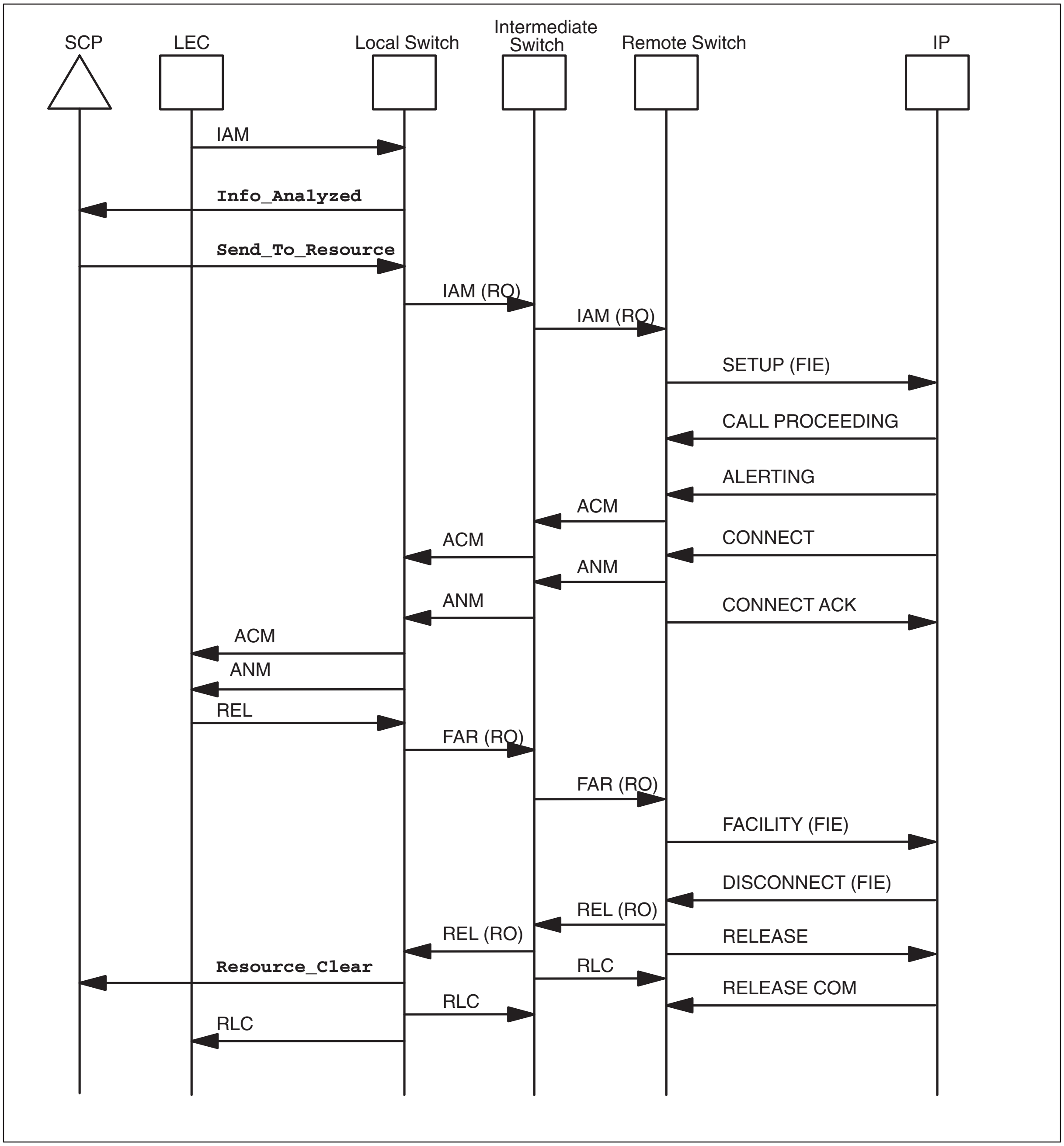
The remote IP is expected to respond to the `cancelIPResource` operation with an ISDN DISCONNECT or SS7 REL message normally containing a Return Result component. The following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains the information that was present in the Return Result component received from the remote IP. The TDISC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — When the T1 timer is not running:
    - – A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component.
    - – The STR-Connection is cleared.
  - — When the T1 timer is running:
    - – The switch shall await the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - – The STR-Connection is cleared.

    Note: In this scenario, the IP has already sent a FACILITY or FAR message containing a Return Result component to the switch. When the user abandons, a FACILITY or FAR message with the Class 5 operation `cancelIPResource` is sent to the IP. However, the IP has already responded to the `sendToIPResource` operation, and it is not allowed to respond to a Class 5 operation. Therefore, the IP sends a DISCONNECT or REL message that does not contain a component.
  - — When the T1 timer expires before the SCP response is received, a fatal application error is detected.

Figure 2-53 provides an example of caller abandon dudring an active STR-connection when the T1 timer is not running.

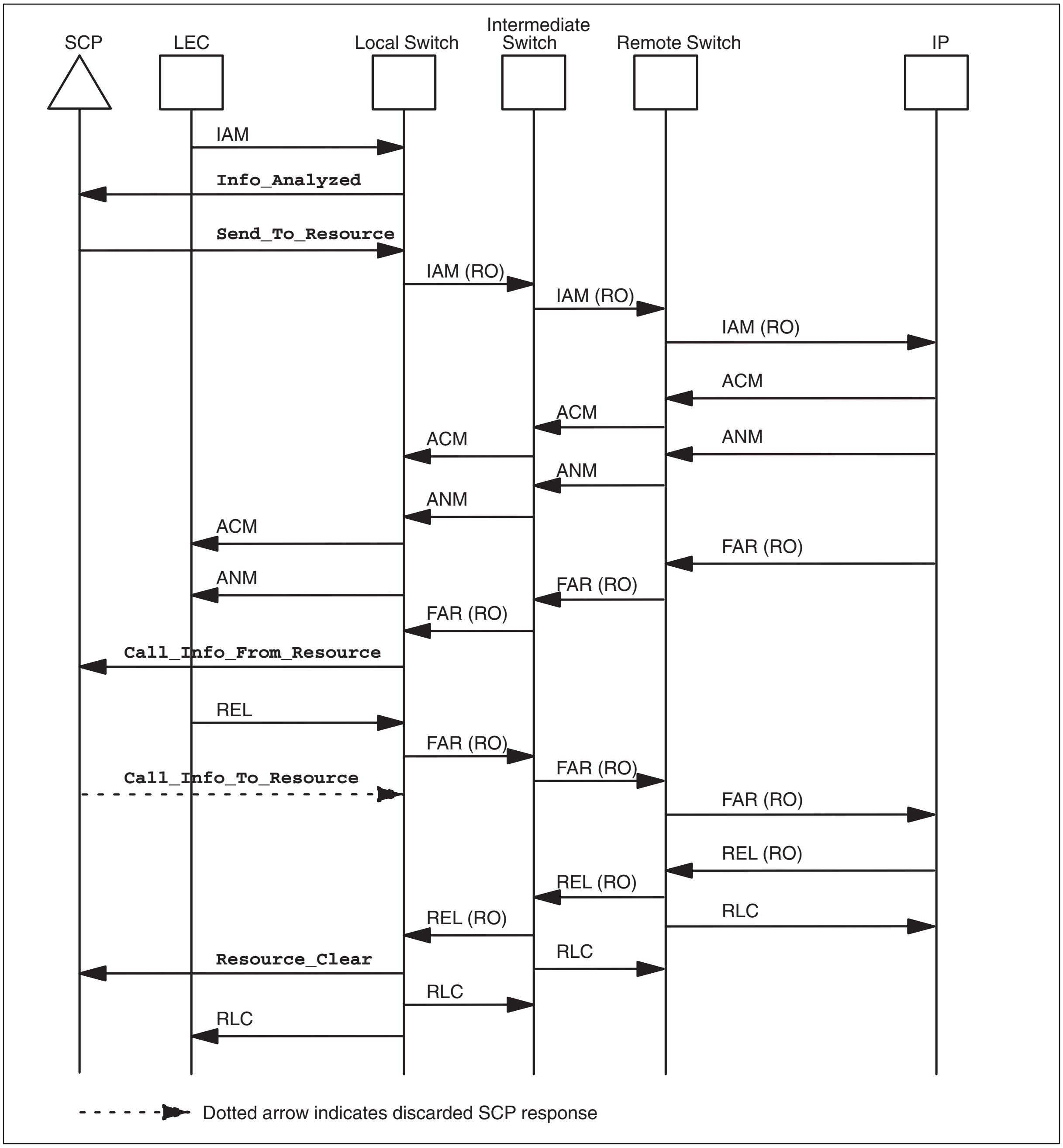
Figure 2-53
Caller abandon during an active STR-Connection, T1 timer not running



It is possible that the caller abandoned while the SCP was waiting for a **`Call_Info_To_Resource`** message. If the message is received after the FAR message with the `cancelIPResource` operation was sent to the IP, the T1 timer is canceled and the **`Call_Info_To_Resource`** message is discarded.

Figure 2-54 provides an example of a caller abandon while the SCP waits for a `Call_Info_To_Resource` message.

Figure 2-54
Call_Info_To_Resource message discarded during caller abandon



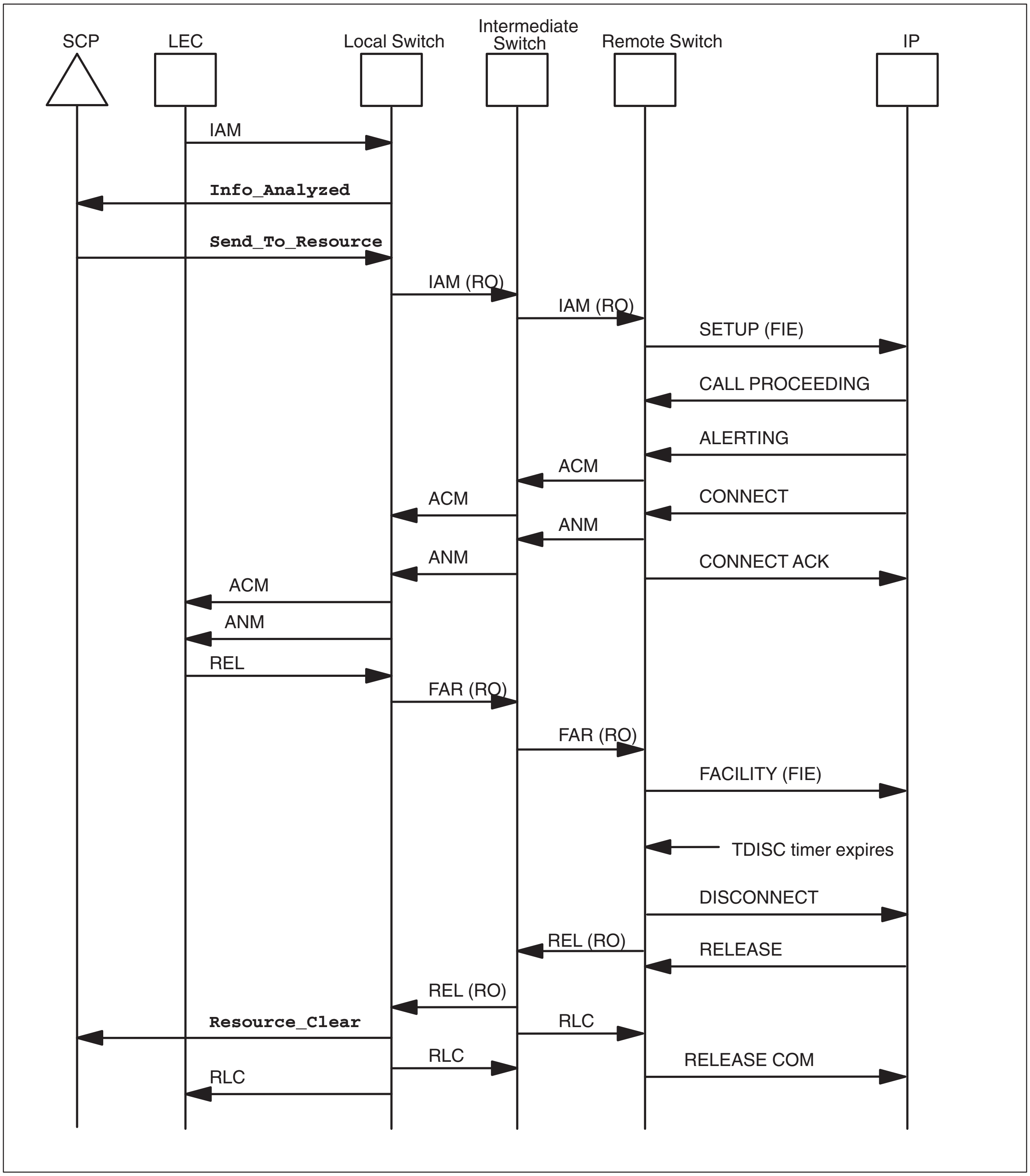
If the TDISC timer expires before the remote IP responds to the FACILITY or FAR message with the `cancelIPResource` operation, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains a Return Error component with the Error Code set to `ipTimeout`. The STR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - When the T1 timer is not running:
    - A **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - The STR-Connection is cleared.
  - When the T1 timer is running:
    - The switch shall await the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`Resource_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - The STR-Connection is cleared.
  - When the T1 timer expires before the SCP response is received, a fatal application error is detected.

    Note: The ***`ClearCause`*** value of `ipTimeout` is not sent to the SCP in this scenario.

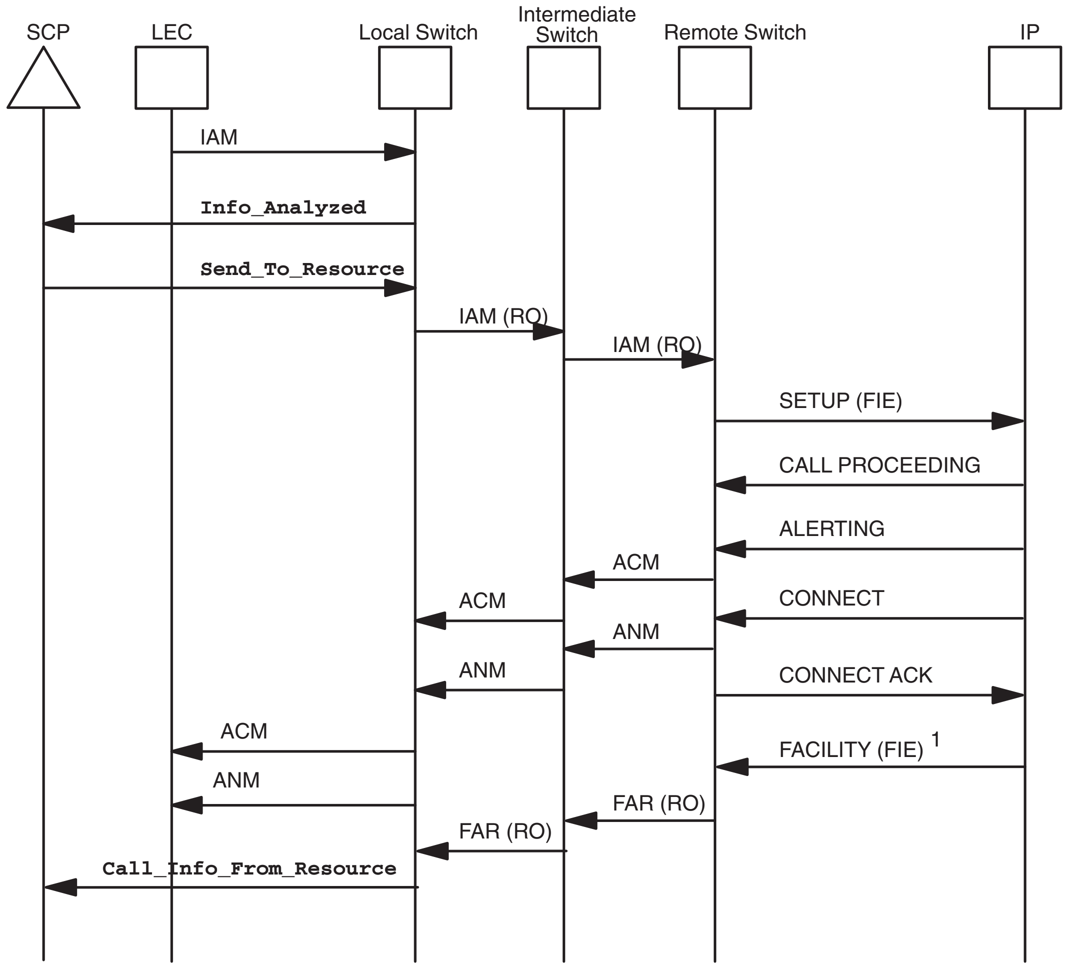
Figure 2-55 provides an example of the TDISC timer expiring before the remote IP responds to the FACILITY message.

Figure 2-55
TDISC timer expires during caller abandon, T1 timer not running



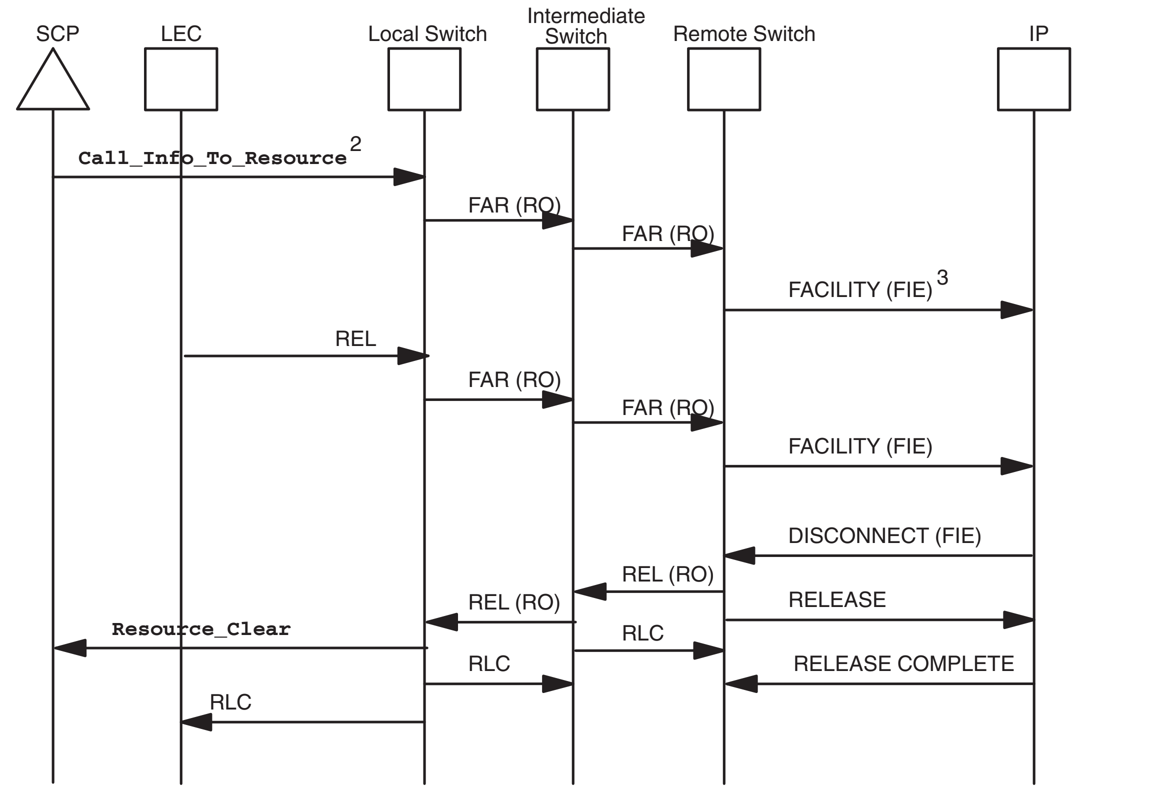
Figures 2-56 and 2-57 provide examples of the early billing during caller abandons for the case of a remote IP.

Figure 2-56
Early billing during caller abandon (continued in Figure 2–55)



1. Subscriber has selected the menu option (for example, “1’ for stock quotes). This FACILITY message contains a Return Result component with the menu choice so the SCP can determine how to bill the call.

Figure 2-57
Early billing during caller abandon (end)



2. This Call_Info_To_Resource message includes billing information (amaDigits extension parameter) to update switch billing.

3. This FACILITY message contains the FlexParameterBlock to instruct the IP to deliver the service (stock quotes).

Cancel_Resource_Event

Since the exchange of data is not supported with the CONNECT_ONLY IPI, the following actions are performed by the local switch when the **`Cancel_Resource_Event`** message is received during an active connection:

- A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `resourceCanceled`.
- The STR-Connection is cleared.
- The call is not cleared toward the calling user.

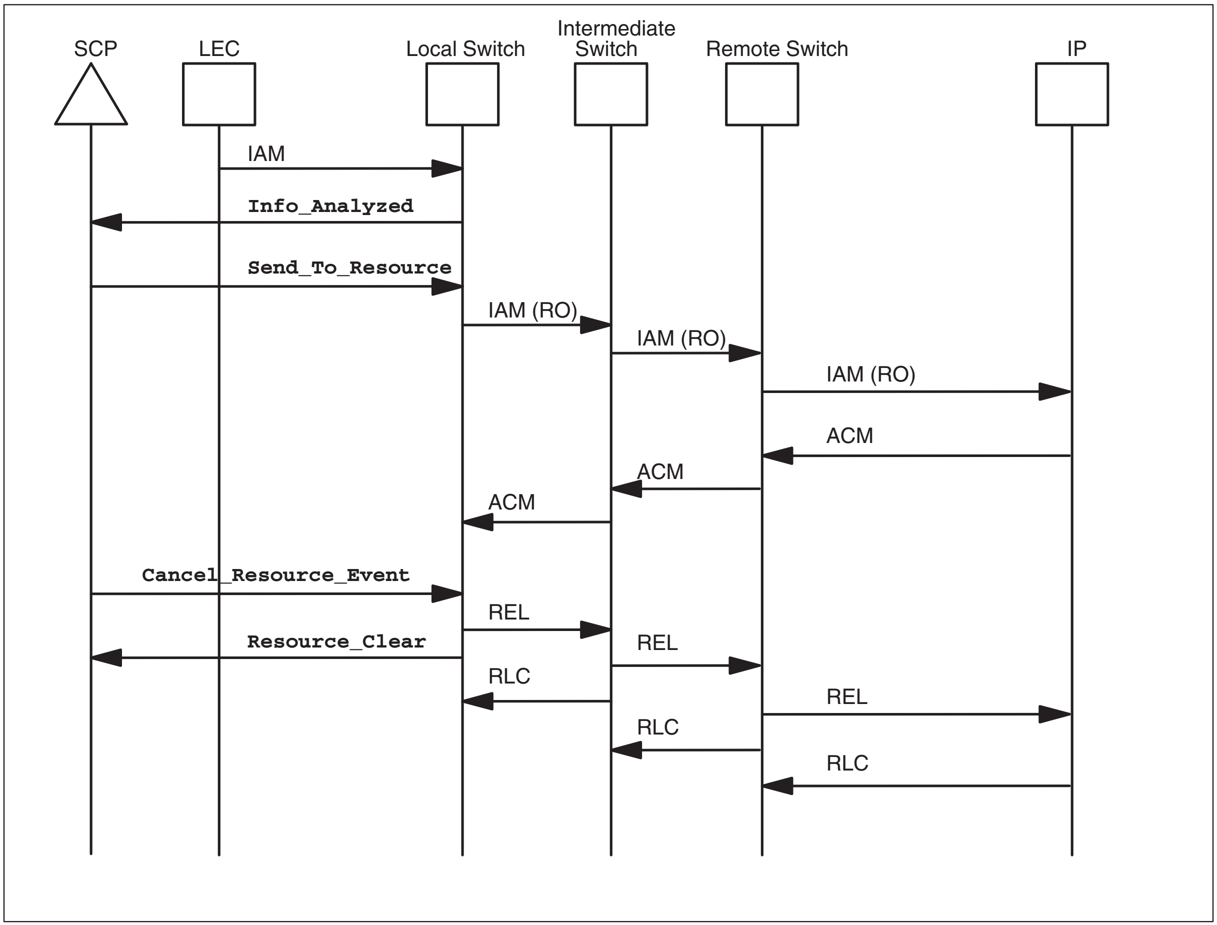
During a STR-Connection with the CONNECT_1129_STYLE IPI, the SCP may send a **`Cancel_Resource_Event`** message to the remote switch to request termination of the connection. When this occurs before the STR-Connection is active (before the IP has answered), the following actions are performed by the local switch:

- A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `resourceCanceled`.
- The STR-Connection is cleared.
- The call is not cleared toward the calling user.

  Note: This behavior differs from Bellcore's *GR-1129-CORE* which states that a FAC message should be sent to the remote switch, which then initiates call clearing if the IP hasn't answered. NetworkBuilder will allow the local switch to initiate call clearing immediately.

Figure 2-58 provides an example of the SCP requesting termination of the STR-Connection before answer indication is received from the IP.

Figure 2-58
Cancel_Resource_Event message received before IP answer



When the `Cancel_Resource_Event` message is received during an active STR-Connection, the following actions are performed:

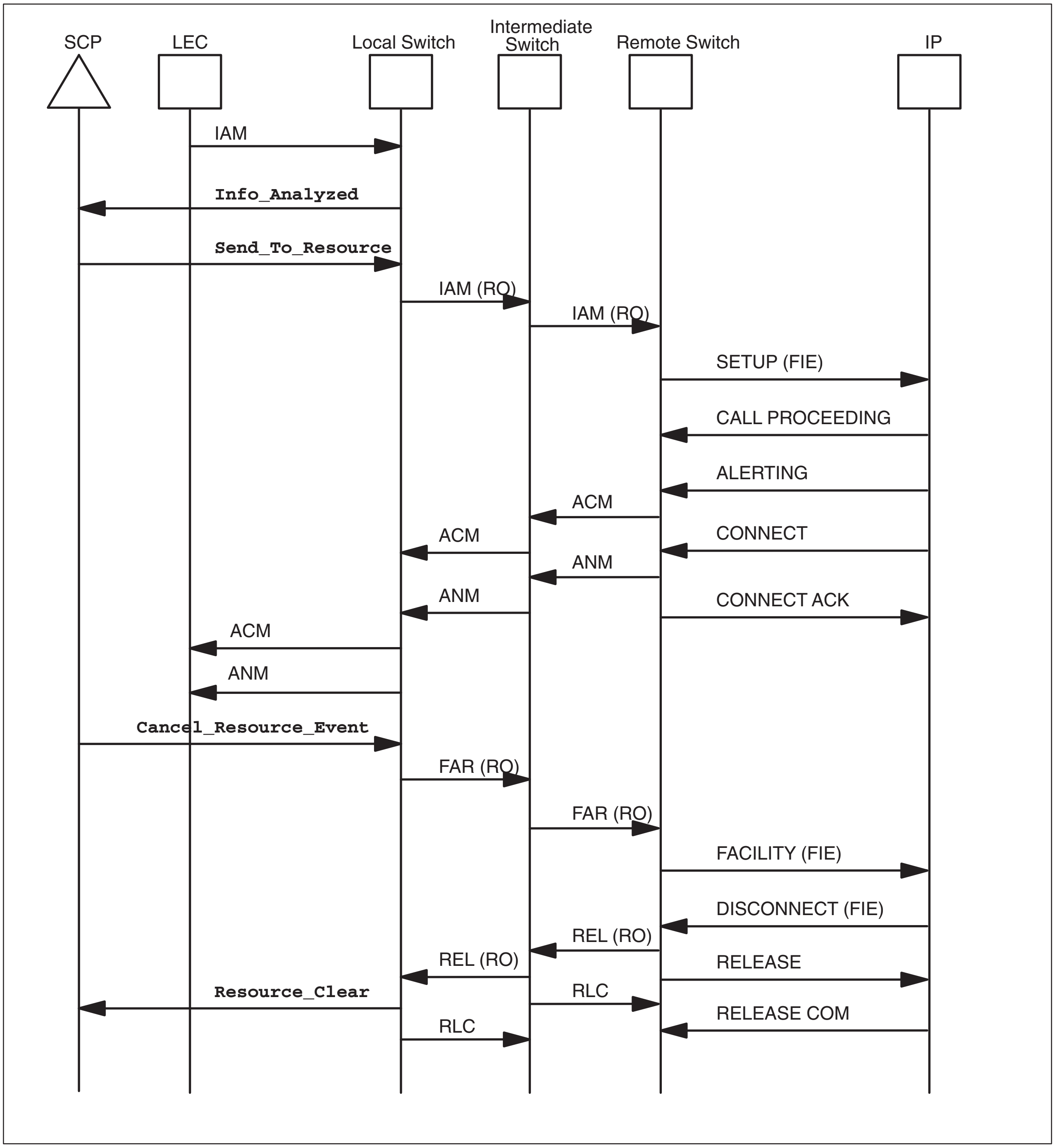
- At the local switch, a FAR message containing an Invoke component with an operation of `cancelIPResource` is sent to the remote switch.
- At the intermediate switch, the FAR message is passed without modification to the remote switch.
- At the remote switch, the following actions are performed:
  - — A FACILITY or FAR message is sent to the remote IP containing the component information received in the incoming FAR message.
  - — The TSTRC timer is canceled.
  - — The TDISC timer is started. This timer specifies the maximum time in seconds in which an IP must respond to a FACILITY or FAR message with the `cancelIPResource` operation.

The remote IP is expected to respond to the `cancelIPResource` operation with a DISCONNECT or REL message containing a Return Result component. The following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains the information that was present in the Return Result component received from the remote IP. The TDISC timer is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `resourceCanceled`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component.
  - — The STR-Connection is cleared.
- The call is not cleared toward the calling user.

Figure 2-59 provides an example of the SCP requesting termination of the STR-Connection after the IP answers.

Figure 2-59
Cancel_Resource_Event message received after IP answers



If the TDISC timer expires before the remote IP responds to the FACILITY or FAR message with the `cancelIPResource` operation, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains a Return Error component with the Error Code set to `ipTimeout`. The STR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.

The **`Cancel_Resource_Event`** message is ignored when it is received as a response to a **`Resource_Clear`** message. If the **`Cancel_Resource_Event`** message is received at any other time when it is not expected, a fatal application error is detected. The following actions are performed:

- A CAIN200 Fatal Application Error log is generated with the REASON field set to “UNEXPECTED MESSAGE”, and final treatment is provided.
- An **`Application_Error`** message is reported to the SCP with the ***`ErrorCause`*** parameter is set to `unexpectedMessage`.

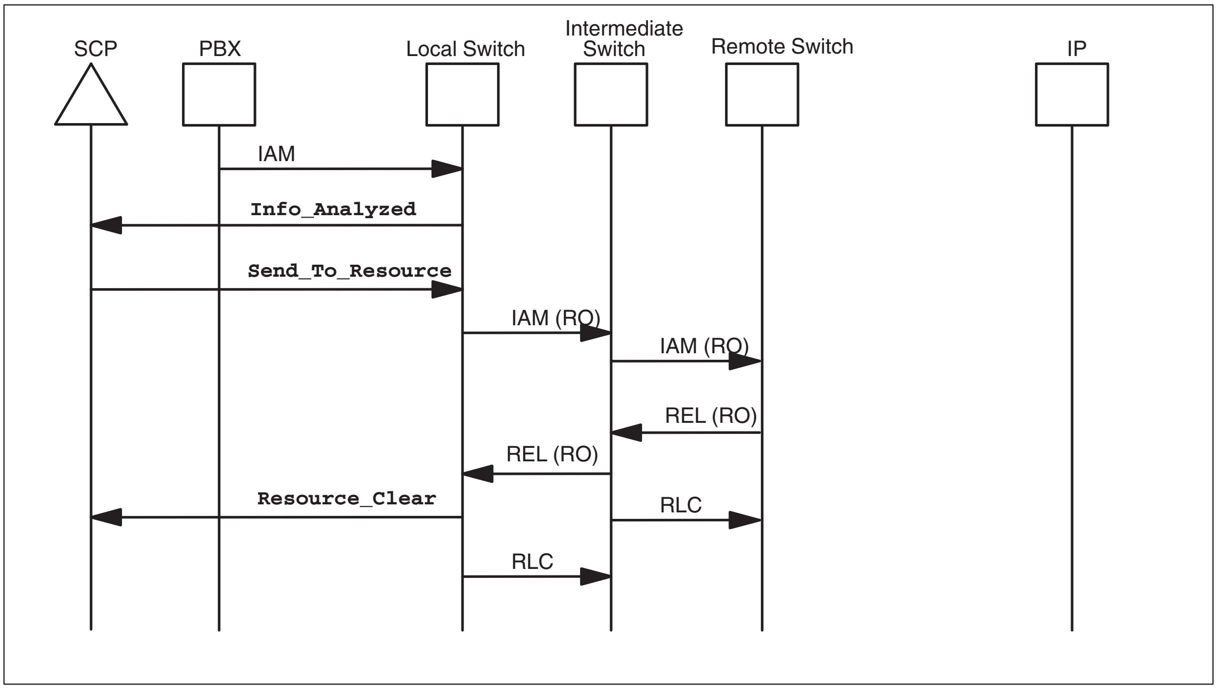
All Channels Busy

As explained earlier, the remote switch attempts to establish a connection to the remote IP when it receives an IAM containing an RO parameter. If the remote switch is unable to locate an idle trunk member terminating to the remote IP, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains a Return Error component with the Error Code set to `channelsBusy`.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `channelsBusy`.
  - — The call is not cleared toward the calling user.

Figure 2-60 provides a CONNECT_1129_STYLE IPI example of abnormal call clearing initiated by the remote switch when the remote switch is unable to locate an idle trunk memeber terminating to the remote IP.

Figure 2-60
All channels busy at the remote switch (call established with SS7 signaling)

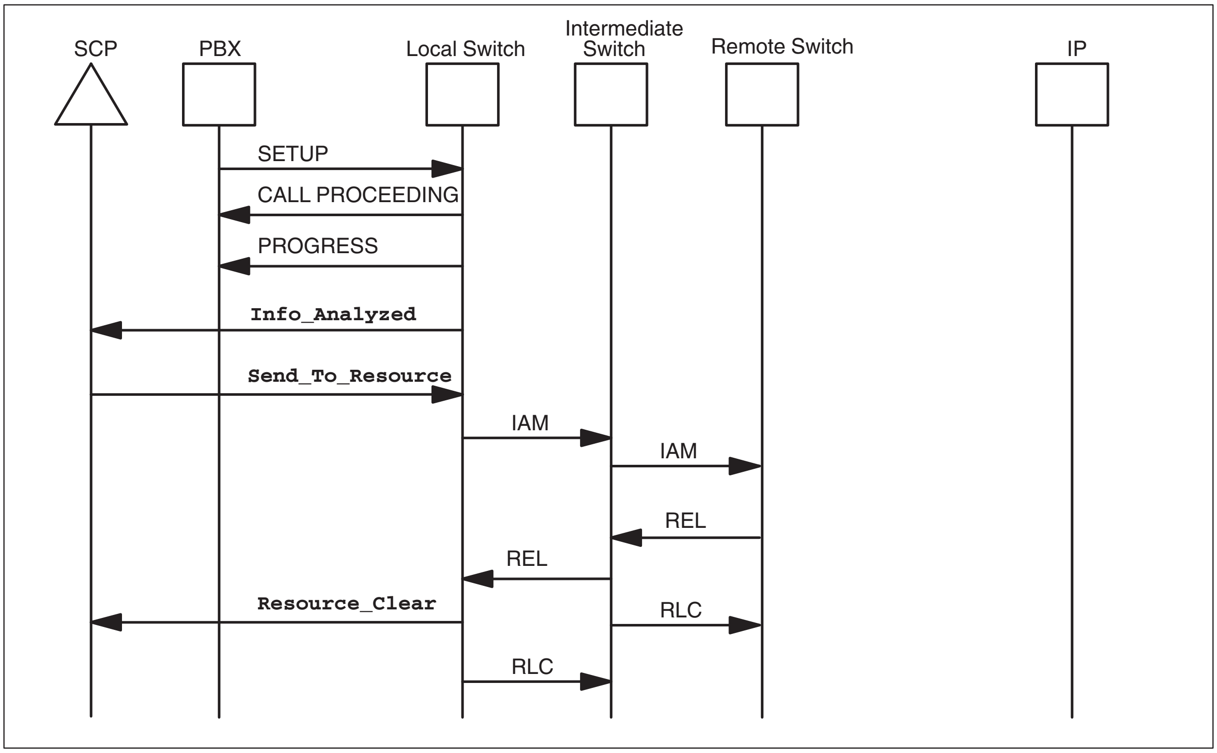


For the CONNECT_ONLY IPI, when the remote switch is unable locate an idle trunk member terminating to the remote IP, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and the local switch.
- At the intermediate switch, the REL message is passed without modification to the local switch.
- At the local switch, a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`. The call is not cleared toward the calling user.

Figure 2-61 provides a CONNECT_ONLY IPI example of abnormal call clearing initiated by the remote switch when the remote switch is unable to locate an idle trunk member terminating to the remote IP.

Figure 2-61
All channels busy at the remote switch (call estalished with ISDN signaling)



TSTRC timer expires

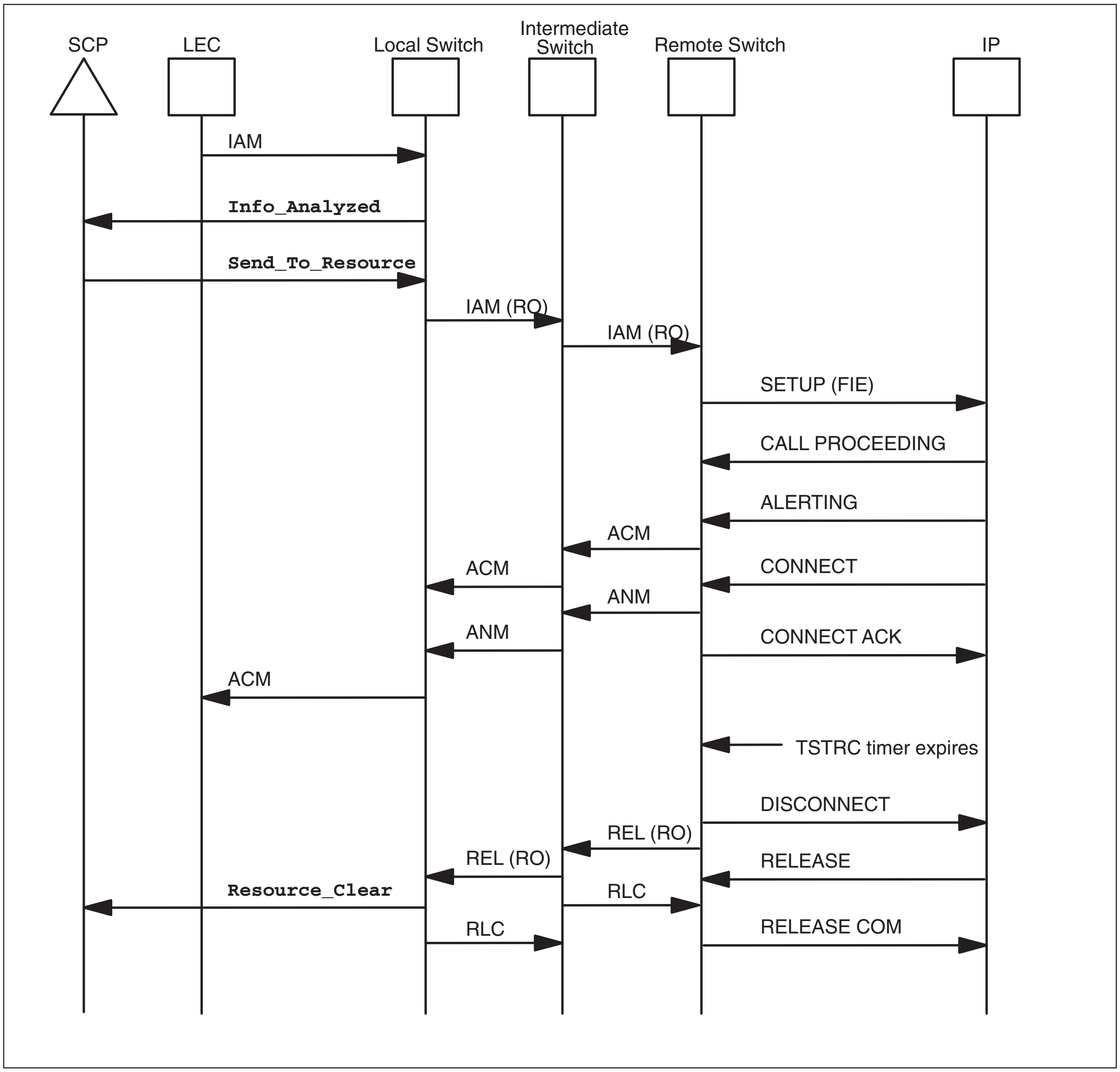
As explained earlier, the remote switch starts the TSTRC timer when the IP answers. If the timer expires during an active STR-Connection, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate and the local switch. The RO parameter contains a Return Error component with the Error Code set to `ipTimeout`. The STR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - When the T1 timer is not running, the following actions are performed:
    - A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
    - The call is not cleared toward the calling user.
  - When the T1 timer is running, the following actions are performed:

- The switch shall await the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The call is not cleared toward the calling user.
- If the T1 timer expires before the SCP response is received, a fatal application error is detected.

Figure 2-62 provides an example of the TSTRC timer expiring at the remote switch during an active STR-Connection when the T1 timer is not running.

Figure 2-62
TSTRC timer expires at the remote switch, T1 timer not running



Unexpected switch errors

The remote switch may encounter unexpected error conditions which prevent the establishment of the STR-Connection. Examples of unexpected errors include:

- The remote switch is unable to translate the Called Party Number and identify a route list.
- The remote switch attempts to terminate to an agent that is not a PRI or SS7 IMT or does not have the IPTRUNK option provisioned.

- An in-switch feature sends the call to a treatment.

When an unexpected error condition occurs, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and the local switch.The Cause Indicator parameter identifies the problem encountered by the remote switch.
- At the intermediate switch, the REL message is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.

Intermediate switch-initiated clearing of a STR-Connection

The intermediate switch may encounter unexpected error conditions which prevent the establishment of the STR-Connection. Examples of unexpected errors include:

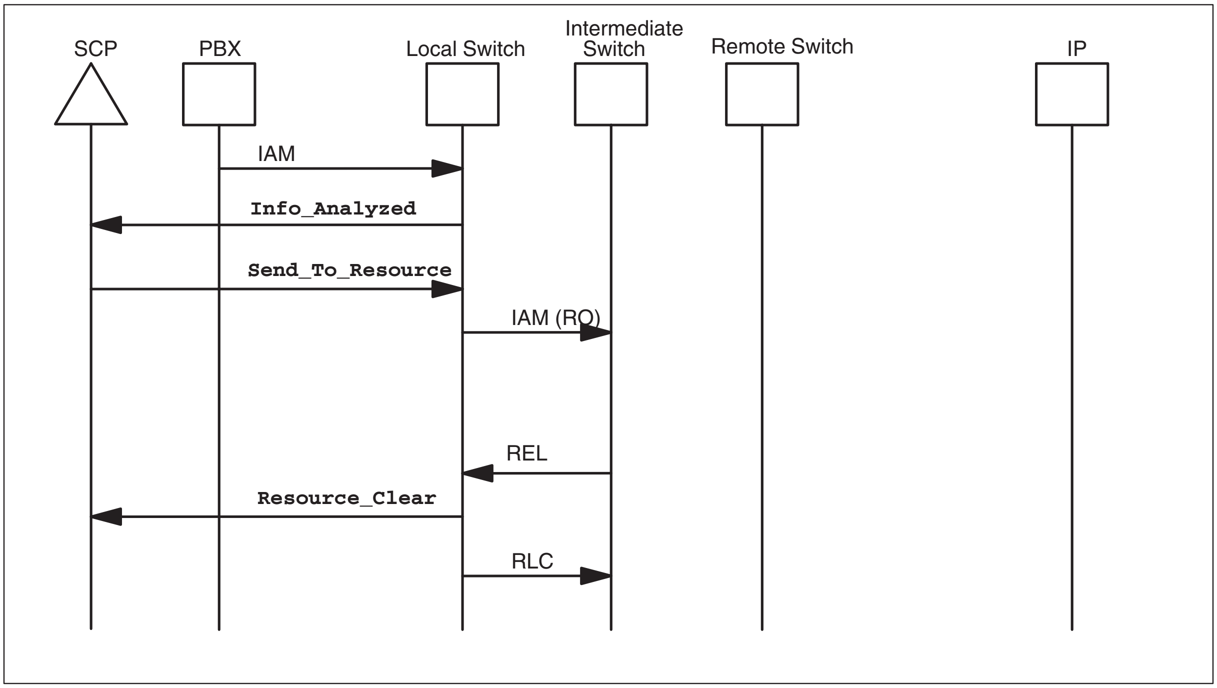
- The intermediate switch is unable to translate the Called Party Number and identify a route list.
- The intermediate switch attempts to terminate to an SS7 IMT that does not include the IPTRUNK option in table TRKGRP for the agent.
- An inswitch feature sends the call to a treatment.

When an unexpected error condition occurs, the following actions are performed:

- At the intermediate switch, a REL message is sent to the local switch. The Cause Indicator parameter identifies the problem encountered by the intermediate switch.
- At the local switch, the following actions are performed:
  - — A **`Resource_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
  - — The STR-Connection is cleared.
  - — The call is not cleared toward the calling user.
  - — Local switch-initiated clearing of a STR-Connection.

Figure 2-63 provides an example of an unexpected error encountered by intermediate switch which prevents the establishment of the STR-Connection.

Figure 2-63
Unexpected error condition at the intermediate switch



SCP response processing

When the `Resource_Clear` message is sent to the SCP in a conversation package, the local switch temporarily suspends call processing and awaits an SCP response. When the response arrives, existing CAIN software processes the message and performs the appropriate actions. Unless otherwise noted, the SCP response messages are not processed any differently following a STR-Connection.

It is important to note that the originating call model has remained at the same TDP during the entire STR-Connection. Therefore, the response message returned by the SCP must be allowed for that TDP. For example, if a call triggers at the ***Info_Collected*** TDP and connects to an IP, a `Continue` message is not allowed as a response following the `Resource_Clear`.

When the SCP responds with a `Continue` message, the local switch continues trigger evaluation. This includes the following actions:

- The local switch checks for additional triggers enabled at the current TDP.
- The local switch checks for additional triggers enabled by other subscribing CAIN groups at the current TDP.

When no additional triggers are available, the local switch continues with in-switch call processing. Depending upon the current TDP, this may result in different actions. An example is listed below:

- A call originates at the local switch on an agent which subscribes to CAIN.
- As the call proceeds through the call model, address digits are collected and translated using in-switch tables to identify a route index.
- Once the route index is identified, the ***Info_Analyzed*** TDP is reached and the appropriate trigger tables are checked.
- Since the trigger table specifies that a query is required, call processing is suspended and an **`Info_Analyzed`** message is sent to the SCP.
- The SCP responds with a **`Send_To_Resource`** containing a ***`DestinationAddress`*** parameter. Using this information, the local switch establishes a connection with the specified IP.
- When the IP finishes providing the requested function, it releases the call. At this point, the local switch sends a **`Resource_Clear`** to the SCP and awaits further instructions.
- The SCP responds with a **`Continue`** message. The local switch checks for additional triggers enabled at the ***Info_Analyzed*** TDP against the current CAIN group, or other subscribing CAIN groups.
- Since no additional triggers are enabled, the local switch proceeds to the **Select_Route** PIC and uses the route index identified prior to the query.

Depending upon the SCP response (for example an **`Analyze_Route`** message), the local switch may attempt to establish a second leg for the call. The second leg behaves as a normal trunk-to-trunk connection, with the following exception:

- On SS7 originations, an ACM is not sent to the originating switch on the second leg of the call. An ACM is only sent to the originating switch during the first leg of the call.

Figures 2-64 and 2-65 provide CONNECT_ONLY IPI examples of second leg calls.

Figure 2-64
SS7 originator with an Analyze_Route message on the second leg

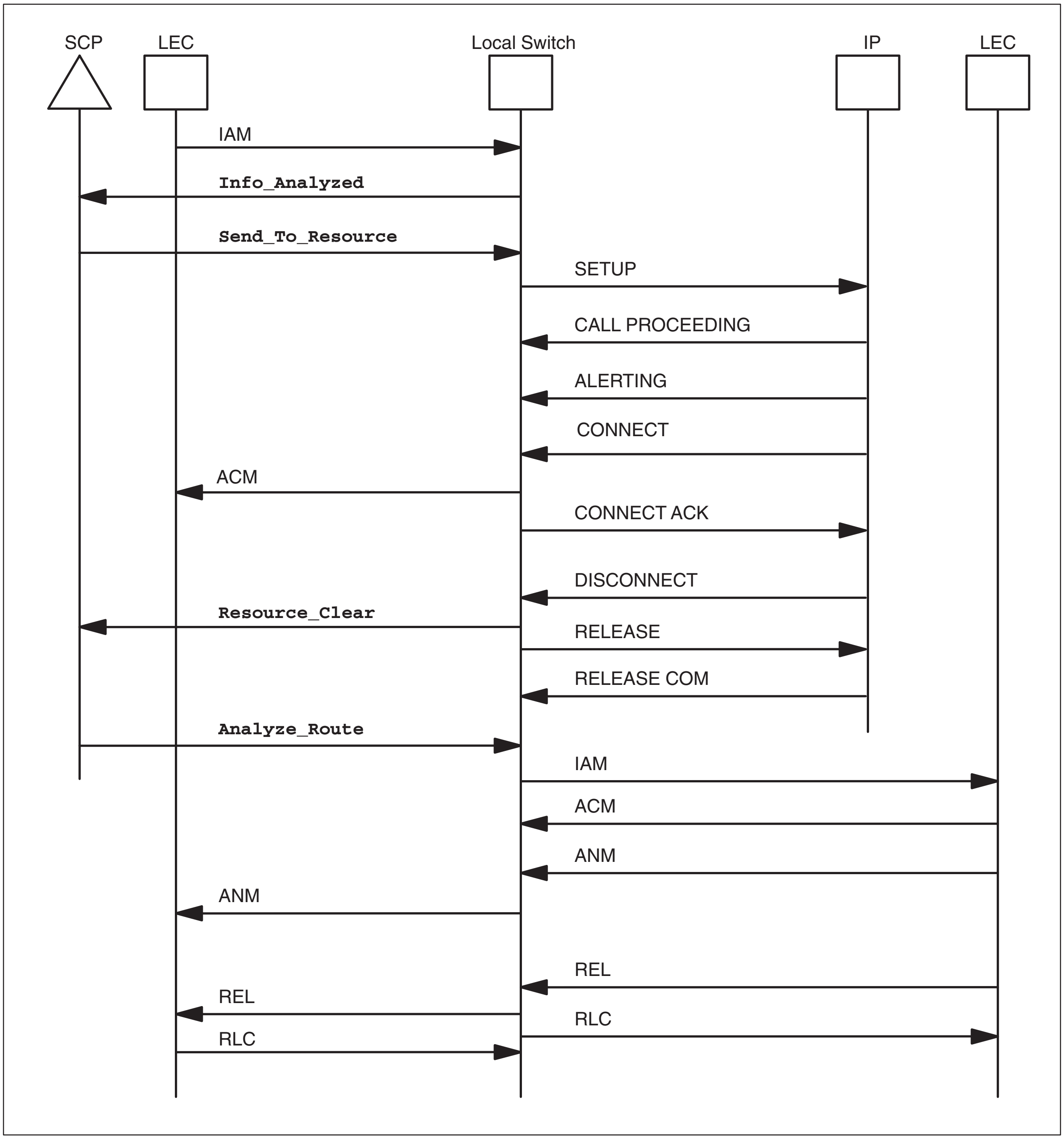
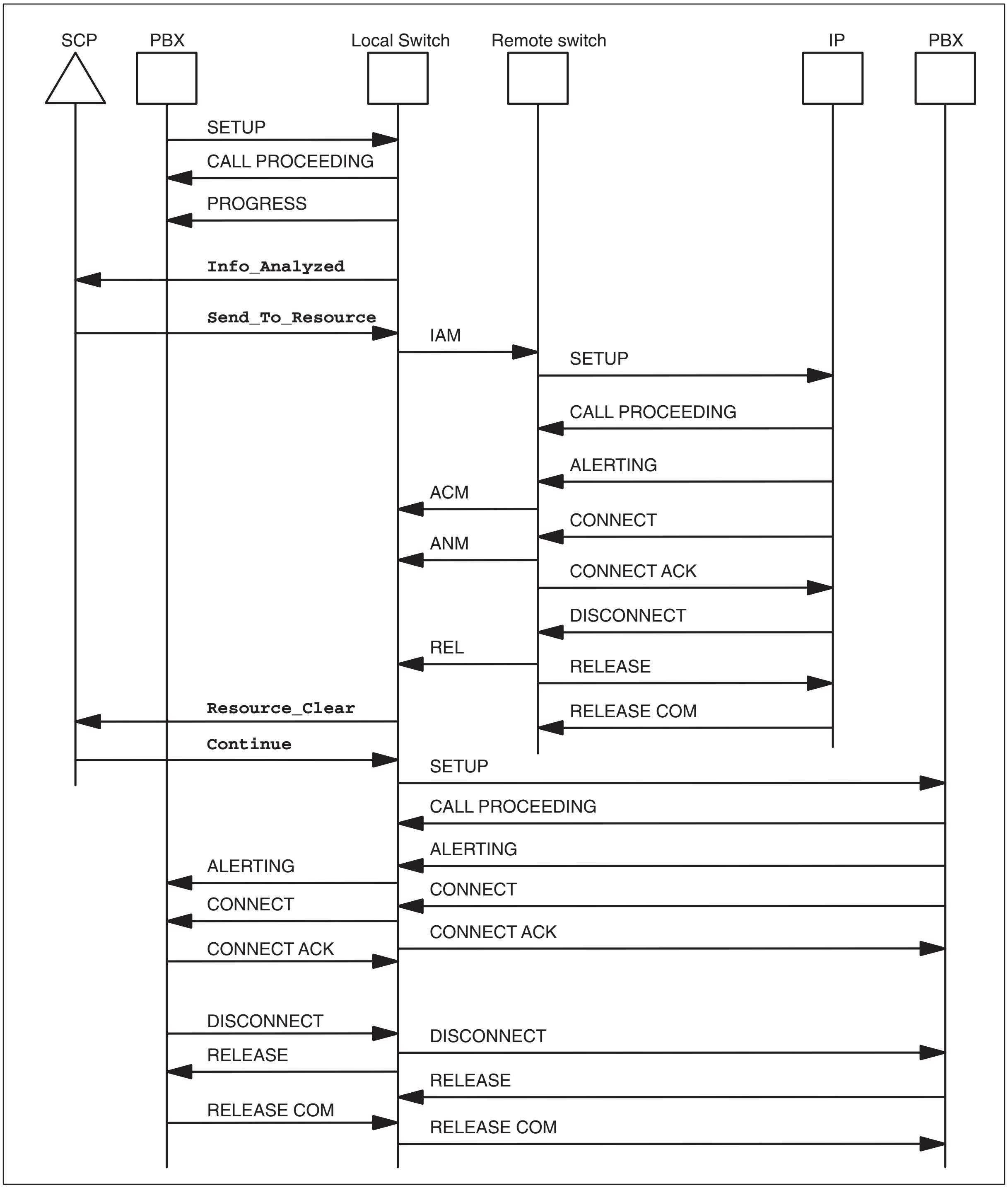


Figure 2-65
PRI originator with Continue message on the second leg



Billing

The following CDR fields are affected by STR-Connection:

- Upon IP connection: CALLEDNO, CNPREDIG, OPART, TPART
- `Send_To_Resource` with ***`AMAMeasure`*** parameter: ANSTYPE and CALLDUR
- Upon connection to IP: RTELIST, RTENO, TERMGRP, and TERMMEM
- Upon disconnection from second leg: DISCTIME, DISCDATE, DISCAMPM, and DISCTYPE

Upon IP connection

When the local switch establishes a STR-Connection to an IP, the following CDR fields are updated:

- CALLEDNO and CNPREDIG - These CDR fields are updated based upon the digits contained in the ***`DestinationAddress`*** parameter.
- OPART and TPART - These CDR fields are updated based upon the STS used to establish the STR-Connection to the IP.

Once the STR-Connection is completed, the SCP may provide additional routing instructions in order to establish a second leg. This additional routing information may overwrite the CDR fields updated during the STR-Connection as described below:

- `Analyze_Route` - Depending upon the contents, this message may overwrite the values stored in CDR fields CALLEDNO, CNPREDIG, OPART, and TPART.
- `Continue` - When this message is received, CAIN overwrites the values for the CDR fields CALLEDNO, CNPREDIG, OPART, and TPART with the original pre-query values for these fields.

Send_To_Resource without AMAMeasure

The following CDR fields are affected for STR-Connections when the `Send_To_Resource` message does not contain the ***`AMAMeasure`*** parameter:

- ANSTYPE - This field is updated with the answer type when the second leg of the call is answered. Existing switch answer type values are used (for example, hardware answer = 04).
- CALLDUR - This field contains the call duration for the second leg of the call. The call duration is started when the second leg answers and is stopped when either the called or calling party disconnects.

These fields are only captured when a second leg is established following a STR-Connection, and the second leg is answered. Table 2-10 and Figure 2-66 provide several examples.

Table 2-10
Answer type and call duration values without AMAMeasure parameter

Scenario	ANSTYPE	CALLDUR
A local switch is unable to establish a STR-Connection due to a fatal application error. Final treatment is provided.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives a **`Disconnect`** message from the SCP. Treatment is applied.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives an **`Analyze_Route`** message from the SCP. The second leg is established and hardware answer is detected after several rings.	The answer type indicates hardware answer (04).	The call duration contains the total time from the second leg answer to the second leg disconnect.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives an **`Analyze_Route`** message from the SCP. The second leg is established, but the call is never answered.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The calling user abandons during an active STR-Connection.	The answer type indicates no answer (00).	No duration value is set in the CDR.

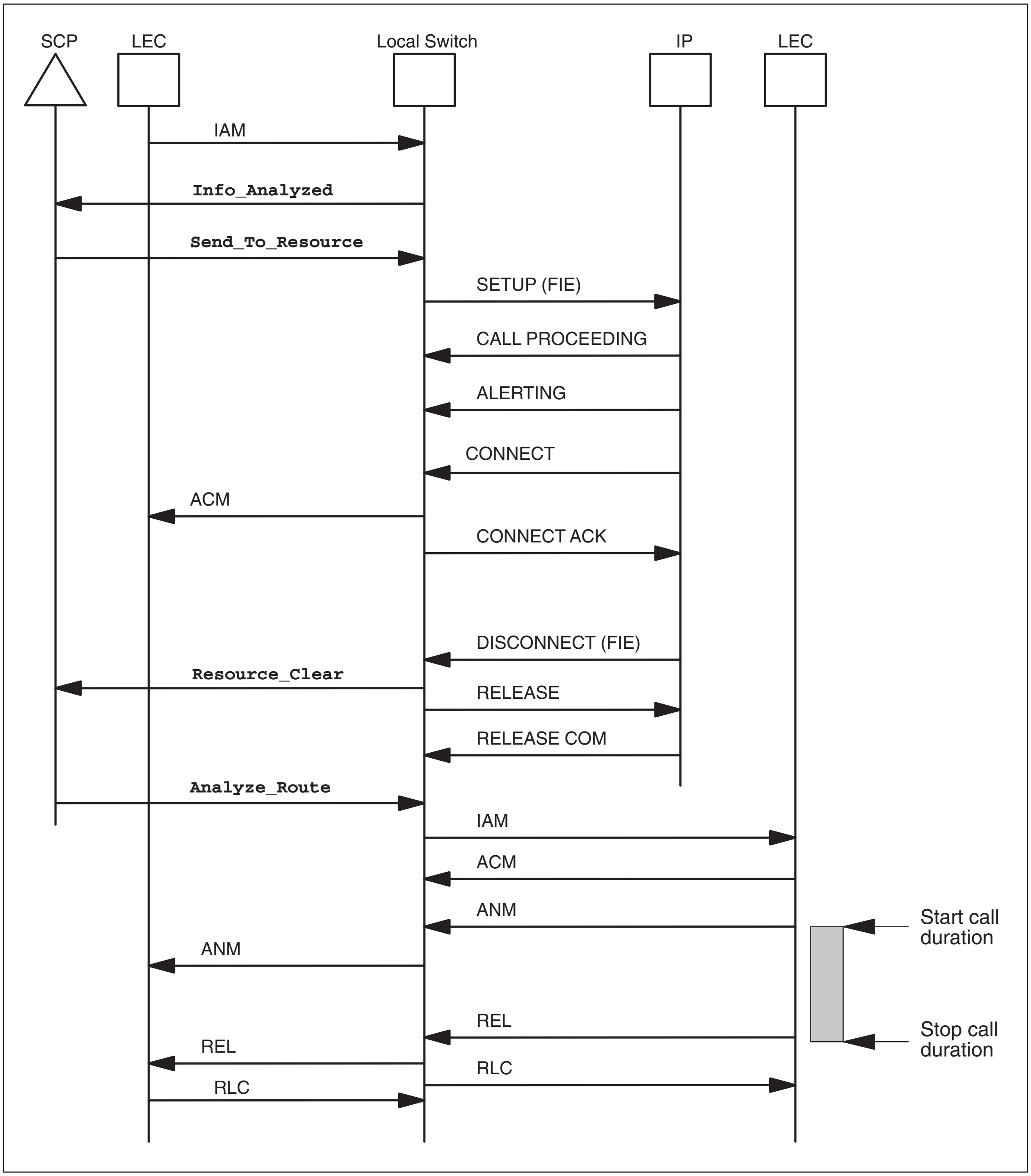
—continued—

Table 2-10
Answer type and call duration values without AMAMeasure parameter (continued)

Scenario	ANSTYPE	CALLDUR
The local switch receives a **`Cancel_Resource_Event`** message during an active CONNECT_ONLY STR-Connection. The local switch sends a **`Resource_Clear`** message and receives an **`Analyze_Route`** message from the SCP. A second leg is established and hardware answer is detected after several rings.	The answer type indicates hardware answer (04).	The call duration contains the total time from the second leg answer to the second leg disconnect.
The local switch establishes a CONNECT_ONLY STR-Connection to a local IP. Once the STR-Connection is completed, a **`Resource_Clear`** message is sent and an **`Analyze_Route`** message is received from the SCP. A second leg is established and hardware answer is detected after several rings. When the called party disconnects, the calling user reoriginates.	The answer type indicates hardware answer (04).On the reoriginated call, the ANSTYPE field is unaffected by the previous STR-Connection.	The call duration contains the total time from the second leg answer to the second leg disconnect.On the reoriginated call, the CALLDUR field is unaffected by the previous STR-Connection.
—end—		

Figure 2-66 provides a CONNECT_1129_STYLE IPI example of the call duration when the **`Send_To_Resource`** message is received without an ***`AMAMeasure`*** parameter. For the CONNECT_ONLY IPI, the ISDN SETUP and DISCONNECT messages do not contain the FIE.

Figure 2-66
Call duration without the AMAMeasure parameter



Send_To_Resource with AMAMeasure

The ANSTYPE and CALLDUR CDR fields are affected when the **`Send_To_Resource`** message contains the ***`AMAMeasure`*** parameter with the value of `connectTimeDestinationRecordedSSP`. If the ***`AMAMeasure`*** parameter is received with any other value it will be as if the parameter was not received. The following sections identify how the ANSTYPE and CALLDUR CDR fields are affected.

ANSTYPE

For normal calls, the ANSTYPE field of the CDR is set when the called party answers. Also upon called party answer, call duration timing is started. Therefore on these calls, the ANSTYPE and CALLDUR fields are closely related since both are based upon called party answer.

During a STR-Connection with the ***`AMAMeasure`*** parameter, the call duration timing is started when the IP answers, not when the called party answers on the second leg. Therefore, a CDR may contain a call duration even if the called party never answered on the second leg or if there wasn’t a second leg.

Since the call duration is not based upon called party answer, two new answer types are added to indicate that CAIN is controlling the call duration:

- 12 (early billing without answer) - This value is placed into the CDR when the IP answered, but the called party on the second leg did not answer. This value is also used when the IP answered, and a second leg of the call was not established.
- 13 (early billing with answer) - This value is placed into the CDR when the IP answered, and the called party on the second leg also answered.

It is important to note that these values are only used when the IP answers. After an abnormal clearing of a STR-Connection, a second leg may be established even though the IP never answered. When this occurs, existing answer type values are used and the CDR is updated when the called party answers on the second leg. Therefore in this scenario, the ***`AMAMeasure`*** parameter has no affect upon the ANSTYPE CDR field.

Figure 2-67 and Table 2-11 provide examples for the ANSTYPE CDR field when the ***`AMAMeasure`*** parameter is present.

CALLDUR

On normal calls, call duration timing is started when the called party answers. The timing continues until the called or calling party disconnects.

During a STR-Connection with the ***`AMAMeasure`*** parameter, call duration timing is started when the IP answers, not when the called party answers on the second leg. The call duration timing is not stopped when the IP releases the call. Instead, call duration timing continues until either the calling party or called party (second leg) disconnects. Therefore when the ***AMAMeasure*** parameter is present, a CDR may contain a call duration even if the called party never answered on the second leg or if there wasn't a second leg.

On normal calls, call duration only contains the time that the calling party and called party are connected (talking). When the ***AMAMeasure*** parameter is present, the call duration may contain additional time due to SCP response delays, network delays, called party alerting. This occurs since the call duration timer runs continuously until call completion. It is does not stop when the IP disconnects and then restart upon second leg answer. An example is listed below:

1 The local switch establishes a STR-Connection to a local IP. The call duration is started when the IP answers.
2 Once the IP provides the requested functionality, the IP initiates normal call clearing. The local switch sends a **Resource_Clear** message to the SCP. The call duration timer continues running.
3 The SCP responds with an **Analyze_Route** message, and the second leg is established to the called party. Since the call duration timer is still running, the duration will include the time spent waiting for an SCP response and the time spent establishing the second leg.
4 After several rings, the called party answers. Since the call duration timer is still running, the duration will include the time that the phone was ringing.
5 The called party disconnects. The call duration timer is stopped.

It is important to note that the call duration is only affected by the ***AMAMeasure*** parameter when the IP answers. After an abnormal clearing of a STR-Connection, a second leg may be established even though the IP never answered. When this occurs, the call duration is calculated as in normal calls.

Figure 2-67 and Table 2-11 provide further examples for the CALLDUR CDR field when the ***AMAMeasure*** parameter is present.

Figure 2-67 provides a CONNECT_1129_STYLE IPI example of the call duration when the **Send_To_Resource** message is received with an ***AMAMeasure*** parameter. For the CONNECT_ONLY IPI, the ISDN SETUP and DISCONNECT messages do not contain the FIE.

Figure 2-67
Call duration with the AMAMeasure parameter

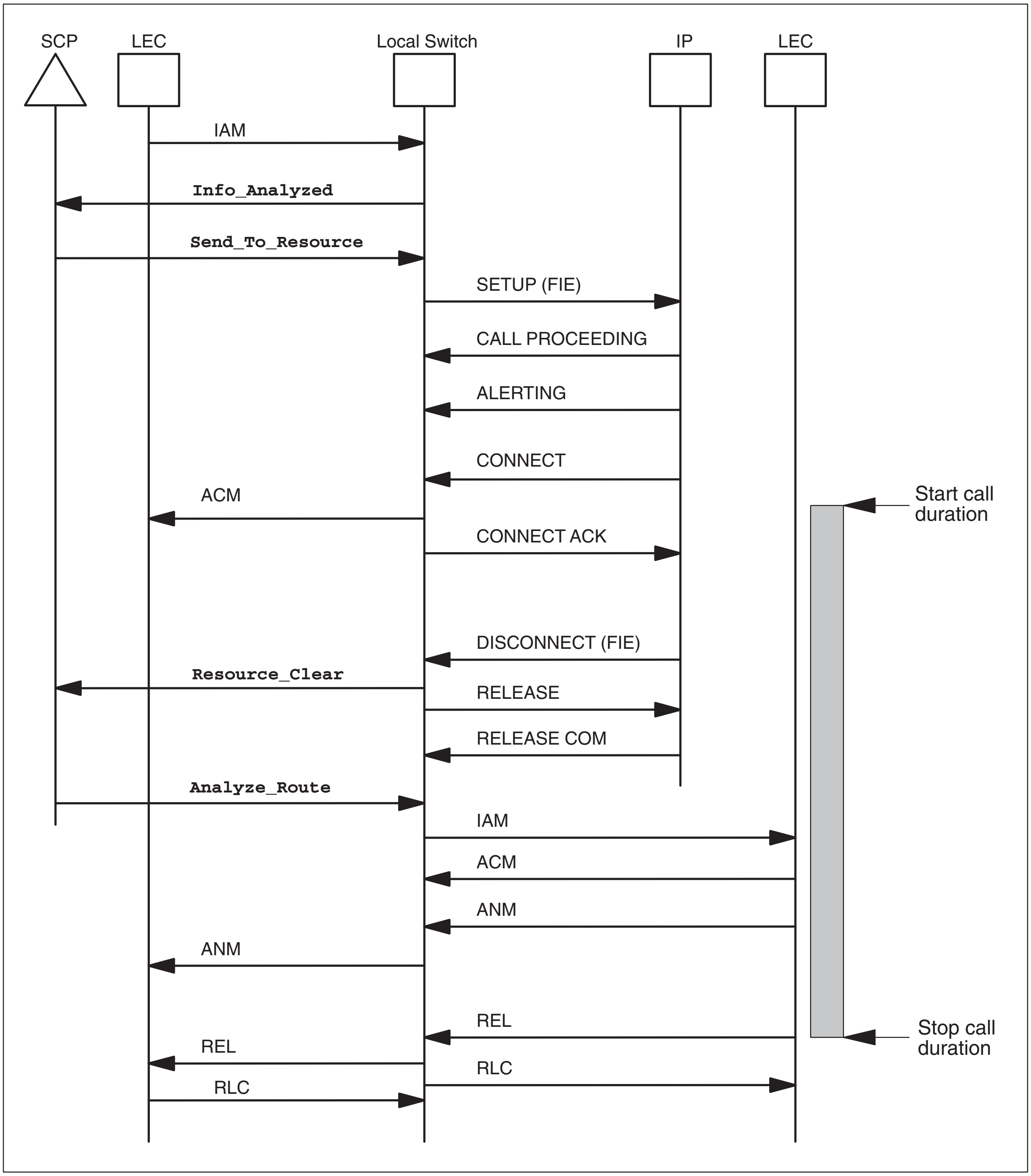


Table 2-11
Answer type and call duration values with AMAMeasure parameter

Scenario	ANSTYPE	CALLDUR
A local switch is unable to establish a STR-Connection due to a fatal application error. Final treatment is provided.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives a **`Disconnect`** message from the SCP. Treatment is applied.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives an **`Analyze_Route`** message from the SCP. The second leg is established and hardware answer is detected after several rings.	The answer type indicates hardware answer (04).	The call duration contains the total time from the second leg answer to the second leg disconnect.
The local IP initiates abnormal call clearing by sending a RELEASE message. The local switch sends a **`Resource_Clear`** and receives an **`Analyze_Route`** message from the SCP. The second leg is established, but the call is never answered.	The answer type indicates no answer (00).	No duration value is set in the CDR.
The calling user abandons during an active STR-Connection.	The answer type indicates that the IP answered, but that there was no answer on the second leg (12).	The call duration timing is started when the IP answers and stopped when the calling user abandons.
The local switch receives a **`Cancel_Resource_Event`** message during an active STR-Connection. The local switch sends a **`Resource_Clear`** message and receives an **`Analyze_Route`** message from the SCP. A second leg is established and hardware answer is detected after several rings.	The answer type indicates that the IP answered, and that there was an answer on the second leg (13).	The call duration timing is started when the IP answers and stopped when the called party disconnects on the second leg.

—continued—

Table 2-11
Answer type and call duration values with AMAMeasure parameter (continued)

Scenario	ANSTYPE	CALLDUR
The local switch establishes a STR-Connection to a local IP. Once the STR-Connection is completed, a `Resource_Clear` message is sent and an `Analyze_Route` message is received from the SCP. A second leg is established and hardware answer is detected after several rings. When the called party disconnects, the calling user reoriginates.	The answer type indicates that the IP answered, and that there was an answer on the second leg (13).On the reoriginated call, the ANSTYPE field is unaffected by the previous STR-Connection.	The call duration timing is started when the IP answers and stopped when the calling user reoriginates.On the reoriginated call, the CALLDUR field is unaffected by the previous STR-Connection.
The local switch establishes a STR-Connection to a local IP. Once the STR-Connection is completed, a `Resource_Clear` message is sent and a `Disconnect` message is received from the SCP. The specified treatment results in a reorder tone being applied. The calling party disconnects after several seconds of listening to the reorder tone.	The answer type indicates that the IP answered, but that there was no answer on the second leg (12).	The call duration timing is started when the IP answers and stopped when the calling party disconnects. Therefore, the call duration includes the time connected to the reorder tone.
—end—		

Upon connection to IP

When the local switch establishes a CONNECT_ONLY STR-Connection to an IP, it updates the following CDR fields:

- RTELIST, RTENO - These CDR fields are updated to identify the route list and number resulting from the translation of the ***`DestinationAddress`*** parameter.
- TERMGRP, TERMMEM - These CDR fields are updated to identify the terminating trunk group and member used to establish the STR-Connection to the IP.

Once the STR-Connection is completed, the SCP may provide additional routing instructions in order to establish a second leg. This additional routing information may overwrite the CDR fields described above.

Upon disconnection from second leg

CDR fields DISCTIME, DISCDATE, DISCAMPM, and DISCTYPE are updated when the calling party disconnects or when the called party disconnects on the second leg. These fields are not updated when the IP initiates normal or abnormal call clearing during a STR-Connection.

Fatal application errors

Fatal application errors occur when CAIN call processing is unable to continue due to an unexpected error. Table 2-12 shows errors that can occur during a STR-Connection.

Table 2-12
Fatal application errors

Error type	Log generated	Reported to SCP?	Action performed
DestinationAddress parameter present in a **Send_To_Resource** in a Response package	CAIN200	Yes (Note 1)	Switch applies AINF
DestinationAddress and ***DisconnectFlag*** present in a **Send_To_Resource** in a Response or Conversation package	CAIN200	Yes (Note 2)	Switch applies AINF
Local switch is unable to identify a route index for the IP	CAIN200	Yes (Note 2)	Switch applies AINF
Selected agent does not use ISDN signaling (between switch and IP)	CAIN200	Yes (Note 2)	Switch applies AINF
Selected agent does not use SS7 signaling (between tandem switches)	CAIN200	Yes (Note 2)	AINF

Note 1: The switch sends an **Application_Error** to the SCP, with ***ErrorCause*** set to “unexpectedCommunication”.

Note 2: The switch sends an **Application_Error** to the SCP, with ***ErrorCause*** set to “erroneousDataValue”.

Nonfatal application errors

Table 2-13 shows errors that can occur during a STR-Connection.

Table 2-13
Nonfatal application errors

Error type	Log generated	Reported to SCP?	Action performed
DestinationAddress contains a nature of address value other than National or VPN	CAIN100	No	National NOA is assumed for the ***DestinationAddress***.
Switch receives a **Send_To_Resource** message with a ***DestinationAddress*** parameter and the STR_CONNECTION_TYPE (table CAINPARM) is set to NONE.	CAIN100	Yes	**Resource_Clear** is sent to SCP in a conversation package with a ***ClearCause*** of taskRefused.
DestinationAddress contains too many digits	CAIN100	No	Excess digits are truncated

Note: For more information on nonfatal application errors, refer to Volume 3, Chapter 10, “Incoming CAIN messages.”

Associated logs

CAIN100, CAIN200

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF, CAINIP

Limitations and restrictions

Generate CDR upon answer is not supported during an STR-Connection to an IP. Therefore, a CDR or log is never generated upon IP answer, regardless of the datafill. However, this feature is supported on second leg calls.

SS7 RLT Enhancements

CAIN does not support SS7 RLT functionality during a STR-Connection to an IP. However, it is supported on the second leg of the call. When the ***AMAMeasure*** parameter is not received from the SCP, the RLT billing functionality is unaffected by CAIN.

When the ***AMAMeasure*** parameter was received from the SCP, the RLT billing functionality is overridden by CAIN. In this scenario, call duration timing starts when an answer indication is received from the IP, regardless of whether the RLT functionality specifies first or last ANM billing. The call duration timer continues running until the call is disconnected. The following scenario provides an example:

- The local switch establishes a STR-Connection to a local IP. The call duration is started when the IP answers.
- Once the IP provides the requested functionality, the IP initiates normal call clearing. The local switch sends a **`Resource_Clear`** message to the SCP. The call duration timer continues running.
- The SCP responds with an **`Analyze_Route`** message, and the second leg is established to the services platform. The office parameter RLT_FIRST_ANM_BILLING is set to N. Since the call duration timer is still running, the duration will include the time spent waiting for an SCP response and the time spent establishing the second leg to the services platform. The call duration is unaffected by the RLT office parameter.
- The services platform responds with a FAR message requesting that the call be redirected to another party. The RLT trunks are released, and the switch establishes a third leg to the called party specified by the services platform. Since the call duration timer is still running, the duration includes the time spent connected to the services platform and the time spent establishing the third leg to the called party.
- After several rings, the called party answers. Since the call duration timer is still running, the duration will include the time that the phone was ringing.
- The called party disconnects. The call duration timer is stopped.

When the services platform attempts to override billing via a FAR message, the action taken is dependent upon the operation requested in the message. When a redirection or bridging FAR is received, the call duration is not affected and the requested operation is performed. When a start or cancel billing FAR is received, a FRJ message is sent to the services platform with a cause indicator value of “previous billing determination”. The call duration is not affected.

CTR-Connections

ATTENTION
Use of the **`Connect_To_Resource`** message requires the CAIN0801 SOC option. Refer to Volume 5, Chapter 5, "NetworkBuilder SOC functionality," for more information.

In AIN 0.2, the Intelligent Peripheral (IP) was introduced as a component of the AIN Architecture. The IP contains functionality and resources capable of exchanging information with a subscriber, such as:

- playing pre-recorded announcements or music
- collecting dual tone multi-frequency (DTMF) digits
- recording voice or modulated voice information
- playing recorded voice or modulated voice information
- performing speaker-dependent or speaker-independent voice recognition

AIN 0.2 supports two types of connections to an IP:

- an STR-or CTR-Connection (an IP connection in response to a **`Send_To_Resource`** message is referred to as a STR-Connection; an IP connection in response to a **`Connect_To_Resource`** message is referred to as a CTR-Connection)
- a connection resulting from a normal termination attempt (for example, the subscriber dials an address served by an IP), referred to as a termination

Within the **`Connect_To_Resource`** message, the ***`DestinationAddress`*** parameter identifies the location of an IP resource. The UCS DMS-250 switch interprets a **`Connect_To_Resource`** message without a ***`DestinationAddress`*** parameter as a request to access an internal switch resource rather than an IP resource.

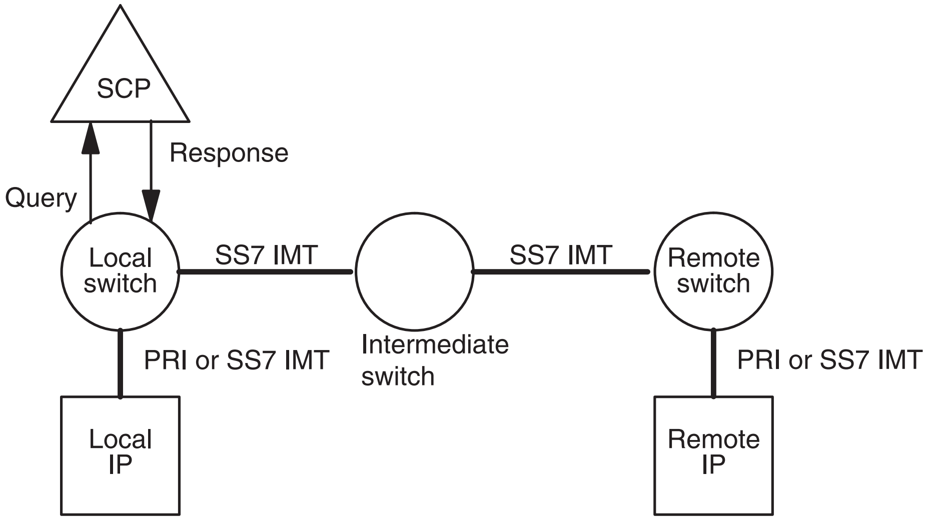
The **Connect_To_Resource** message is supported only in response to the **O_Mid_Call** TDP-Request or EDP-Request, **O_Disconnect** EDP-Request, and **Timeout** EDP-Request messages.

Note: The **Connect_To_Resource** message functions the same for both two party and multi-party calls. For complete information on **Connect_To_Resource** interactions during multi-party calls, refer to Volume 3, Chapter 4,"Call configuration model."

Terminology

Figure 3-1 shows a network diagram and the terminology used in association with a CTR-Connection.

Figure 3-1
Network diagram with an IP



Local switch — The switch where a TDP was encountered and trigger criteria met. The switch queries the SCP for data relating to the processing of the call.

Local IP — An IP with a direct ISDN or SS7 connection to the local switch. The IP is accessed when the SCP requests a CTR-Connection from the local switch.

Intermediate switch — A tandem switch used to complete the connection between a local switch and a remote switch, when a direct connection between the local and remote switch is not available.

Remote switch — A tandem switch used to complete the connection between a local switch and a remote IP. A remote switch is used when the local switch does not have a direct ISDN or SS7 connection to the desired IP.

Remote IP — An IP that does not have a direct ISDN or SS7 connection to the local switch. The local switch must connect to another switch that has a direct ISDN or SS7 connection to the IP.

Although the ***DestinationAddress*** parameter is listed as an optional parameter for the **Connect_To_Resource** message, this discussion of CTR-Connections assumes the ***DestinationAddress*** parameter is present.

Note: Before UCS08, information the ***AMAMeasure*** parameter currently supplies was supplied by the ***AnswerIndicator*** parameter.

Table 3-1 provides the CTR-Connection parameters for the **`Connect_To_Resource`** message.

Table 3-1
CTR-Connection parameters

Parameter	Usage	Definition
`ResourceType`	Required	The CAIN framework ignores the contents of this parameter. However, any data passed to the UCS DMS-250 switch must be properly formatted and encoded to ensure proper decoding at the switch.
`StrParameterBlock`	Required	The CAIN framework ignores the contents of this parameter. However, any data passed to the UCS DMS-250 switch must be properly formatted and encoded to ensure proper decoding at the switch.
`LegID`	Optional	This parameter specifies which call leg the UCS DMS-250 switch should connect to the resource.
`AMADigitsDialedWC`	Optional	This parameter contains digit strings to be populated into one or more of the following CDR fields: PINDIGS, ACCTCD, BILLNUM, CIC, ORIGPVN, or TERMPVN
`DestinationAddress`	Optional	This parameter contains the address of the intelligent peripheral. This address is used with an STS to identify a route index for the CTR-Connection. The UCS DMS-250 switch uses an STS from one of the following: • `servTranslationScheme` extension parameter in the **`Connect_To_Resource`** message • default STS provisioned in table CAINXDFT • STS identified through existing call processing software (for example, table PARTOSTS)

—continued—

Table 3-1
CTR-Connection parameters (continued)

Parameter	Usage	Definition
AMAMeasure	Optional	When present with the value of `connectTimeRecordedSSP`, this parameter indicates that the call duration (CALLDUR field of the CDR) includes the time connected to an IP. This parameter has no affect on answer indication. The ANSTYPE field of the CDR is updated to indicate early billing (prior to the second call leg answering). If the value is `connectTimeRecordedDestinationSCP` or `connectTimeNotRecorded`, no timing is begun. If more than one **Send_To_Resource** or **Connect_To_Resource** is sent during a single call, subsequent ***AMAMeasure*** parameters will not reset nor stop timing, but will start timing if it has not already begun. ***Note:*** If the ***AMAMeasure*** parameter is received with a value other than `connectTimeRecordedSSP`, it is treated as if the parameter is not present.
ExtensionParameter		Extension parameters require the CAIN0200 SOC option.
servTranslation Scheme	Optional	This extension parameter contains a serving translation scheme.
billSequence Number	Optional	This extension parameter contains 32 bits of SCP-defined billing data that is stored in the CDR.
strConnection Type	Optional	This extension parameter indicates the type of connection protocol (IPI) to be used to establish communication between the SCP, switch, and an IP resource.
	—end—	

Connectivity to an IP

IP connectivity defined by Bellcore's *GR-1129-CORE* specifies that only ISDN signaling is allowed for connections between a UCS DMS-250 switch (local or remote) and an IP. It only supports SS7 signaling for connections between local and remote switches. However, NetworkBuilder expands upon the IP connectivity defined by Bellcore's *GR-1129-CORE* allowing direct SS7 connectivity to an IP in addition to direct ISDN connectivity.

Timer TDISC

Timer TDISC provides a maximum time limit for the IP to respond to a FAR message with the `cancelIPResource` operation. The TDISC_TIMER field in table CAINPARM determines the maximum time duration for the TDISC timer.

- Range – 1 to 4 seconds
- Default – The timer will have a default value of 4 seconds.

Timer TSTRC

Timer TSTRC provides a maximum time limit for a CTR-Connection to an IP. It is started when the IP answers and canceled when the CTR-Connection is cleared, either by the UCS DMS-250 switch or IP. The TSTRC_TIMER field in table CAINPARM determines the time duration for the TSTRC timer.

- Range – 0 to 60 minutes, with the value of 0 disabling the timer
- Default – The timer will have a default value of 6 minutes.

SS7 connectivity

The IPTRUNK option in table TRKGRP is used on an ISUP IMT trunk to determine if the trunk is directly connected to an IP or connected to another switch. The presence of the IPTRUNK option indicates that the ISUP IMT trunk is directly connected to an IP, where absence of the IPTRUNK option indicates the trunk is connected to another switch. The IPTRUNK option is only available when the GRPTYPE field in table TRKGRP for the trunk is either IMT or PRA250.

The local switch normally provides the same functionality, regardless of whether an ISUP IMT trunk is used to connect to a local IP or remote switch. However, because the TDISC and TSTRC timers are maintained by the switch directly connected to the IP, these timers are started at the local switch when directly connecting to an IP on an IMT trunk that includes the IPTRUNK option. Otherwise, the timers are started at the remote switch that is directly connected to the IP.

Note: For the IMT trunk type it is also necessary for the ISUPIDX field in table TRKGRP to contain the value UCS2UCS.

ATTENTION

For an IP to use this functionality, the SS7 signaling requirements defined by Bellcore's *GR-1129-CORE* and those stated here must be followed.

PRI connectivity

The IPTRUNK option in table TRKGRP is used on a PRI trunk to determine whether or not the trunk is directly connected to an IP. The presence of the IPTRUNK option indicates that the PRI trunk is directly connected to an IP, where absence of the IPTRUNK option indicates the trunk is not connected to an IP. The IPTRUNK option is only available when the GRPTYPE field in table TRKGRP for the trunk is either IMT or PRA250.

Note: PRI trunks cannot be used as tandem trunks.

The local switch normally provides the same functionality, regardless of whether the PRI trunk includes the IPTRUNK option or not. However, because the TDISC and TSTRC timers are maintained by the switch directly connected to the IP, these timers are started at the local switch when directly connecting to an IP on a PRI trunk that includes the IPTRUNK option. Otherwise, the timers are not started.

Intelligent Perpheral Interface (IPI) overview

In a STR-Connection the UCS DMS-250 switch and IP communicate through an IPI. An IPI provides an interface that is met by both a switch and IP. NetworkBuilder supports the following IPIs:

- CONNECT_ONLY
- CONNECT_1129_STYLE

ATTENTION

The CONNECT_ONLY IPI was intended to provide initial support for connecting to an IP until the development of the CONNECT_1129_STYLE IPI could be completed.

Nortel Networks now recommends the use of the CONNECT_1129_STYLE IPI since it provides similar functionality with additional enhancements.

Although both the CONNECT_ONLY and CONNECT_1129_STYLE IPIs are based upon the IPI defined by Bellcore's *GR-1129-CORE*, there is a significant difference between the two. As outlined below, the CONNECT_ONLY IPI does not support the exchange of data between the SCP and IP through the switch.

Note: The Global IMT SOC (CAIN0605) is not supported. If a **`Connect_To_Resource`** message is received with a ***`DestinationAddress`*** parameter, a **`CTR_Clear`** message with a ***`ClearCause`*** of `taskRefused` will be sent to the SCP.

CONNECT_ONLY IPI

The CONNECT_ONLY IPI is intended for use when there is a direct communication link between the SCP and IP. Bellcore does not define this interface, nor does it prohibit an IP from directly communicating to an SCP or a Service Management System (SMS). It provides the following capabilities:

- Creates a connection between a subscriber and an IP via a voice channel in the switch and allows an IP to use its internal resources and functionality to exchange information with that subscriber.
- Allows calls to originate and terminate over the IPI.

CONNECT_1129_STYLE IPI

The CONNECT_1129_STYLE IPI is based upon the IPI defined by Bellcore's *GR-1129-CORE*. It provides the following capabilities:

- Creates a connection between a subscriber and an IP through a voice channel in the switch and allows an IP to use its internal resources and functionality to exchange information with that subscriber.
- Allows a switch to interwork an exchange of data between an IP and the SCP.
- Allows calls to originate and terminate over the IPI.

Signaling

For the CONNECT_ONLY IPI, ISDN and SS7 signaling are supported for call establishment and call clearing only. Without a communication link between the SCP and IP, services requiring an exchange of data will not function correctly.

For the CONNECT_1129_STYLE IPI, ISDN and SS7 signaling are supported for call establishment and call clearing, and direct ISDN and SS7 connections between a switch (local or remote) and an IP.

Note: The IPI defined by Bellcore's *GR-1129-CORE* specifies that only ISDN signaling is allowed for connections between a switch (local or remote) and an IP. It only supports SS7 signaling for connections between local and remote switches.

Signaling for either IPI involves:

- messaging between the switch and SCP
- messaging between the switch and IP
- messaging between switches for remote IP access

Signaling using the CONNECT_ONLY IPI

Carrier AIN CTR-Connections with the CONNECT_ONLY IPI have limited data exchange capabilities as listed above. The following example of a CONNECT_ONLY illustrates the limitations:

1 A call triggers at a TDP. The UCS DMS-250 switch temporarily suspends call processing and launches a query to an SCP.

2 The SCP responds with a **`Connect_To_Resource`** message containing a ***`DestinationAddress`*** parameter, which indicates a CTR-Connection to an IP is needed.

3 The UCS DMS-250 switch translates the address contained in the ***`DestinationAddress`*** parameter to identify a route list for terminating to the IP.

4 The UCS DMS-250 switch initiates a connection to the IP by sending a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a remote IP).

Limitation: Bellcore specification *GR-1129-CORE* specifies that the information contained in the ***`StrParameterBlock`*** is passed to the IP using a Facility Information Element (FIE) for ISDN, or a Remote Operations (RO) parameter for SS7. The CONNECT_ONLY IPI does not support this interaction. Data in the ***`StrParameterBlock`*** is discarded.

5 The UCS DMS-250 switch establishes a connection with the IP.

Limitation: During an active connection to an IP, Bellcore's *GR-1129-CORE* states that data may be exchanged between the SCP and IP using a FACILITY message between the switch and IP and a **`Call_Info_From_Resource`** or **`Call_Info_To_Resource`** message between the switch and SCP. The CONNECT_ONLY IPI does not support this functionality.

6 Once the IP completes its function, it releases the call and the UCS DMS-250 switch sends a **`CTR_Clear`** message to the SCP and awaits further instructions.

Limitation: When the IP releases the call, Bellcore's *GR-1129-CORE* specifies that the data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***`IPReturnBlock`*** parameter of the **`Resource_Clear`** message. The CONNECT_ONLY IPI does not support this functionality.

Due to these limitations with the CONNECT_ONLY IPI, services requiring an exchange of data will not function correctly, unless there is direct communication between the SCP and IP.

Limitations and restrictions

For the CONNECT_ONLY IPI, STR-Connections have the following limitations and restrictions:

- During a CONNECT_ONLY STR-Connection, originating call model triggers on the local switch are not evaluated. Once the CTR-Connection is completed (the switch sends a **`CTR_Clear`** message to the SCP), the originating call model triggers may again be evaluated.
- During an active CONNECT_ONLY CTR-Connection, manual and auto reorigination is blocked. Once the CTR-Connection is completed, reorigination is re-enabled when necessary.
- If a fatal application error is detected during a CONNECT_ONLY CTR-Connection, final treatment is always applied. Default routing is not supported for fatal errors encountered during a CTR-Connection.
- In UCS08, terminating call model triggers on the local switch are not evaluated.

Signaling using the CONNECT_1129_STYLE IPI

The following example illustrates Carrier AIN CTR-Connections with the CONNECT_1129_STYLE IPI:

1. A call triggers at a TDP. The switch temporarily suspends call processing and launches a query to an SCP.
2. The SCP responds with a **`Send_To_Resource`** message containing a ***`DestinationAddress`*** parameter, which indicates a CTR-Connection to an IP is needed. In addition to the address of the IP, the SCP provides an ***`StrParameterBlock`*** which contains variable information needed by the IP such as, the requested resource and the function to be performed.
3. The UCS DMS-250 switch translates the address contained in the ***`DestinationAddress`*** parameter to identify a route list for terminating to the IP.
4. The UCS DMS-250 switch initiates a connection to the IP by sending a SETUP message (for ISDN terminations to a local IP) or an Initial Address Message (IAM) (for SS7 terminations to a local or remote IP). The information contained in the ***`StrParameterBlock`*** is passed to the IP using a Facility Information Element (FIE) for ISDN, or a Remote Operations (RO) parameter for SS7.
5. The UCS DMS-250 switch establishes a connection with the IP. During an active connection to an IP, data may be exchanged between the SCP and IP using a FACILITY message between the switch and IP and a **`Call_Info_From_Resource`** or **`Call_Info_To_Resource`** message between the switch and SCP.

6 During an active connection to an IP, the SCP can request the switch to terminate the connection between a subscriber and an IP by sending a **`Cancel_Resource_Event`** message to the switch. The switch notifies the IP of the termination request using a FACILITY message. The IP responds with a DISCONNECT message and releases the call. The data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***`IPReturnBlock`*** parameter of the **`CTR_Clear`** message.

7 Once the IP completes its function, it releases the call and the UCS DMS-250 switch sends a **`CTR_Clear`** message to the SCP and awaits further instructions. When the IP releases the call, the data contained in the FIE of the ISDN DISCONNECT message or the RO parameter of the SS7 RELEASE message is passed to the SCP in the ***`IPReturnBlock`*** parameter of the **`CTR_Clear`** message.

Component and operation type background

The exchange of messages between the UCS DMS-250 switch and the IP is based upon a simple request/reply exchange. Basically, the invoker (switch or the IP) requests a service from the performer (IP or the switch). After providing the service, the performer is expected to respond with either success or failure to the request from the invoker.

Components are used for the messaging exchange. A component may consist of a request to perform an operation at the remote end. A component may also indicate the success or failure of the requested operation. An operation indicates the service which is to be provided by the performer.

The following four components are used for 1129-style IP interaction:

- Invoke: This component is send by the invoker to initiate a service at a remote end (service performed by the performer).
- Return Result: This component is sent by the performer to indicate to the invoker that the requested service was performed correctly.
- Return Error: This component is sent by the performer to indicate to the invoker that the requested service could not be performed.
- Reject: This component is sent by either the invoker or performer to reject a received component.

For the SS7 protocol, these components are placed into a Remote Operations (RO) parameter; for the PRI protocol, these components are placed into a Facility Information Element (FIE). The formats for the RO parameter and FIE are shown in Figure 3-2 and Figure 3-3 respectively.

Figure 3-2
Remote Operation (RO) parameter format

<table>
<tr><td>bit 8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>bit 1</td><td>octet</td></tr>
<tr><td colspan="8">Parameter ID = 00110010</td><td>1</td></tr>
<tr><td colspan="8">Parameter Length</td><td>2</td></tr>
<tr><td>Ext</td><td>Spare</td><td colspan="6">Protocol Profile</td><td>3</td></tr>
<tr><td colspan="8">Component</td><td>4 – n</td></tr>
</table>

Figure 3-3
Facility Information Element (FIE) format

<table>
<tr><td>bit 8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>bit 1</td><td>octet</td></tr>
<tr><td colspan="8">Facility Information Element Identifier = 00110010</td><td>1</td></tr>
<tr><td colspan="8">Length of Information Element</td><td>2</td></tr>
<tr><td>Ext</td><td>Spare</td><td colspan="6">Protocol Profile</td><td>3</td></tr>
<tr><td colspan="8">Component</td><td>4 – n</td></tr>
</table>

The four components used for exchanging messages between a UCS DMS-250 switch and an IP are briefly described in the following four sections.

Invoke Component

The Invoke component is used to initiate a service at the remote end. The Invoke component contains a parameter which identifies the operation, and any additional parameters needed by the remote end in order to perform the requested service. Two operation parameters are supported in the Invoke component:

- `sendToIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 1}.
- `cancelIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 2}.

Return Result Component

When the requested service is performed successfully, the performer sends a Return Result component to the invoker. The Return Result component may contain parameters to be returned to the invoker. The operation value parameter is included only when parameters are present in the component. One operation is supported in the Return Result component:

- `sendToIPResource` – This operation is identified by the following seven bytes – {1 3 17 105 2 1 1}.

Return Error Component

When the requested service can not be performed, the performer sends a Return Error component to the invoker. The Return Error component contains an Error Value which indicates the reason for failure. It may also contain a parameter which provides additional information regarding the error.

Reject Component

A Reject component is sent by either the invoker or performer to reject a received component (for example, Invoke). The components are rejected for such reasons as protocol violations, unrecognized components, or unrecognized parameters.

Caller interactions

The UCS DMS-250 switch procedures for processing a **`Connect_To_Resource`** message within a Conversation package are based on the procedures defined in *GR-1298-CORE*. The connection can be made to a single leg of the call, either originating or terminating, or to the entire call.

If the UCS DMS-250 switch cannot play the announcement or collect digits due to the unavailability or failure of switch hardware or switch resources, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `failure` and a ***`FailureCause`*** of `unavailableResources`.

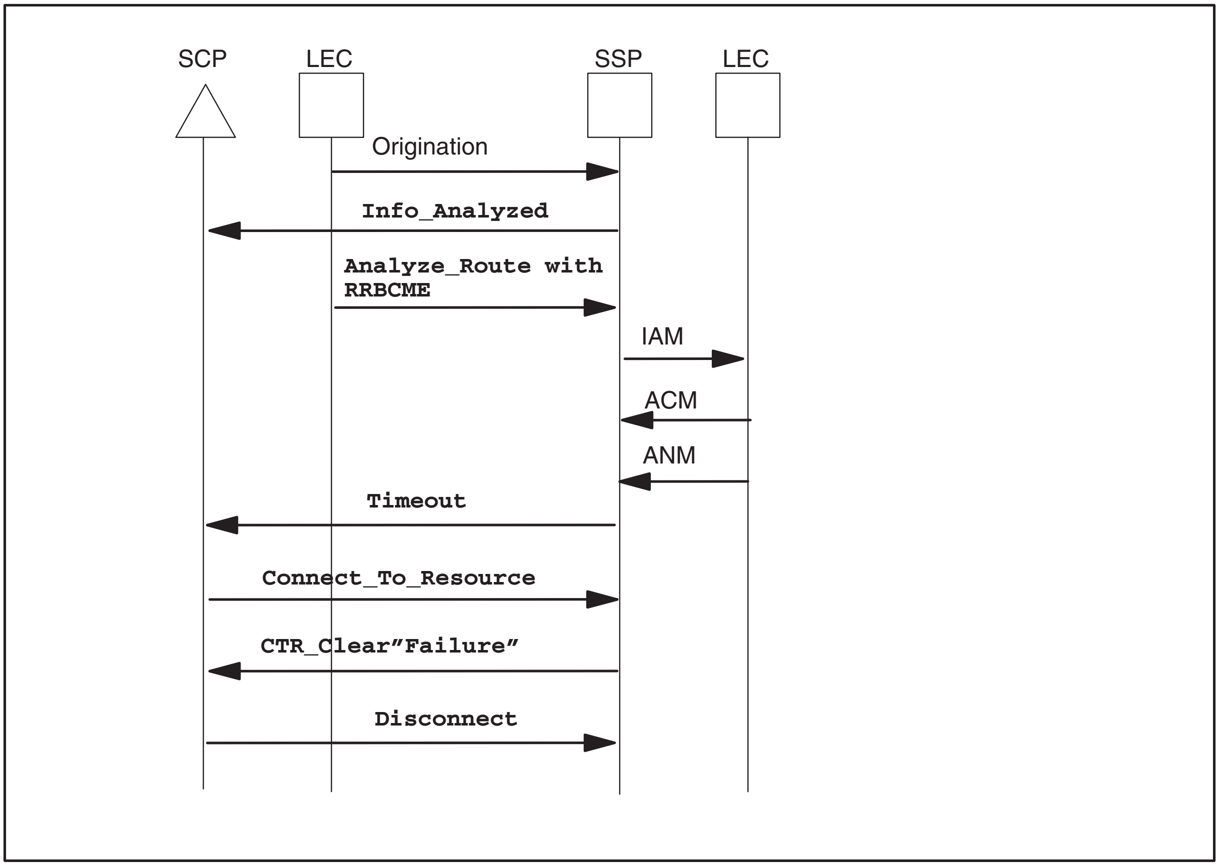
Figure 3-4 illustrates this scenario.

If the requested resource is not installed or implemented on the UCS DMS-250 switch, then it is treated by the switch as follows:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.

- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-4
Unavailability or failure of switch hardware or switch resources



IP resources

When the UCS DMS-250 switch receives a **`Connect_To_Resource`** message which is destined for an IP (that is, the message contains a ***`DestinationAddress`*** parameter) the message processing is similar to the **`Send_To_Resource`** message processing.

Determination of Intelligent Peripheral Interface (IPI)

The interaction context between the SCP and IP can be either CONNECT_ONLY or CONNECT_1129_STYLE. The method for a particular interaction can be specified by one of three ways, with an implied precedence ordering. They are as follows, given from highest to lowest precedence:

- The `strConnectionType` extension parameter in the **`Connect_To_Resource`** message.

- The STR_CONNECTION_TYPE in table CAINXDFT for the associated CAINGRP.
- The tuple STR_CONNECTION_TYPE in table CAINPARM.

If the IPI determined from the application of the above precedence ordering is determined to be NONE, then a **`CTR_Clear`** message is send to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `taskRefused`.

A **`Connect_To_Resource`** message containing the DestinationAddress parameter is used to initiate an CTR-Connection to a IP. When the message is received by the switch, the contents of the message are first validated.

If the ***`DestinationAddress`*** parameter is present in a **`Connect_To_Resource`** message received in a response package, an application error is detected. The following actions are performed:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- A CAIN200 Application Error log is generated.
- The call is not cleared toward the controlling leg or the passive leg.

When both the ***`DestinationAddress`*** and ***`DisconnectFlag`*** parameters are present in a **`Connect_To_Resource`** message, an application error is detected. The following actions are performed:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- A CAIN200 Application Error log is generated.
- The call is not cleared toward the controlling leg or the passive leg.

The contents of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters are not validated by the UCS DMS-250 switch when connecting to an IP. However, the parameter sizes are verified to ensure that the maximum size limits are not exceeded. If the maximum size is exceeded, the following actions are performed:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- The call is not cleared toward the controlling leg or the passive leg.

Establishing an CTR-Connection to a Local IP

Once a route index is identified, the local switch attempts to establish a connection to the local IP. It is important to note that existing UCS DMS-250 software is used to establish the connection. Therefore, inswitch features may interact with the CTR-Connection. Several examples are listed below:

- During the **`Authorize_Termination`** PIC, the UCS DMS-250 performs bearer capability screening on the terminating trunk using table BCCOMPAT. When bearer capability screening fails, the switch route advances to the next available trunk group in the route list.
- During the Present_Call PIC, the UCS DMS-250 constructs the ISDN SETUP message to be sent to the IP. Delivery of the Calling Party Information (CPI) element is controlled using the ANIDELV option of table CALLATTR.

Using the route list identified by the ***`DestinationAddress`*** parameter, the local switch selects a trunk group from the route list and attempts to locate an idle trunk member within the group. If no idle trunk members are present, the local switch route advances to the next available trunk group in the route list.

If the local switch is unable to locate an idle trunk member within the route list, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `channelsBusy`.
- The call is not cleared toward the controlling leg or the passive leg.

If the selected trunk group is not a PRI trunk provisioned with the IPTRUNK option, a ISUP IMT trunk provisioned with the IPTRUNK option, or a ISUP IMT (implies establishment of a Remote IP CTR-Connection) a fatal application error is detected. The following actions are performed:

- A **`CTR_Clear`** message is sent to the SCP in a conversation package with a ***`ClearCause`*** parameter value of `abort`.
- A CAIN200 Application Error log is generated.
- The call is not cleared toward the controlling leg or the passive leg.

If an idle PRI trunk member is located, the local switch constructs an ISDN SETUP message and sends it to the local IP. The following information elements are built using the data provided in the **`Connect_To_Resource`** message:

- Called Party Information (CPI) Element – This element is built using the address contained in the ***`DestinationAddress`*** parameter.

- Facility Information Element (FIE) – The FIE contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the switch.

If an idle ISUP IMT (IPTRUNK) trunk member is located, the local switch constructs an SS7 IAM message and sends it to the local IP. The following parameters are built using the data provided in the **Connect_To_Resource** message:

- Called Party Number (CPN) – This parameter is built using the address contained in the ***DestinationAddress*** parameter.
- Remote Operations (RO) – This parameter contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the local switch.

Note: For CONNECT_ONLY, the contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are discarded. On PRI terminations, an FIE is not added to the outgoing SETUP message. On ISUP IMT terminations, a RO parameter is not added to the outgoing IAM message.

Once the SETUP or IAM message is sent, the local switch waits for a response message from the local IP.

When the long call duration timer is enabled for the terminating PRI or ISUP IMT (IPTRUNK) agent, the timer is started using the value provisioned in table TRKGRP1. The long call duration timer is provided by an inswitch feature, and is similar in behavior to the T_No_Answer timer. If the long call duration timer expires before an ISDN CONNECT or SS7 ANM message is received from the local IP, the following actions are performed:

Note: The **Connect_To_Resource** is acting on an established 2-party call, therefore the long call duration feature is not allowed to tear down the previous existing call. This functionality closely mimics the timer T303 expiration.

- A **CTR_Clear** message in a response package is sent to the SCP with a ***ClearCause*** parameter value of `ipTimeout`.
- The CTR-connection is cleared.
- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-5 and Figure 3-6 illustrate this scenario for an SS7 call establishment and a PRI call establishment respectively.

Figure 3-5
SS7 Call Establishment to a Local IP using ISDN Signaling

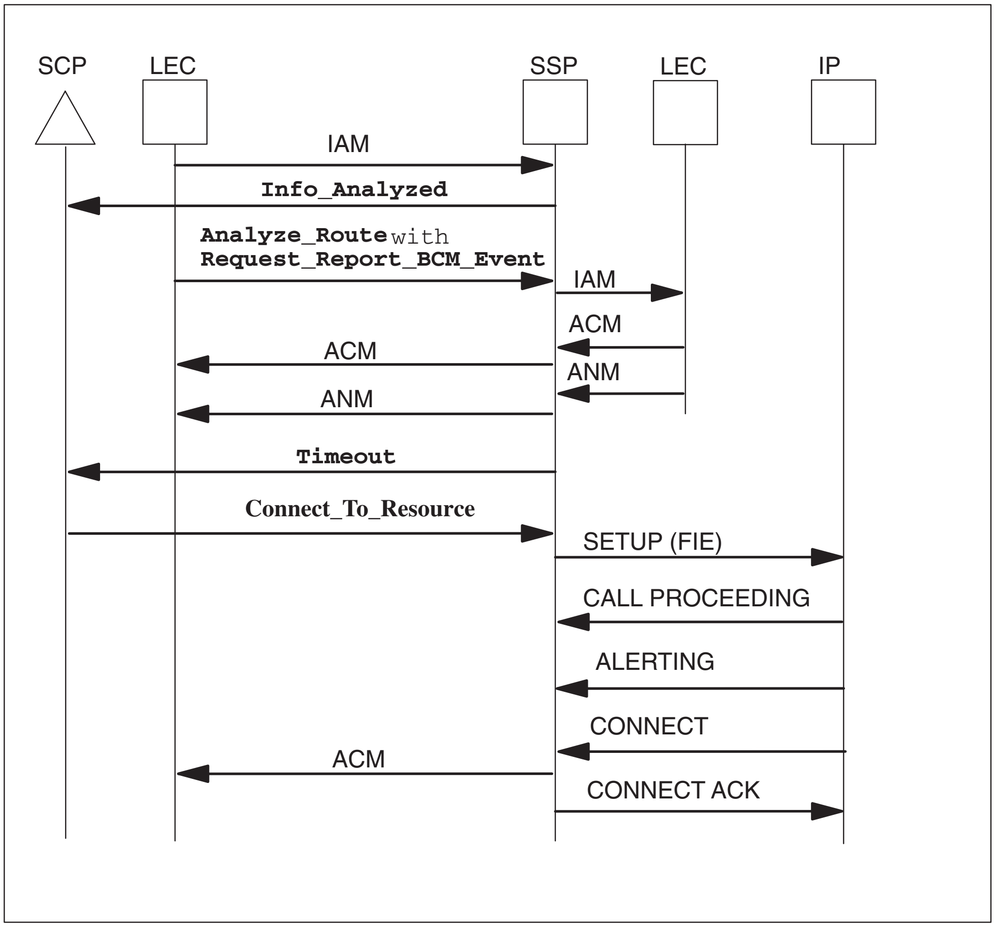
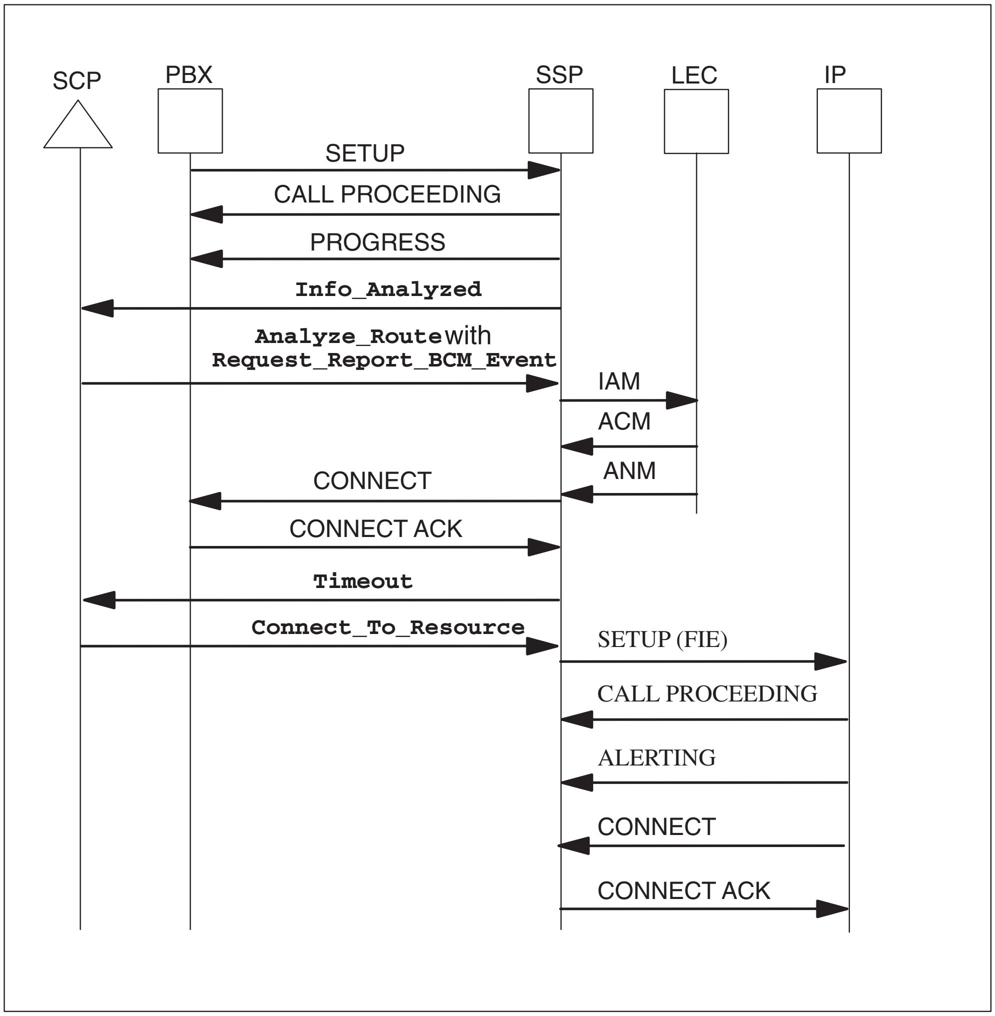


Figure 3-6
PRI Call Establishment to a Local IP using ISDN Signaling



Signaling During an Active Connection to a Local IP

An CTR-Connection is considered active when an answer indication is received from the IP. While the connection is active to a local IP with ISDN signaling, the UCS DMS-250 switch may receive a FAC message containing an FIE with a Return Result component. While the connection is active to a local IP with SS7 signaling, the UCS DMS-250 switch may receive a FAR message containing an RO parameter with a Return Result component. These messages allow the SCP and IP to exchange intermediate information during a CTR-Connection.

For CONNECT_ONLY, signaling during an active connection to exchange intermediate information is not supported. If the switch receives a FAR or FAC message, it is assumed to be a supplemental service. The following actions are performed:

- A **`CTR_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `suppServiceInvoked`.
- The existing UCS DMS-250 software processes the incoming FAR or PRI FAC message.

If there is already an outstanding **`Call_Info_To_Resource`** message for the last **`Call_Info_From_Resource`** message, the following actions are performed :

- The UCS DMS-250 switch initiates call clearing toward the IP by sending a DISCONNECT or REL message.
- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.
- The call is not cleared toward the controlling leg or the passive leg.

Upon receipt of a valid FAC or FAR message, the UCS DMS-250 switch builds a **`Call_Info_From_Resource`** message, containing the ***`IPReturnBlock`*** parameter when it is present in the Return Result component. The message is sent to the SCP.

If the SCP responds with an **`Application_Error`** or **`Failure_Report`** message, the T1 timer expires, or a fatal application error is detected, the following actions are performed:

- The UCS DMS-250 switch initiates call clearing toward the IP by sending a DISCONNECT or REL message.
- AIN Final Treatment (AINF) is provided toward the controlling leg and the passive leg.

The expected response to the **`Call_Info_From_Resource`** message is the **`Call_Info_To_Resource`** message. The message is processed differently depending upon the presence or absence of the ***`ResourceType`*** and ***`StrParameterBlock`*** parameters.

Call_Info_To_Resource without ResourceType and StrParameterBlock parameters

- In this scenario, the SCP has determined that the IP is no longer needed for the call. The following actions are performed:
  - — The UCS DMS-250 switch initiates call clearing toward the IP by sending a DISCONNECT or REL message.

— A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`.

Figure 3-7 and Figure 3-8 illustrate this scenario for PRI and SS7 respectively.

Figure 3-7
Call_Info_To_Resource message without parameters - PRI

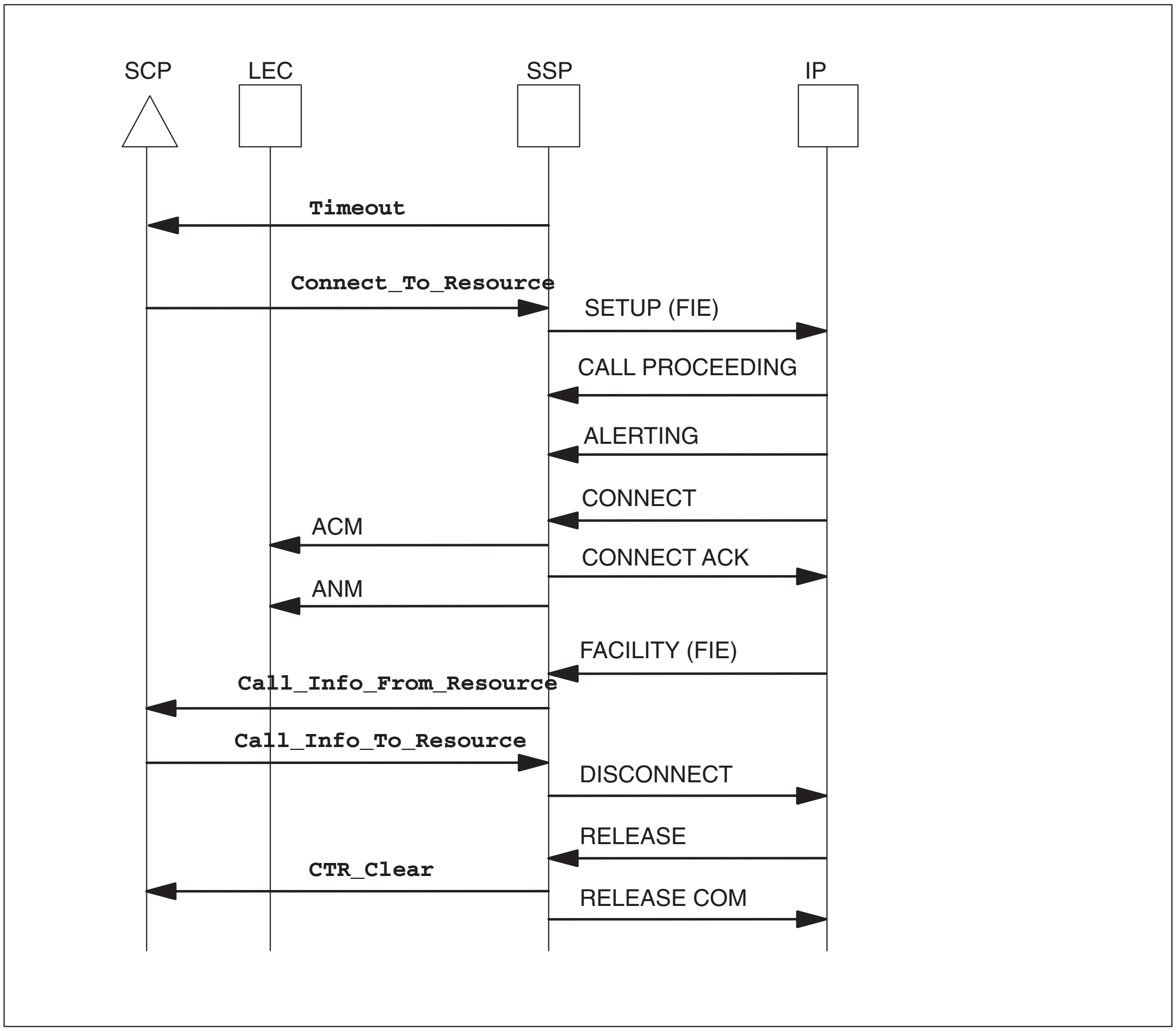
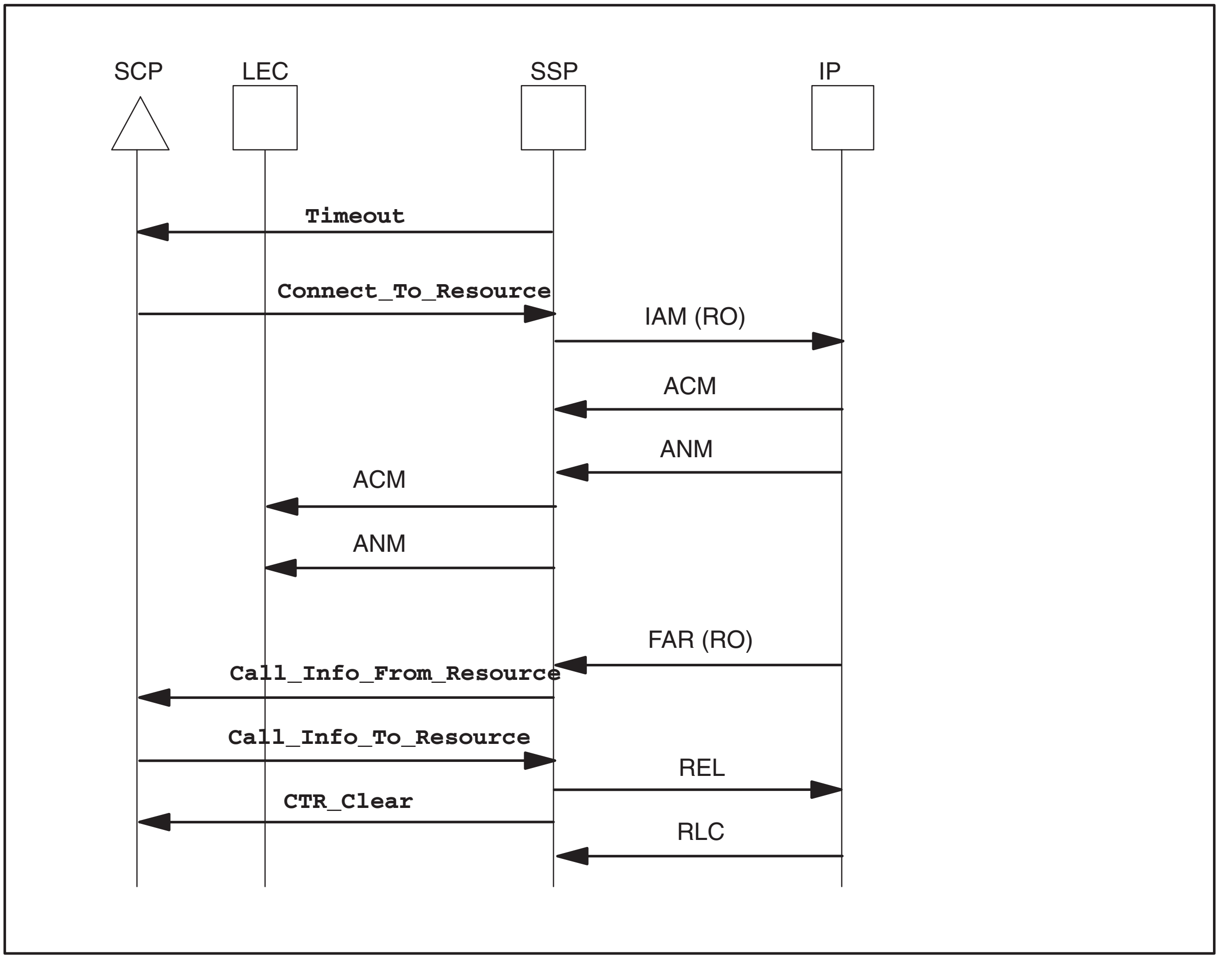


Figure 3-8
Call_Info_To_Resource message without parameters - SS7



Call_Info_To_Resource without ResourceType and StrParameterBlock parameters

— In this scenario, the SCP has determined that additional information needs to be passed to the IP. The following actions are performed:

– A PRI FAC message with an FIE or a SS7 FAR message with an RO parameter is sent to the IP. The FIE or RO contains an Invoke component with an operation of `sendToIPResource`. The contents of the ***ResourceType*** and ***StrParameterBlock*** parameters are placed into the Invoke component without modification by the UCS DMS-250 switch.

Figure 3-9 and Figure 3-10 illustrate this scenario for PRI and SS7 respectively.

Note: `connectToIPResource` operation is not yet supported by PRI based IPs. Therefore `sendToIPResource` operation value is used.

Figure 3-9
Call_Info_To_Resource message with parameters - PRI

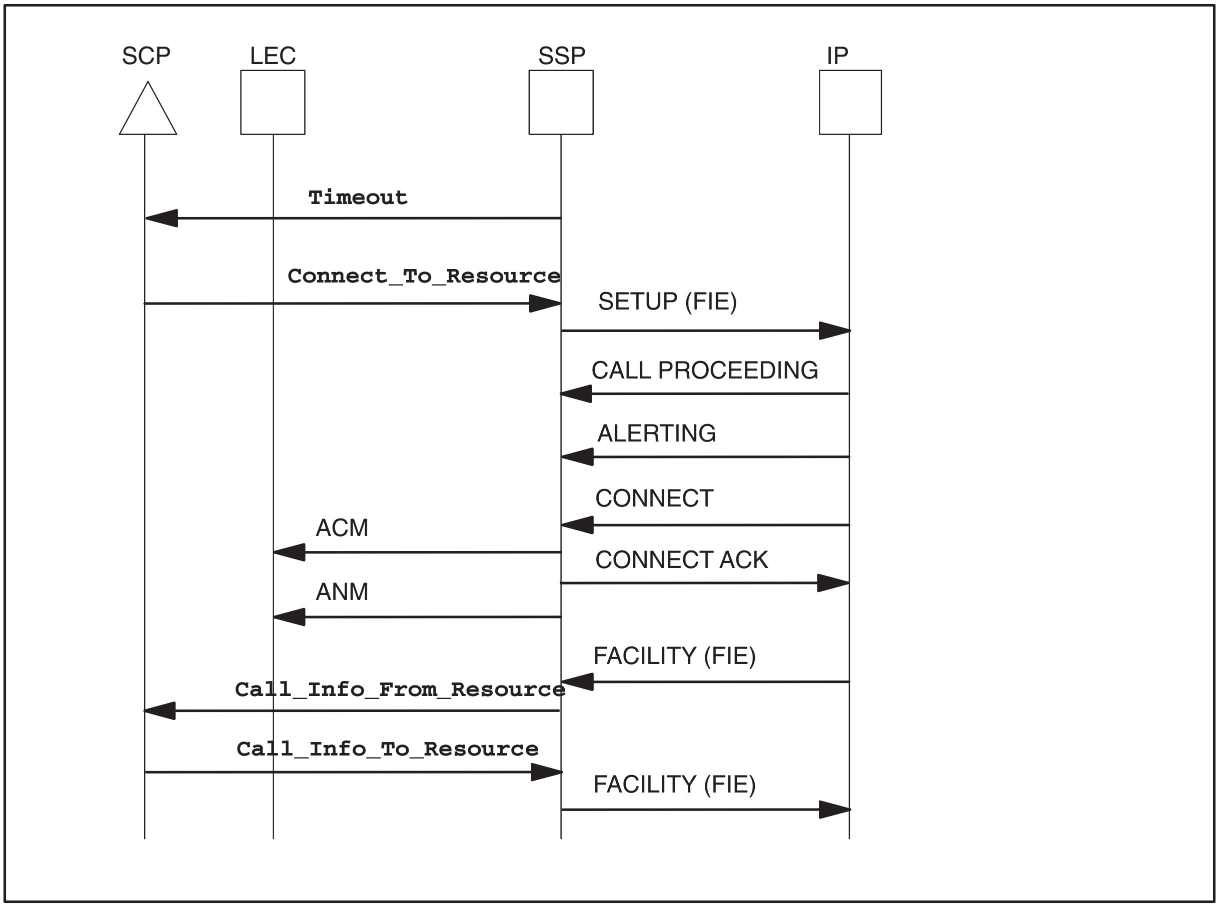
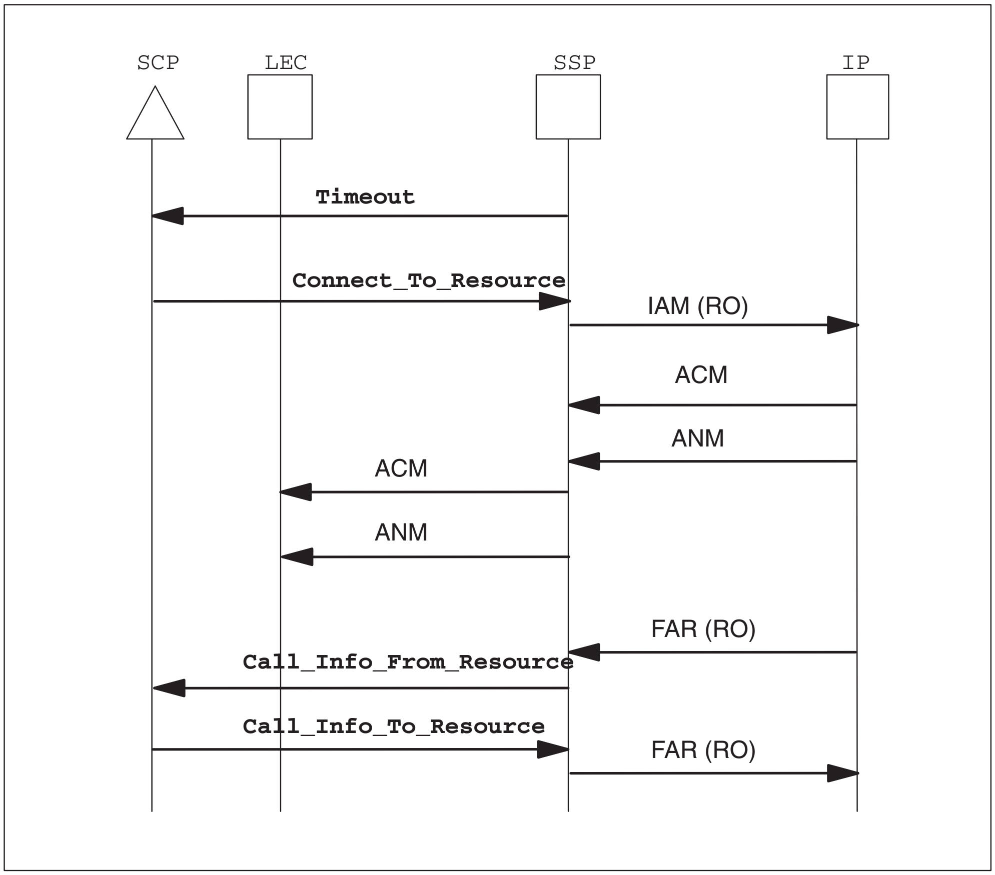


Figure 3-10
Call_Info_To_Resource message with parameters - SS7



- Timer TSTRC is canceled.

During an active connection, the UCS DMS-250 switch may receive a PRI FAC with a FIE, a FAC with an RO, or a FAR message with a facility indicator indicating a service. This occurs when the IP is requesting a supplemental service on the UCS DMS-250 switch such as Release Link Trunk (RLT) or Billing Information.

Release Link Trunk (RLT)

The RLT supplemental service is not supported during a CTR-Connection. If the IP requests such a service the UCS DMS-250 switch performs the following:

- A FRJ or FAC message is sent to the IP indicating that the requested operation is rejected by the switch.
- The call is not cleared toward the controlling leg or the passive leg.

- The CTR-Connection is maintained.

Refer to *UCS DMS-250 SS7 RLT Feature Application Guide* and UCS *DMS-250 PRI RLT Feature Application Guide* for RLT feature functionality and message protocol details.

Billing Information

The Billing Information supplemental service allows an IP using ISDN signaling to provide billing information to update the Call Detail Record (CDR) fields BILLNUM, ACCTCD, and PINDIGS. This supplemental is also disallowed during a CTR-Connection.

IP-Initiated Clearing of a Connection to a Local IP

Once the SETUP or IAM message has been sent, the local IP may respond with a message that clears the CTR-Connection. The following two sections describe the messages which clear the CTR-Connection.

Disconnect/Release Message

When initiating normal call clearing, the local IP sends a DISCONNECT message or Release (REL) message with a Cause Indicator indicating normal clearing. When the DISCONNECT or REL message contains a Return Result component, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-11 and Figure 3-12 illustrate this scenario for PRI and SS7 respectively.

For CONNECT_ONLY, the exchange of data using the DISCONNECT or REL message is not supported. When either message is received, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`. The Return Result component is ignored if it is present in the incoming DISCONNECT or REL message.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-11
IP initiated clearing with a Return Result component - PRI

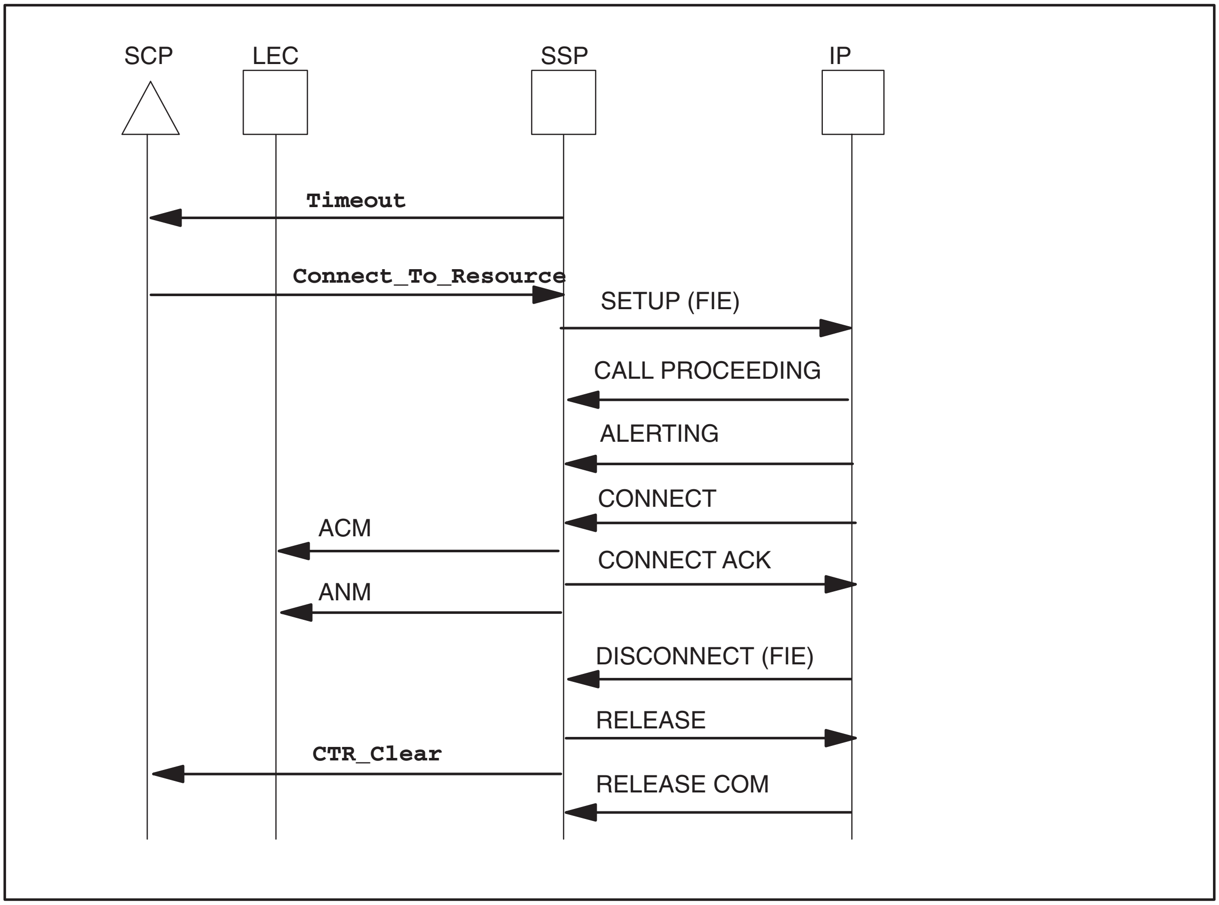
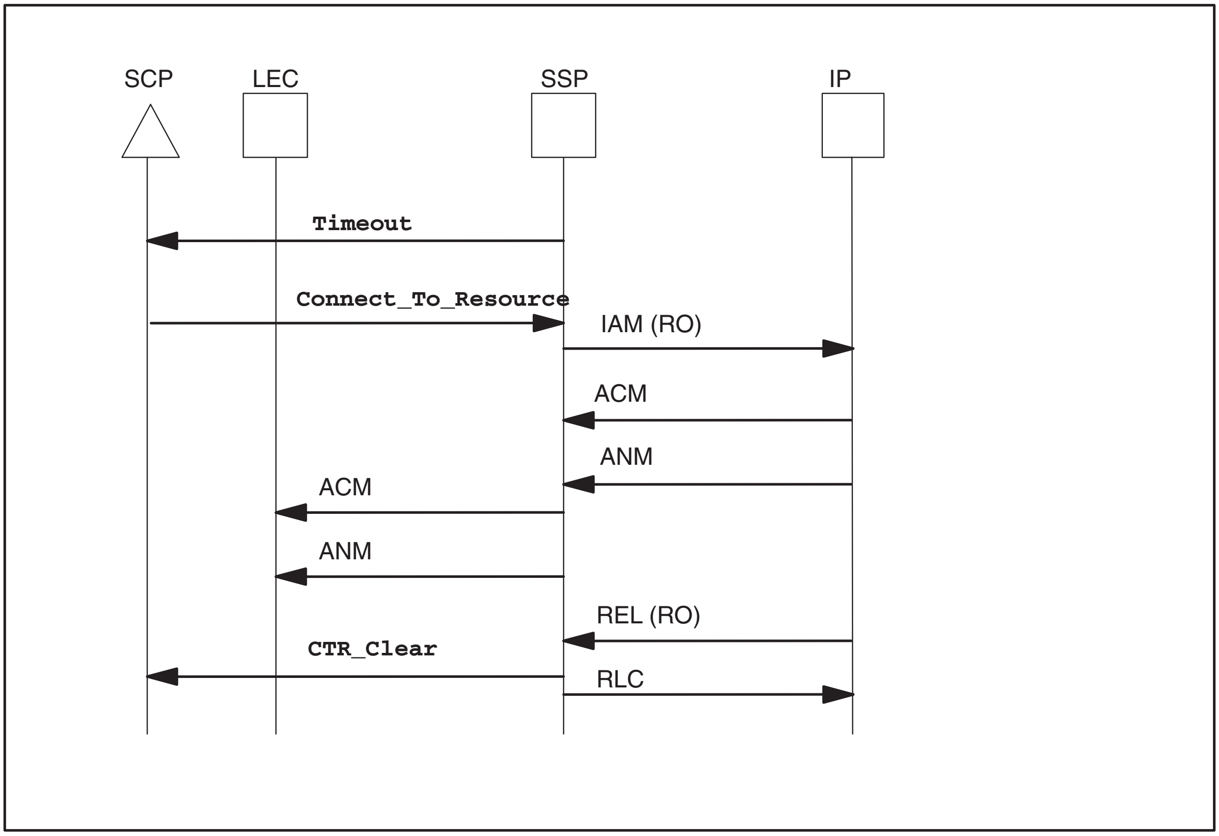


Figure 3-12
IP Initiated Clearing with a Return Result Component - SS7



When the DISCONNECT or REL message contains a Return Error component, the following actions are performed:

- A **CTR_Clear** message in a conversation package is sent to the SCP. The value of the ***ClearCause*** parameter is based upon the contents of the Error Value field of the Return Error component. Volume 3, Chapter 12, “Incoming CAIN message parameters,” provides a mapping of the Error Values to ***ClearCause*** parameter values. The ***ClearCauseData*** parameter is included in the **CTR_Clear** message when the error parameter is present in the Return Error component.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

When the DISCONNECT or REL message contains a Reject component, the following actions are performed:

- A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `protocolError`.
- The CTR-Connection is cleared and timer TSTRC is canceled.

- The call is not cleared toward the controlling leg or the passive leg.

If the UCS DMS-250 switch receives a DISCONNECT or REL message without a component, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

For CONNECT_ONLY, a DISCONNECT or REL message without a component is expected for this IPI. As stated earlier in this section, a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `normal`.

Release Complete/Release Message

When initiating abnormal call clearing, the local IP may send a RELEASE COMPLETE message or Release (REL) message with a Cause Indicator indicating abnormal clearing. When the message contains a Reject component, the following actions are performed:

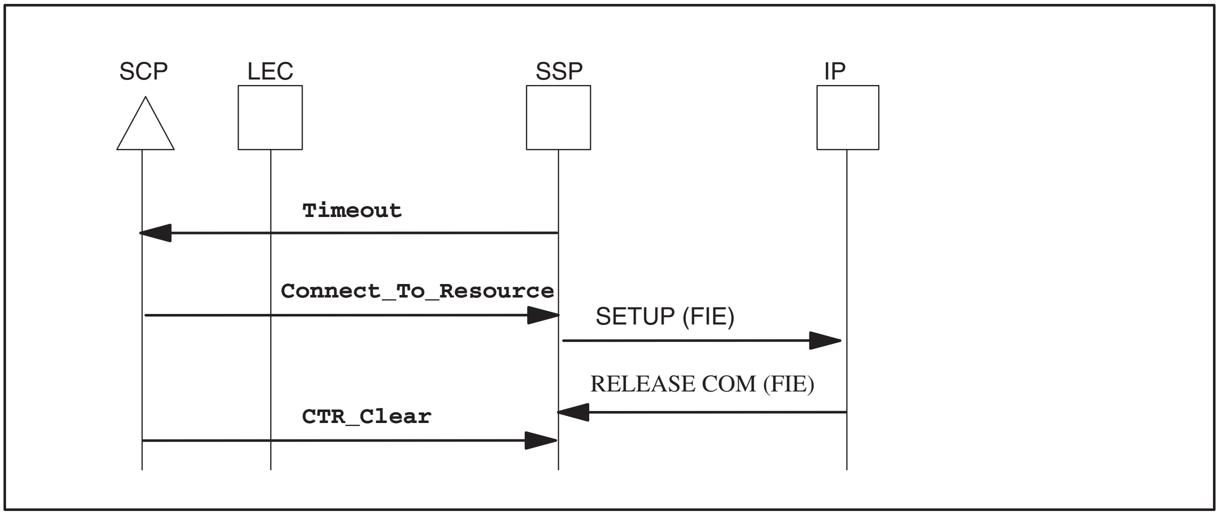
- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-13 illustrates this scenario.

For CONNECT_ONLY, the exchange of data using the RELEASE COMPLETE or REL message is not supported. When either message is received, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`. The Reject component is ignored if it is present in the incoming RELEASE COMPLETE or REL message.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

Figure 3-13
IP Initiated clearing with a Reject component



If the RELEASE COMPLETE or REL message does not contain a component, the following actions are performed:

- A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of `abort`.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

FAR or FAC Message

When initiating abnormal call clearing during an active connection, the local IP may send a PRI FAC or SS7 FAR message. When the FAC or FAR message contains a Return Error component, the following actions are performed:

- A **CTR_Clear** message in a conversation package is sent to the SCP. The value of the ***ClearCause*** parameter is based upon the contents of the Error Value field of the Return Error component. Volume 3, Chapter 12, “Incoming CAIN message parameters,” provides a mapping of the Error Values to ***ClearCause*** parameter values. The ***ClearCauseData*** parameter is included in the **CTR_Clear** message when the error parameter is present in the Return Error component.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

When the FAC or FAR message contains a Reject component, the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `protocolError`.
- The CTR-Connection is cleared and timer TSTRC is canceled.
- The call is not cleared toward the controlling leg or the passive leg.

For CONNECT_ONLY, the exchange of data using the FAR or FAC message is not supported.

Switch-initiated Clearing of a Connection to a Local IP

The CTR-Connection to the local IP is cleared by the local switch when the controlling leg or passive leg specified in the ***`LegID`*** parameter abandons, when the calling leg specified in the ***`LegID`*** parameter abandons, or when timer TSTRC expires. These scenarios are described in the following sections.

Caller abandon

The controlling leg or passive leg specified in the ***`LegID`*** parameter may decide to abandon the call during a CTR-Connection to the local IP. When this occurs before the CTR-Connection is active (before the IP has answered), the following actions are performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.

Figure 3-14 and Figure 3-15 illustrate this scenario for PRI and SS7 respectively.

Figure 3-14
Caller abandon before an Active CTR-Connection - PRI

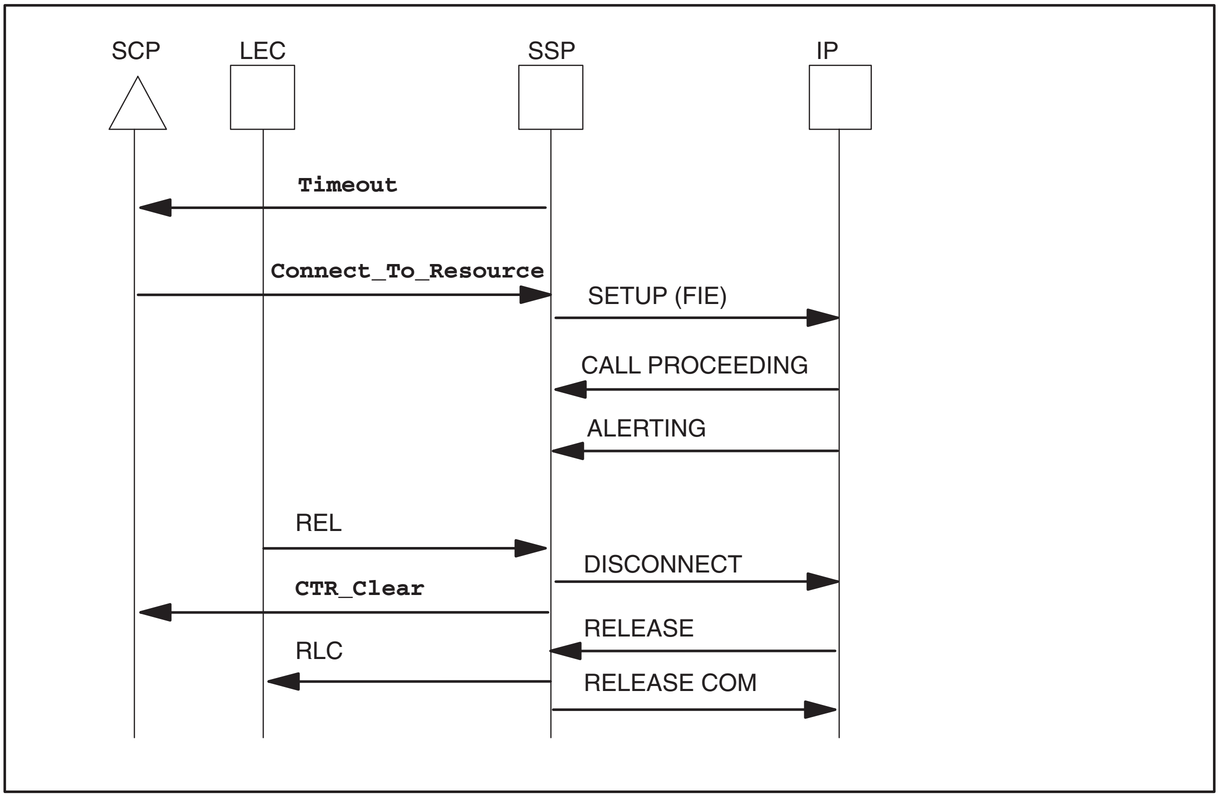
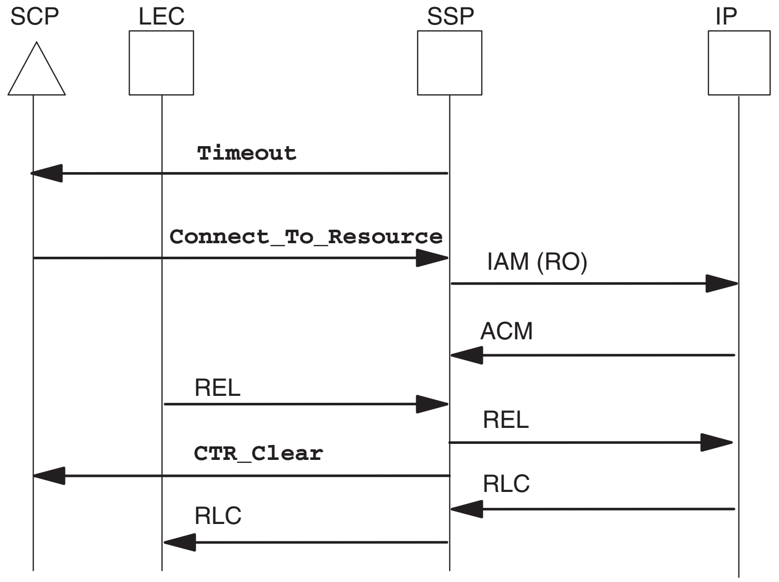


Figure 3-15
Caller abandon before an Active CTR-Connection - SS7



When the controlling leg or passive leg specified in the ***LegID*** parameter abandons during an active CTR-Connection, the following actions are performed:

- A FAC or FAR message containing an Invoke component with an operation of cancelIPResource is sent to the IP.
- The IP Disconnect Timer (TDISC) is started. This timer specifies the maximum time in seconds in which an IP must respond to a FAC or FAR message with the cancelIPResource operation.
- Timer TSTRC is canceled.

For CONNECT_ONLY, since the exchange of data is not supported, the following actions are performed upon caller abandon during an active connection:

- A CTR_Clear message in a response package is sent to the SCP with a ClearCause parameter value of userAbandon.
- The CTR-Connection is cleared and timer TSTRC is canceled.

It is possible that the caller abandoned while the SCP was waiting for a **Call_Info_To_Resource** message. If the message is received after the FAC or FAR message with the cancelIPResource operation was sent to the IP, the T1 timer is canceled and the **Call_Info_To_Resource** message is discarded .

The IP is expected to respond to the `cancelIPResource` operation with a DISCONNECT or REL message normally containing a Return Result component.

- If the T1 timer is not running, the following actions are performed:
  - A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`. The IPReturnBlock is included in the **`CTR_Clear`** message when it is present in the Return Result component.
  - The CTR-Connection is cleared and timer TDISC is canceled.
  - The call is not cleared toward the controlling leg or the passive leg.
- If the T1 timer is running, the following actions are performed.
  - The CTR-Connection is cleared and timer TDISC is canceled.
  - The local switch awaits the receipt of the **`Call_Info_To_Resource`** message from the SCP. A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`. The **`Call_Info_To_Resource`** is discarded.
- If the T1 timer expires, a fatal application error is detected.

Figure 3-16 illustrates this scenario without the T1 timer. Figure 3-17 illustrates this scenario with the T1 timer running.

Figure 3-16
Caller abandon during an Active CTR-Connection

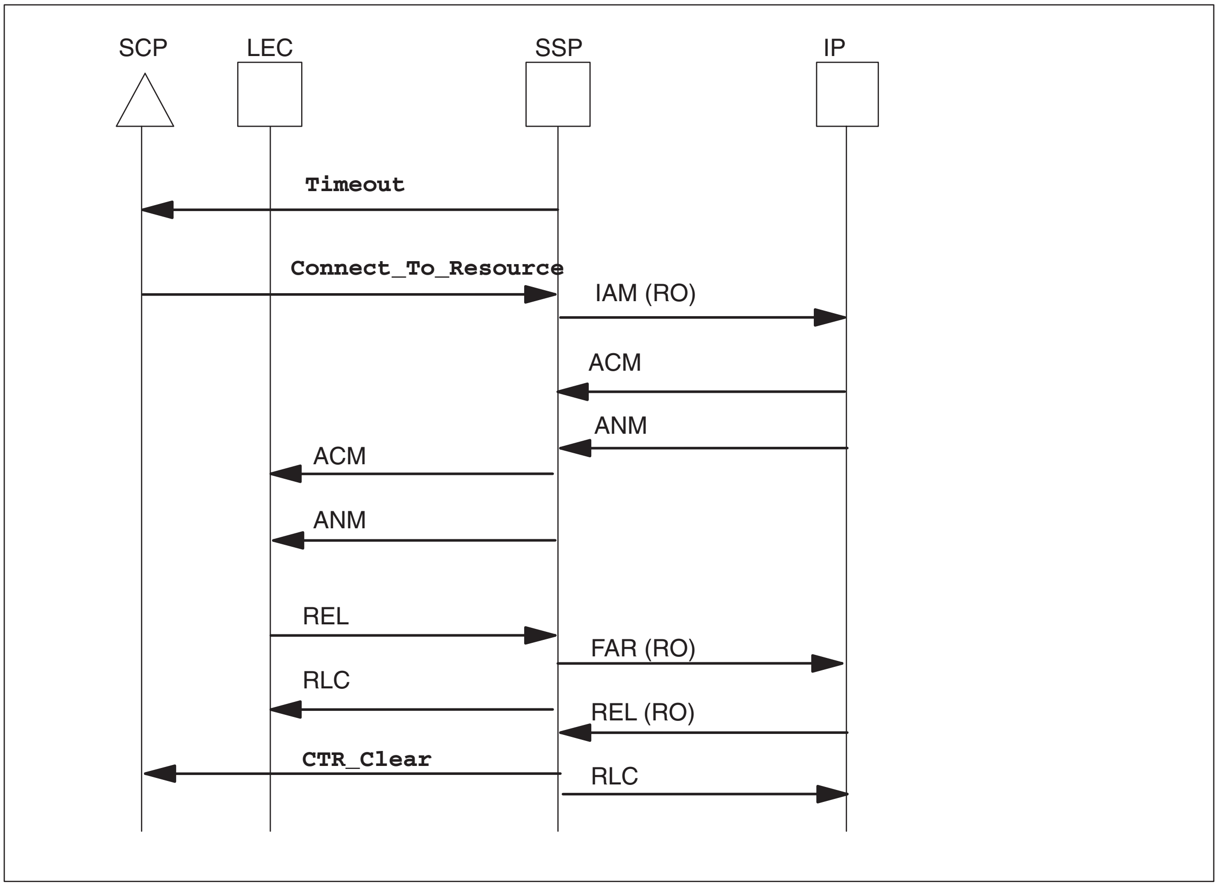
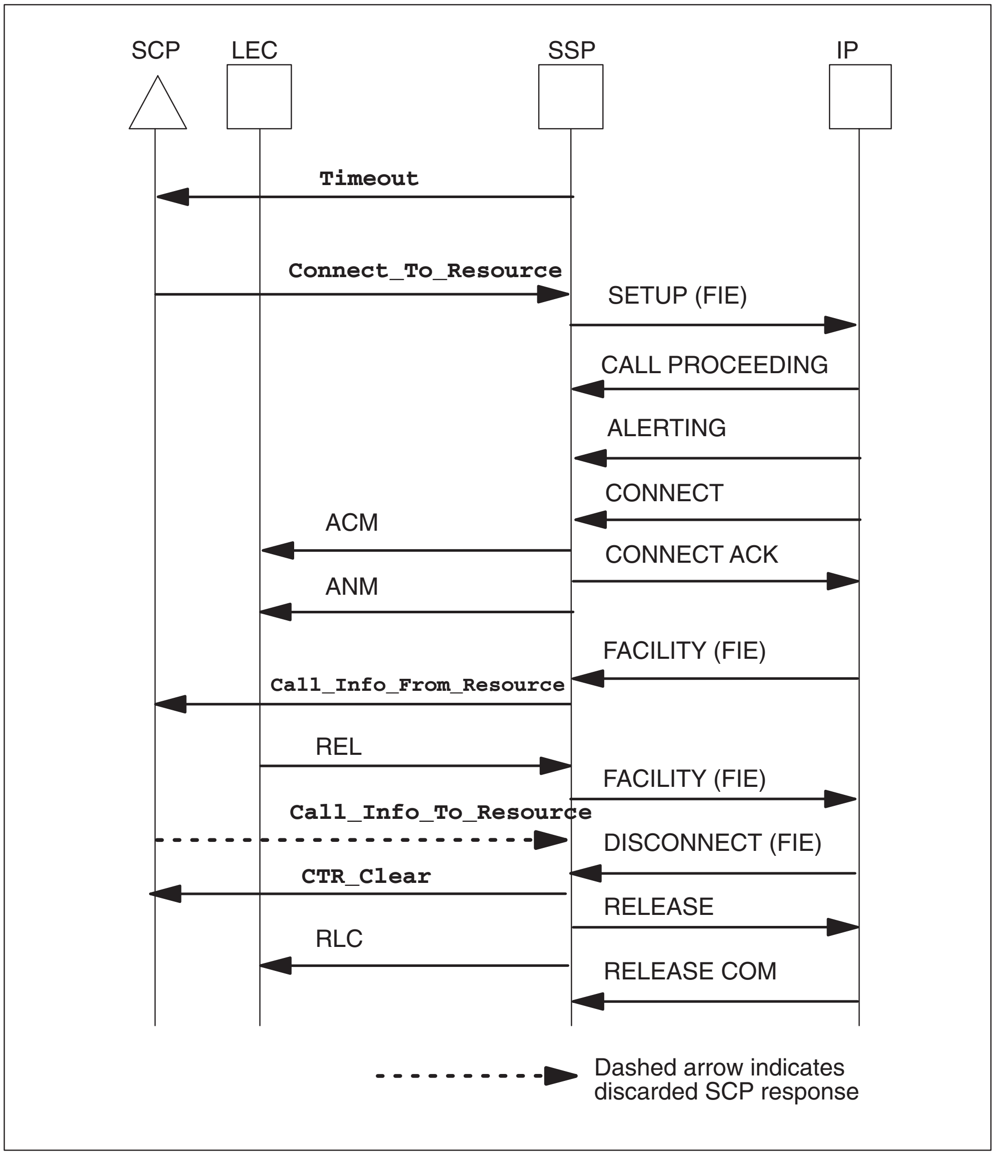


Figure 3-17
Caller abandon during an Active CTR-Connection with T1 timer running



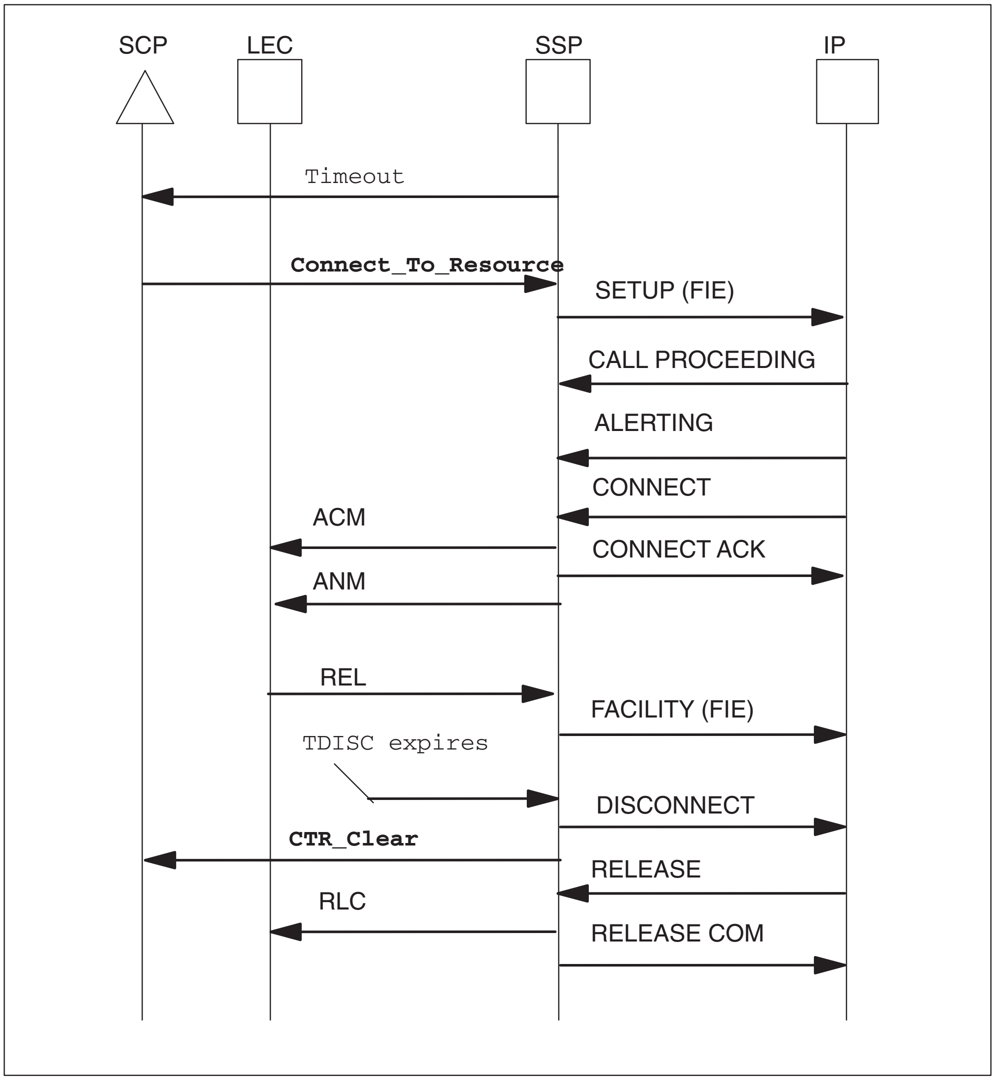
If timer TDISC expires before the IP responds to the FAC or FAR message with the `cancelIPResource` operation, the following actions are performed:

- If timer T1 is not running, the following actions are performed:
  - — A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
  - — The switch clears the CTR-Connection.
  - — The call is not cleared toward the controlling leg or the passive leg.

- If timer T1 is running, the following actions are performed:
  - — The switch clears the CTR-Connection.
  - — The local switch awaits the receipt of the **`Call_Info_To_Resource`** message from the SCP. When the message is received, it is discarded. A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
  - — The call is not cleared toward the controlling leg or the passive leg.
- If the T1 timer expires, a fatal application error is detected.

Figure 3-18 illustrates the scenario when the TDISC timer expires and the T1 timer is not running.

Figure 3-18
Timer TDISC Expires During Caller abandon, T1 timer not running



Timer TSTRC

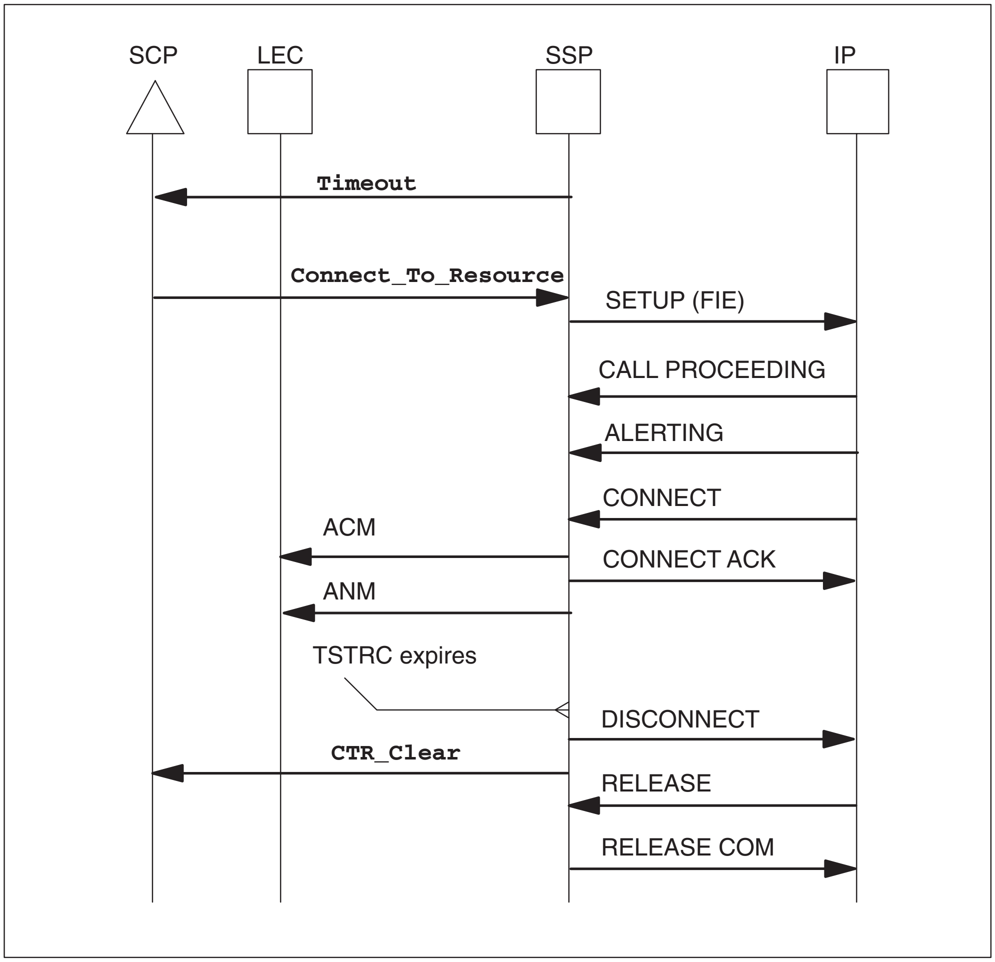
Timer TSTRC provides a maximum time limit for an CTR-Connection to an IP. It is started when the IP answers and canceled when the CTR-Connection is cleared, either by the switch or IP. If timer TSTRC expires, the following actions are performed:

- When the T1 timer is not running, the following actions are performed:
  - — A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
  - — The call is not cleared toward the controlling leg or the passive leg.
- When the T1 timer is running, the following actions are performed:

- The UCS DMS-250 switch awaits the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The call is not cleared toward the calling user.
- If the T1 timer expires before the SCP response is received, a fatal application error is detected.

Figure 3-19 illustrates the scenario when the TSTRC timer expires and the T1 timer is not running.

Figure 3-19
Timer TSTRC Expires During an CTR-Connection, T1 Timer Not Running



Fatal Application Errors

If the UCS DMS-250 switch receives a response from the SCP which is not supported, a fatal application error is detected. The receipt of an **`Analyze_Route`** or a **`Collect_Information`** response are not supported in response to a **`CTR_Clear`** message. The following actions are performed:

- A CAIN200 Fatal Application Error log is generated and final treatment is provided.
- An **`Application_Error`** message is reported to the SCP. The ***`ErrorCause`*** parameter is set to `dataError`.

Message Flows to Support the Remote IP

The following sections describe the message flow for managing a CTR-Connection at the local, intermediate, and remote switches.

Triggers at the Local, Intermediate, and Remote switches

As stated earlier, a CTR-Connection to an IP may be modeled as creating a new instance of the terminating call model. When connecting to a remote switch, the terminating call model triggers are supported at both the local and remote switchs. Triggering is not supported on intermediate switches.

When a terminating trigger is encountered at the local switch, the trigger is processed in the same manner as when connection to a local IP.

For CONNECT_ONLY, since the exchange of data is not supported, the remote switch is unable to determine that a call is involved in a CTR-Connection to an IP. Therefore, the remote switch processes the terminating trigger as it would on a normal call.

Terminating triggers at the remote switch are processed as follows:

- If the trigger action is QUERY and an CTR-canceling (**`Send_To_Resource`** is a CTR-canceling response) response is received from the SCP, the following actions are performed:
  - — A Call Progress (CPG) message containing an RO parameter is sent to the local switch. The message contains a Return Error component with an Error Code of `ctrCancelled`.
- If the trigger action is QUERY and a **`Disconnect`** message (CTR-canceling) is received from the SCP, the following actions are performed:
  - — A REL message containing an RO parameter is sent to the local switch. The message contains a Return Error component with an Error Code of `ctrCancelled`.

- If the trigger action is QUERY and a non-canceling response is received from the SCP, the CTR-Connection is maintained and the message is processed. A message is not sent to the local switch.
- If the trigger action is either IGNORE, LEAVE_TDP, or CONT_NOTRIG, the CTR-Connection is maintained and the trigger action is processed.
- If the trigger action is BLOCK, the following actions are performed:
  - — A REL message containing an RO parameter is sent to the local switch. The message contains a Return Error component with an Error Code of `ctrCancelled`.

Call Establishment at the Local switch

A **`Connect_To_Resource`** message containing the ***`DestinationAddress`*** parameter is used to initiate an CTR-Connection to an IP. The message is processed according to the rules previously described in this document.

If the **`Connect_To_Resource`** message is valid, then the switch attempts to route the leg or call to the IP as described for **`Send_To_Resource`**.

If the local switch is unable to locate an idle trunk member within the route list, the following action is performed:

- A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.

When the long call duration timer is enabled for the terminating IMT agent, the timer is started using the value provisioned in table TRKGRP1. The long call duration timer is provided by an inswitch feature, and is similar in behavior to the T_No_Answer timer. If the long call duration timer expires before an Answer Message (ANM) is received by the local switch, the following actions are performed:

If the long call duration timer expires before an SS7 ANM message is received by the local switch, the following actions are performed:

Note: The **`Connect_To_Resource`** is acting on an established 2-party call, therefore the long call duration feature will not be allowed to tear down the previous existing call. This function closely mimics the timer T303 expiration.

- A **`CTR_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
- The CTR-connection is cleared.
- The call is not cleared toward the controlling leg or the passive leg.

Call Establishment at the Intermediate switch

Existing UCS DMS-250 software establishes the call at the intermediate switch . The intermediate switch:

- processes the incoming IAM containing the RO parameter.
- translates the Called Party parameter and identifies a route index.
- elects a terminating trunk group and identifies an idle trunk member.
- constructs an outgoing IAM containing the unmodified RO parameter and sends it to the Remote switch.

For CONNECT_ONLY, the RO parameter is not present in the incoming IAM message received by the intermediate switch.

Call Establishment at the Remote switch

The remote switch uses the existing UCS DMS-250 software to process the incoming IAM containing the RO parameter. The remote switch translates the address contained in the Called Party parameter and identifies a route index.

Signaling During an Active Connection to a Remote IP

An CTR-Connection to an IP is considered active when an answer indication is received from the remote IP. During an active connection, the FAC and FAR messages allow the SCP and IP to exchange intermediate information.

Remote IP-Initiated Clearing of a CTR-Connection

Once the SETUP or IAM message has been sent, the remote IP may respond with a message that clears the CTR-Connection. The following sections describe the messages which clear the CTR-Connection.

Switch-initiated Clearing of a Connection to a Remote IP

The CTR-Connection to the local IP is cleared by the switch when the controlling leg or passive leg specified in the ***LegID*** parameter abandons, when the calling leg specified in the ***LegID*** parameter abandons, or when timer TSTRC expires. These scenarios are described in the following sections.

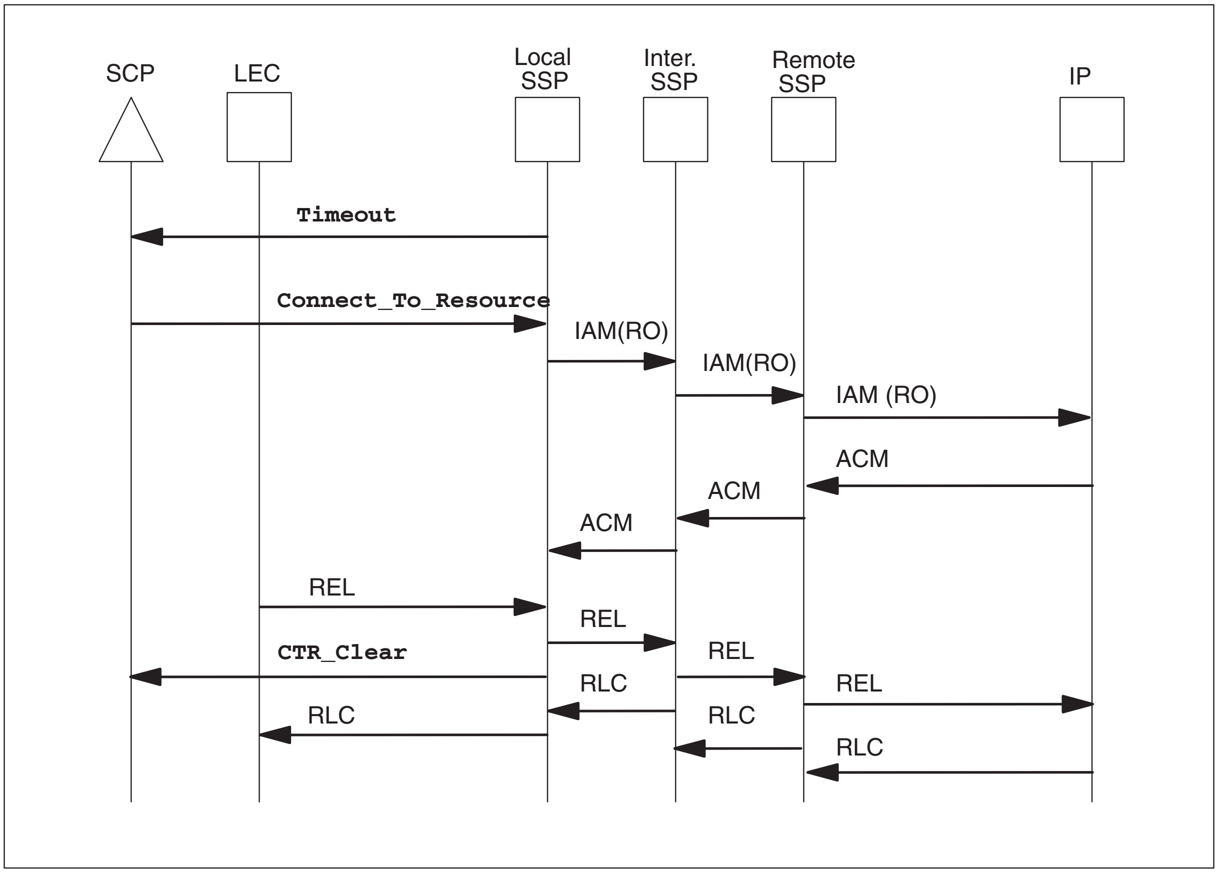
Caller abandon

The controlling leg or passive leg specified in the ***LegID*** parameter may decide to abandon the call during a CTR-Connection to the local IP. When this occurs before the CTR-Connection is active (before the IP has answered), the following action is performed:

- A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of userAbandon.

Figure 3-20 illustrates this scenario.

Figure 3-20
Caller abandon before an Active CTR-Connection to remote IP



When the controlling leg or passive leg specified in the ***LegID*** parameter abandons during an active IP CTR-Connection, the following actions are performed:

- At the local switch, the following actions are performed:
  - A FAR or FAC message containing an Invoke component with an operation of `cancelIPResource` is sent to the remote switch.
- At the intermediate switch, the FAR or FAC message is passed without modification to the remote switch.
- At the remote switch, the following actions are performed:
  - A FAR or FAC message is sent to the remote IP containing the component information received in the incoming FAR or FAC message.
  - Timer TSTRC is canceled.

- — Timer TDISC is started. This timer specifies the maximum time in seconds in which an IP must respond to a FAR or FAC message with the `cancelIPResource` operation.

For CONNECT_ONLY, since the exchange of data is not supported, the following actions are performed by the local switch upon caller abandon during an active connection:

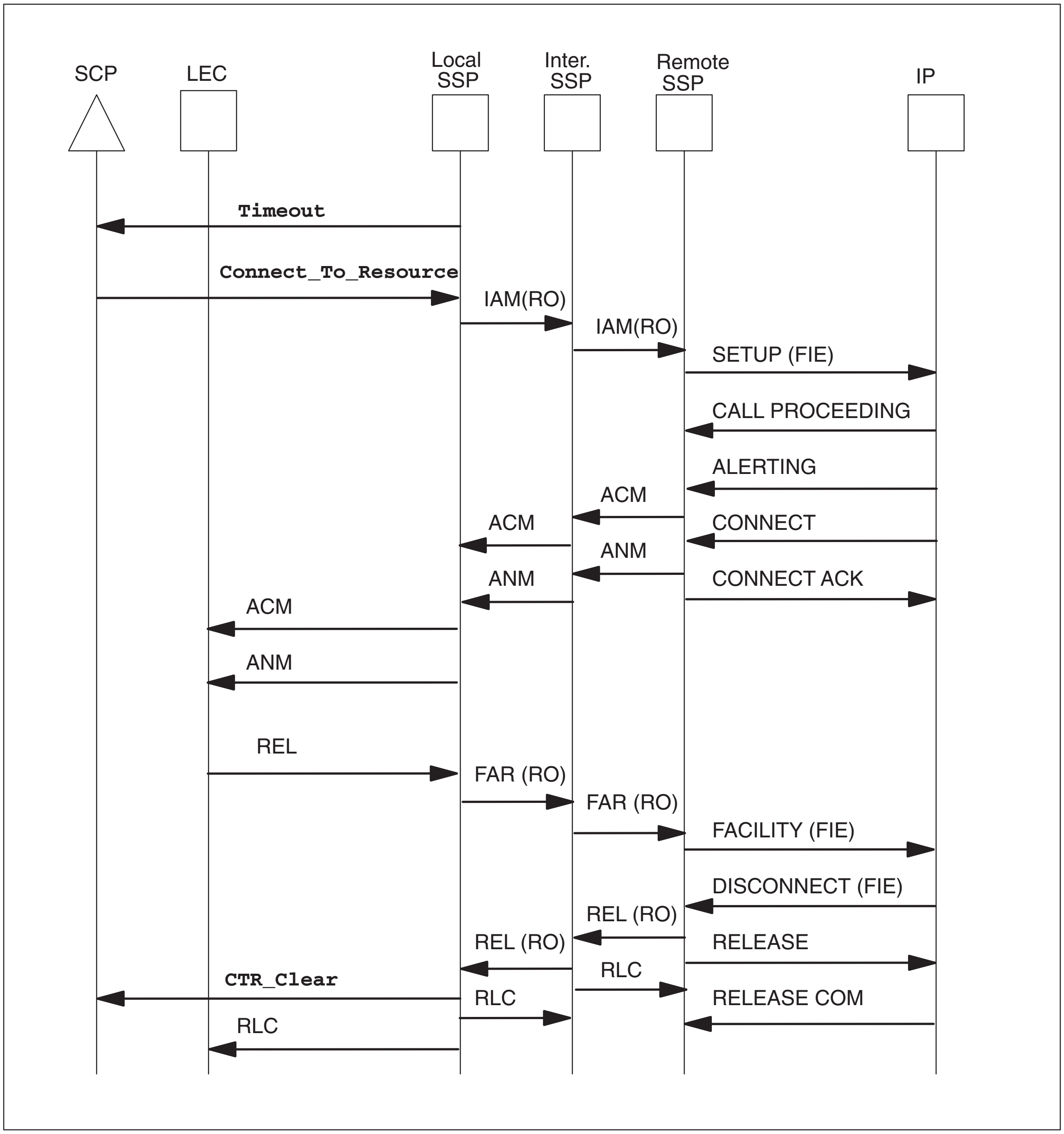
- A **`CTR_Clear`** message in a response package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
- The CTR-Connection is cleared.

The remote IP is expected to respond to the `cancelIPResource` operation with a DISCONNECT or REL message normally containing a Return Result component. The following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate then the local switch. The RO parameter contains the information that was present in the Return Result component received from the remote IP. Timer TDISC is canceled.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - — When the T1 timer is not running, the following actions are performed:
    - – A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`. The ***`IPReturnBlock`*** parameter is included in the message when it is present in the Return Result component.
    - – The CTR-Connection is cleared.
  - — When the T1 timer is running, the following actions are performed:
    - – The switch awaits the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - – The CTR-Connection is cleared.
  - — When the T1 timer expires before the SCP response is received, a fatal application error is detected.

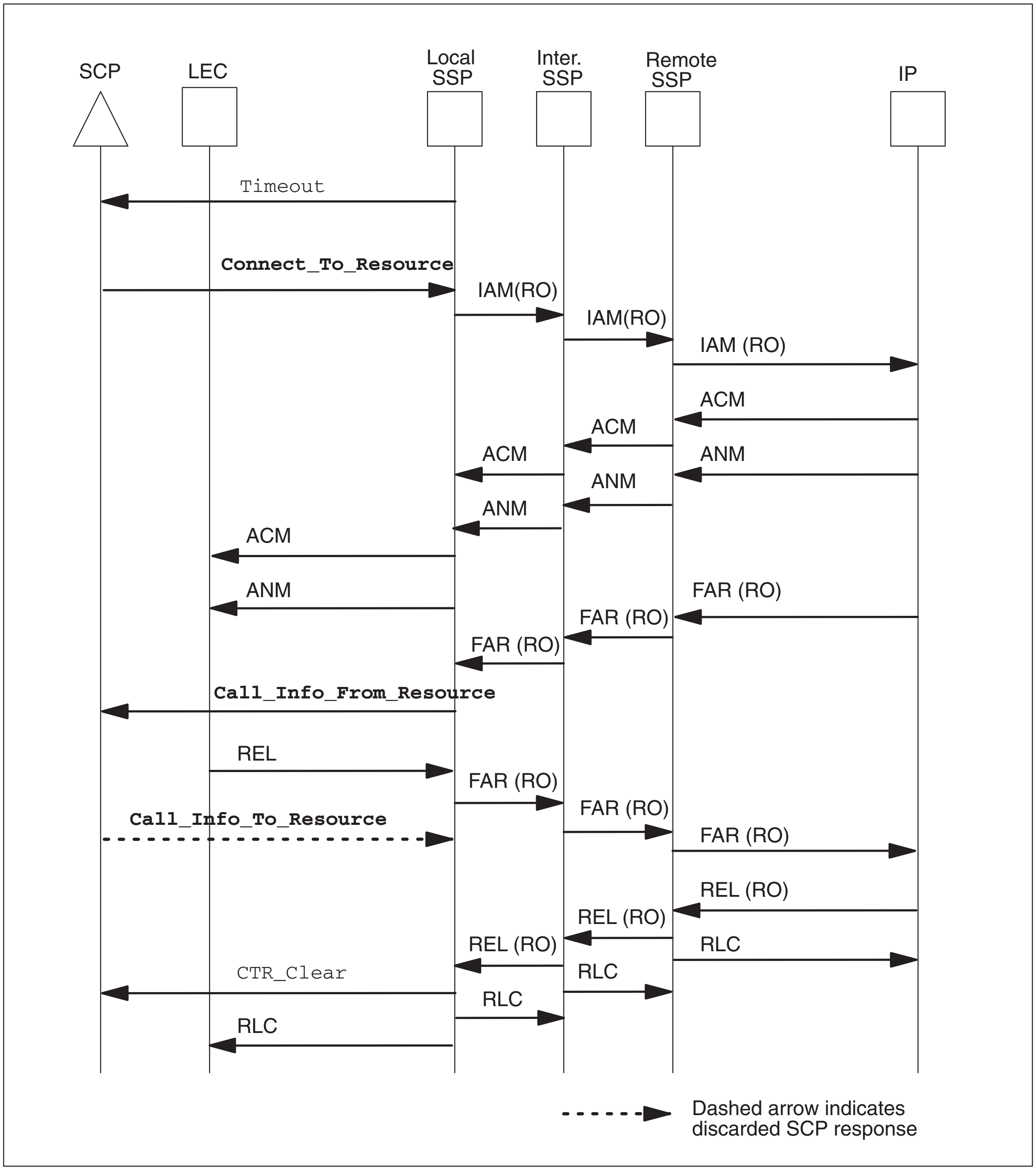
Figure 3-21 illustrates this scenario.

Figure 3-21
Caller abandon During Active IP CTR-Connection, T1 Timer not Running



It is possible that the caller abandoned while the switch was waiting for a `Call_Info_To_Resource` message. If the message is received after the FAR or FAC message with the `cancelIPResource` operation was sent to the IP, the T1 timer is canceled and the `Call_Info_To_Resource` message is discarded. Figure 3-22 illustrates this scenario.

Figure 3-22
Call_Info_To_Resource message discarded during caller abandon



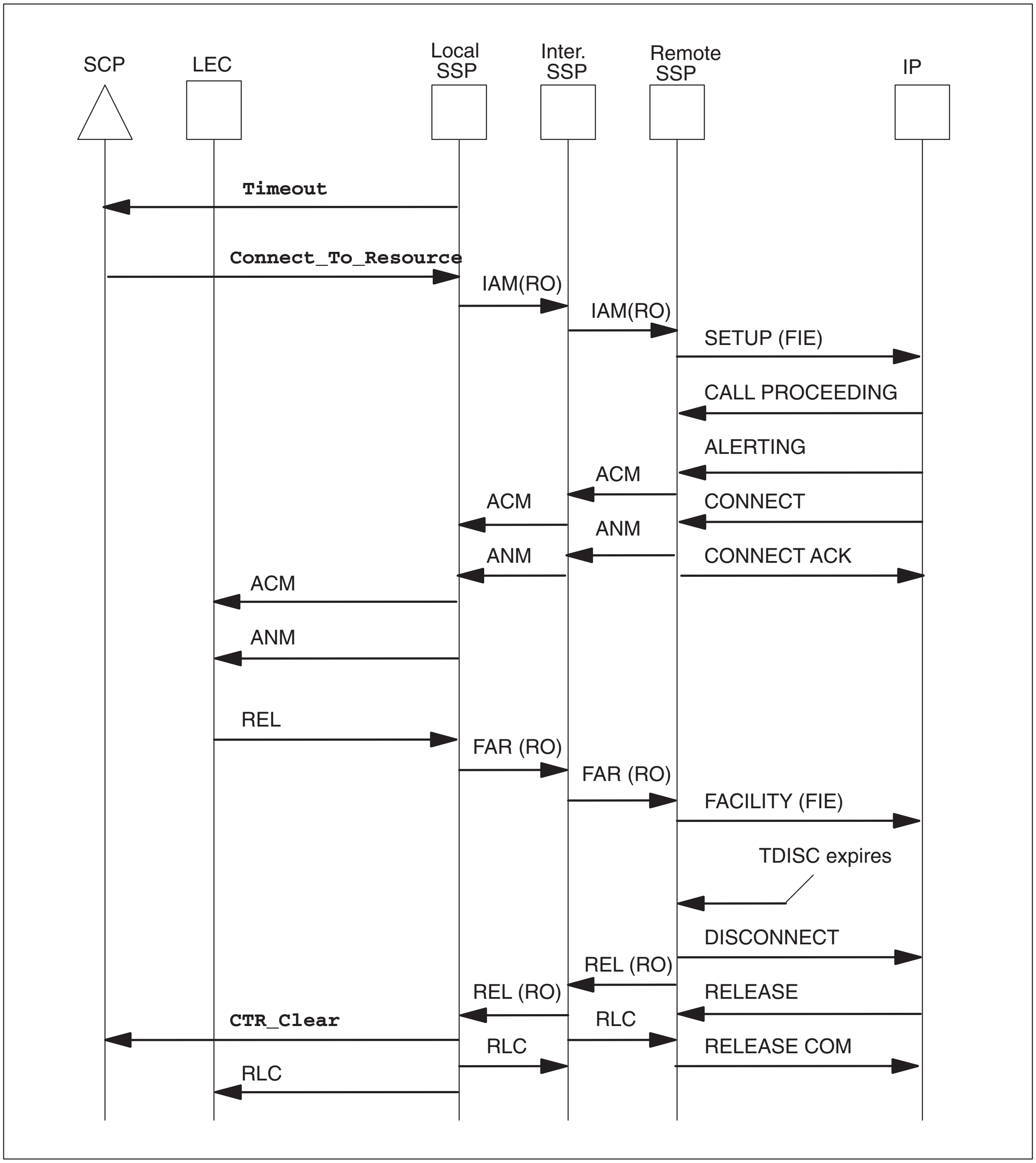
If timer TDISC expires before the remote IP responds to the FAR or FAC message with the `cancelIPResource` operation, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate then the local switch. The RO parameter contains a Return Error component with the Error Code set to `ipTimeout`. The IP CTR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - When the T1 timer is not running, the following actions are performed:
    - A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - The CTR-Connection is cleared.
  - When the T1 timer is running, the following actions are performed:
    - The switch awaits the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `userAbandon`.
    - The CTR-Connection is cleared.
  - When the T1 timer expires before the SCP response is received, a fatal application error is detected.

  Note: The ***`ClearCause`*** value of `ipTimeout` is not sent to the SCP in this scenario.

Figure 3-23 illustrates this scenario.

Figure 3-23
Timer TDISC expires during caller abandon, T1 timer not running



Remote switch-initiated Clearing of a CTR-Connection

The remote switch may send a REL message that clears the CTR-Connection. The following sections describe several scenarios where this occurs.

All channels busy

As explained earlier, the remote switch attempts to establish a connection to the remote IP when it receives an IAM containing an RO parameter. If the remote switch is unable to locate an idle trunk member terminating to the remote IP, the following actions are performed:

- At the remote switch, a REL message containing a RO parameter is sent to the intermediate then the local switch. The RO parameter contains a Return Error component with the Error Code set to `channelsBusy`.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `channelsBusy`.
  - The call is not cleared toward the controlling leg or passive leg.

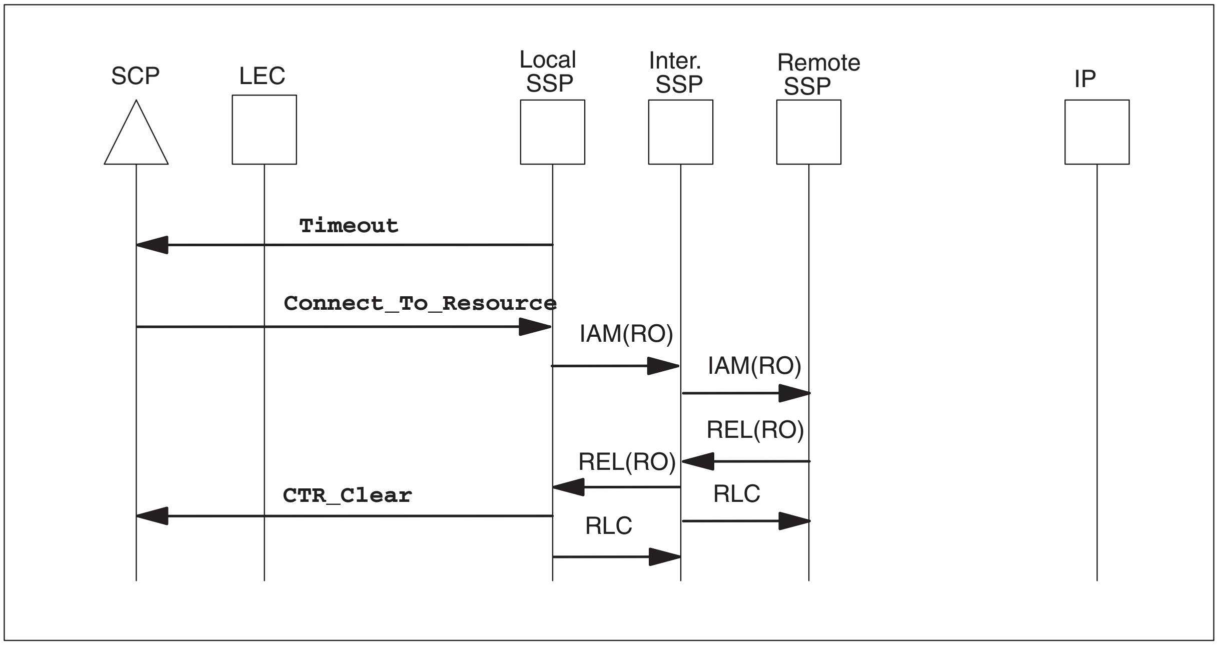
Refer to Figure 3-23 “Timer TDISC expires during caller abandon, T1 timer not running.”

For CONNECT_ONLY, if the remote switch is unable to locate an idle trunk member terminating to the remote IP, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate and then local switch.
- At the intermediate switch, the REL message is passed without modification to the local switch
- At the local switch, a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
- The call is not cleared toward the controlling leg or passive leg.

Figure 3-24 illustrates this scenario.

Figure 3-24
All channels busy at the remote switch



Timer TSTRC Expires

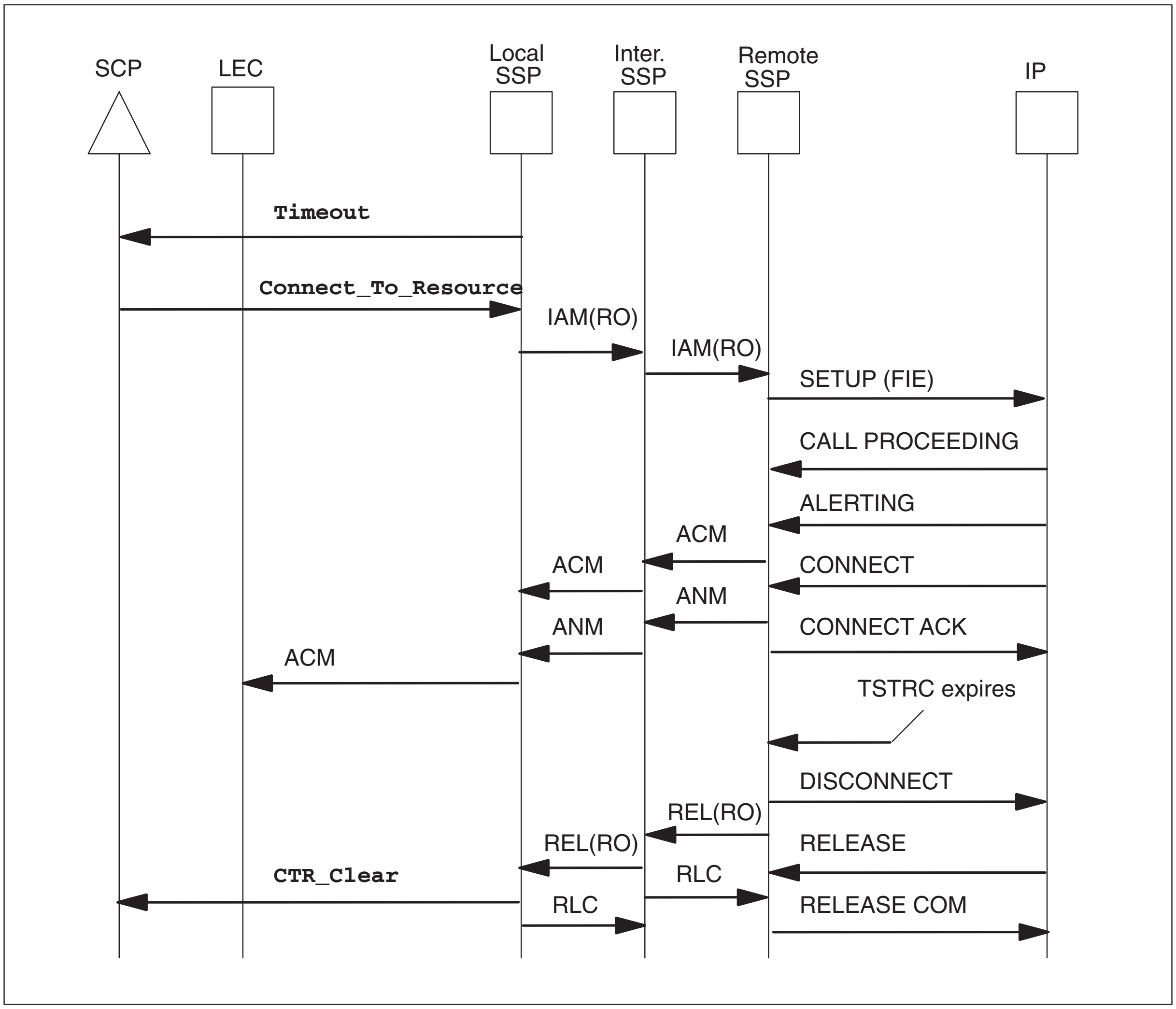
The remote switch starts timer TSTRC when the IP answers. If the timer expires during an active CTR-Connection, the following actions are performed:

- It the remote switch, a REL message containing a RO parameter is sent to the intermediate then the local switch. The RO parameter contains a Return Error component with the Error Code set to `ipTimeout`. The IP CTR-Connection is cleared.
- At the intermediate switch, the REL message containing the RO parameter is passed without modification to the local switch.
- At the local switch, the following actions are performed
  - — When the T1 timer is not running, the following actions are performed:
    - – A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`.
    - – The call is not cleared toward the controlling leg or passive leg.
  - — When the T1 timer is running, the following actions are performed:

- — The switch awaits the receipt of a **`Call_Info_To_Resource`** message from the SCP. When received, the message is discarded and a **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `ipTimeout`. The call is not cleared toward the controlling leg or passive leg.
- — If the T1 timer expires before the SCP response is received, a fatal application error is detected.

Figure 3-25 illlustrates this scenario.

Figure 3-25
Timer TSTRC Expires at the Remote switch, T1 Timer Not Running



Unexpected switch Errors

The remote switch may encounter unexpected error conditions which prevent the establishment of the CTR-Connection. Examples of unexpected errors include:

- The remote switch is unable to translate the Called Party Number and identify a route list.
- The remote switch attempts to terminate to a non-PRI or ISUP IMT(IPTRUNK) agent.
- An inswitch feature sends the call to a treatment.

When an unexpected error condition occurs, the following actions are performed:

- At the remote switch, a REL message is sent to the intermediate/local switch.The Cause Indicator parameter identifies the problem encountered by the remote switch.
- At the intermediate switch, the REL message is passed without modification to the local switch.
- At the local switch, the following actions are performed:
  - A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.
  - The CTR-Connection is cleared.
  - The call is not cleared toward the controlling leg or passive leg.

Intermediate switch-initiated Clearing of a CTR-Connection

The intermediate switch may encounter unexpected error conditions which prevent the establishment of the CTR-Connection. Examples of unexpected errors include:

- The intermediate switch is unable to translate the Called Party Number and identify a route list.
- The intermediate switch attempts to terminate to a non-ISUP IMT agent.
- An inswitch feature sends the call to a treatment.

When an unexpected error condition occurs, the following actions are performed:

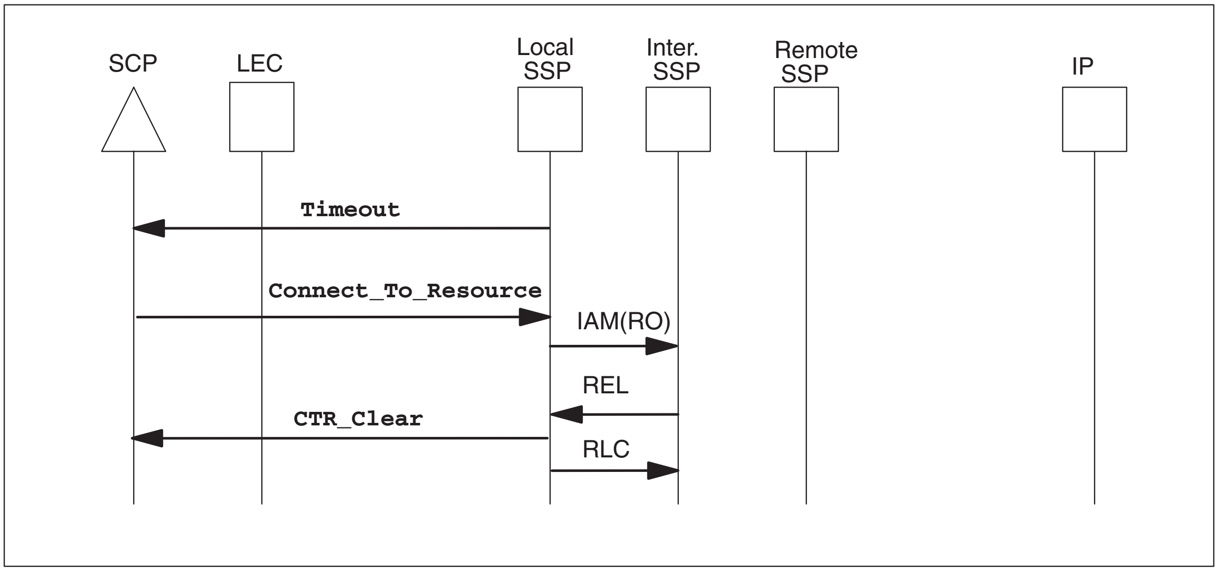
- At the intermediate switch, a REL message is sent to the local switch. The Cause Indicator parameter identifies the problem encountered by the intermediate switch.
- At the local switch, the following actions are performed:
  - A **`CTR_Clear`** message in a conversation package is sent to the SCP with a ***`ClearCause`*** parameter value of `abort`.

— The CTR-Connection is cleared.

— The call is not cleared toward the controlling leg or passive leg.

Figure 3-26 illustrates this scenario.

Figure 3-26
Unexpected Error Condition at the Intermediate switch



Billing interactions with the AMAMeasure parameter

The inclusion of an ***AMAMeasure*** parameter with a value of `connectTimeRecordedDestinationSSP` in the **`Connect_To_Resource`** messages indicates that the switch is required to bill the connection to the resource (inswitch resource or IP). The switch will produce a CDR for the Resource connection. This billing record is avoided by the absence of the ***AMAMeasure*** parameter in the **`Connect_To_Resource`** message.

The CDR generated for the CTR-Connection appears as if the originator (***LegID*** 0) placed the call to the resource. Major differences between CTR-Connection and STR-Connection are:

- the ANSTYPE field of the CDR will never have a value of 12 (early billing without answer) or 13 (early billing with answer) for a CTR-Connection.
- the CALLDUR of the CTR-Connection is included with the CALLDUR of the initial call (***LegID*** 0 to ***LegID*** 1).

CDR fields of interest for calls with AMAMeasure

The BILLNUM, ANISP, INFODIG, and ANISUFF for the CTR-interaction are copied from the initial CDR.

For SS7 terminations, the Network-specific Information (NSI) parm in the ACM or ANM messages may update the COMPCODE, ANSTYPE, FINTKGRP, FINSID, and FINTKMEM fields of the CDR. The NSER0003 SOC (Inter/Intra IMT) must be enabled. Additionally, the NSI is only processed in the ACM message for Intra IMTs.

CDR fields of interest for calls without AMAMeasure

The absence of the ***`AMAMeasure`*** parameter essentially freezes the recording unit. The CDR is not updated with any information pertaining to the CTR-interaction.

OFCVAR CDR_UNAVAIL_BLOCK

When this office parameter is set to Y, a CTR-interaction may fail when the switch is unable to allocate a recording unit. The CTR-interaction failure would occur when a **`Connect_To_Resource`** is recevied containing an ***`AMAMeasure`*** parameter and recording unit allocation fails.

RLT Interactions with CTR-Connections

The terminator of the call (***`LegID`*** 1) or the ***`DestinationAddress`*** parameter may connect the call to an IP or Enhanced Services Platform (ESP) that utilizes Release Line Trunking (RLT). There are several interactions between the behavior RLT produces on the bridging UCS DMS-250. The handling of these scenarios is outside the scope of this feature. The following sections discuss how a CTR-Connection handles the RLT requests. For more information about RLT, refer to the *UCS DMS-250 SS7 RLT Feature Application Guide*.

Redirection & Third-Party Interaction

In both cases, a Billing FAR or FAC is sent from the ESP to the Bridging UCS DMS-250 to initiate billing on the switch. The Bridging UCS DMS-250 sends an SS7 FAA or a PRI FAC message to the ESP indicating the acceptance of the incoming message and an FRJ/FAC indicating the rejection of the FAR. A call involved in a CTR-Connection always sends an FRJ/FAC to any FAR/FAC from an ESP. This prevents any RLT functionality from being carried out. The CTR-Connection is maintained until the ESP releases from the call.

Operator Initiated

Currently there are no interactions for this type of RLT call. Neither leg of the call can enter a CTR-Connection.

Feature Interactions

Reorigination with Specialized Tone Receiver during a two-party call is disabled during a CTR-connection.

Fatal application errors

Fatal application errors occur when CAIN call processing is unable to continue due to an unexpected error. Table 3-2 provides errors that can occur during a CTR-Connection.

Table 3-2
Fatal application errors

Error type	Log generated	Reported to SCP?	Action performed
DestinationAddress parameter present in a **Connect_To_Resource** in a Response package	CAIN200	Yes (Note 1)	Switch applies AINF
DestinationAddress and ***DisconnectFlag*** present in a **Connect_To_Resource** in a Response or Conversation package	CAIN200	Yes (Note 2)	Switch applies AINF
Local switch is unable to identify a route index for the IP	CAIN200	Yes (Note 2)	A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of channelsBusy. The call is not cleared toward the controlling or passive leg.
Remote switch is unable to identify a route index for the IP	CAIN200	Yes (Note 2)	A **CTR_Clear** message in a conversation package is sent to the SCP with a ***ClearCause*** parameter value of abort. The call is not cleared toward the controlling or passive leg.

Note 1: The switch sends an **Application_Error** to the SCP, with ***ErrorCause*** set to unexpectedCommunication.

Note 2: The switch sends an **Application_Error** to the SCP, with ***ErrorCause*** set to erroneousDataValue.

—continued—

Table 3-2
Fatal application errors (continued)

Error type	Log generated	Reported to SCP?	Action performed
Remote switch attempts to terminate to a non-PRI or ISUP IMT (IPTRUNK) agent, or to a PRI lacking the IPTRUNK option	CAIN200	Yes (Note 2)	a **`CTR_Clear`** message is sent with a ***`ClearCause`*** value of `abort`.
Selected agent does not use SS7 signaling (between tandem switches)	CAIN200	Yes (Note 2)	AINF

Note 1: The switch sends an **`Application_Error`** to the SCP, with ***`ErrorCause`*** set to `unexpectedCommunication`.

Note 2: The switch sends an **`Application_Error`** to the SCP, with ***`ErrorCause`*** set to `erroneousDataValue`.

—end—

Nonfatal application errors

Table 3-3 provides errors that can occur during a CTR-Connection.

Table 3-3
Nonfatal application errors

Error type	Log generated	Reported to SCP?	Action performed
DestinationAddress contains a nature of address value other than National or VPN	CAIN100	No	National NOA is assumed for the ***DestinationAddress***.
Switch receives a **Connect_To_Resource** message with a ***DestinationAddress*** parameter and the STR_CONNECTION_TYPE (table CAINPARM) is set to NONE.	CAIN100	Yes	**CTR_Clear** is sent to SCP in a conversation package with a ***ClearCause*** of taskRefused.
DestinationAddress contains too many digits	CAIN100	No	Excess digits are truncated

Note: For more information on nonfatal application errors, refer to Volume 3, Chapter 10, “Incoming CAIN messages.”

Associated logs

CAIN100, CAIN200

Associated OMs

CAINMSGR, CAINAGOM, CAINTRIG, CAINUIF, CAINIP

Restrictions/limitations

SOC options CAIN0600 Coin Digit Collect, CAIN0603 STR Connection, CAIN0606 1129-Style IP, CAIN0607 Virtual IP, and CAIN0800 Mid Call Services 1, as well as SOC option CAIN0801 Mid Call Services 2 need to be activated to enable all the functionality of **Connect_To_Resource**.

CAIN does not allow a **Request_Report_BCM_Event** component to be received in a package with a **Connect_To_Resource** operation. This is considered a fatal application error.

The connectToIPResource operation is not yet supported by PRI based IPs. Therefore the sendToIPResource operation value is used.

During digit collection for a circuit mode data call, no prompt is played.

Termination_Attempt query messages shall include the ***STRConnection*** parameter instead of the ***CTRConnection*** parameter.

Multiple resources applied to call legs is not supported.

A request for in-switch digit collection requesting anything other than fixed 0 digits results in a **CTR_Clear** in a conversation package with a ***ClearCause*** of taskRefused.

If the switch receives a **Connect_To_Resource** message with a ***LegID*** set to 2 in a call configuration other than CC6 or CC11, then the ***LegID*** setting is ignored, and the resource is played to all parties on the call.

The **Connect_To_Resource** message is supported serially. The UCS DMS-250 switch does not support the receipt of a **Connect_To_Resource** message while it is processing another **Connect_To_Resource** message. A **CTR_Clear** message with a ***ClearCause*** of taskRefused is sent if such a situation occurs.

Virtual IP interaction

The NetworkBuilder Virtual IP (VIP) feature enables the UCS DMS-250 switch to simulate an Intelligent Peripheral (IP) for services that require announcements or announcements plus digit collection. This is accomplished through two IP-based parameters, ***FlexParameterBlock*** and ***IPReturnBlock***, during conversational messaging between the UCS DMS-250 switch and SCP.

Note: The **Connect_To_Resource** information discussed within this chapter is specific to reorigination scenarios at ***O_Mid_Call***.

Conversational messaging is the means by which the SCP instructs the switch to play tones or announcements, play tones or announcements and collect digits, or route to an IP for further processing. Virtual IP uses a single **Send_To_Resource** and **Resource_Clear** (or **Connect_To_Resource** and **CTR_Clear**) conversation to control multiple announcements or announcements and digit collections.

A benefit of Virtual IP is its consolidation of multiple collectibles into one conversational message pair, reducing switch real time processing. If the SCP chooses to engage the UCS DMS-250 switch in Virtual IP conversational messaging, seven data collectibles are supported (five at a time): announcement, authcode, card, address, PIN, account code, and the UNKNOWN collectible. If more data (more than the allotted five) needs to be collected, the SCP can engage the UCS DMS-250 switch in additional Virtual IP or conventional conversational messaging.

Virtual IP supports the ***FlexParameterBlock*** resource type in **Send_To_Resource** and **Connect_To_Resource** messages. In this discussion of Virtual IP, a **Send_To_Resource** or **Connect_To_Resource** message with a ***ResourceType*** of ***FlexParameterBlock*** will have its ***StrParameterBlock*** referred to as the ‘FlexParameterBlock’.

If a **Send_To_Resource** or **Connect_To_Resource** message does not include a ***DestinationAddress*** parameter, NetworkBuilder will invoke the Virtual IP capability. In this case, the ***FlexParameterBlock*** parameter supplies the instructions for the UCS DMS-250 switch to use in its interaction with the user. The ***IPReturnBlock*** parameter then supplies the

SCP (through a **Resource_Clear** or **CTR_Clear** message) with the subscriber dialed digits that were collected by the Virtual IP, based on the ***FlexParameterBlock*** collectible list.

The resource type tells the UCS DMS-250 switch how to decode the ***StrParameterBlock*** portion of the **Send_To_Resource** or **Connect_To_Resource** message.Virtual IP does not differentiate between the **Send_To_Resource** and **Connect_To_Resource** messages when processing the ***FlexParameterBlock***.

The ***FlexParameterBlock*** combines the functionality of the conventional **Send_To_Resource** options (play an announcement, and play an announcement and collect digits) into a single parameter, so that both functions can be sent to the switch as collectibles within the ***FlexParameterBlock***.

The ***FlexParameterBlock*** resource type uses the ***StrParameterBlock*** choice of ***FlexParameterBlock*** to define the following information:

- type of collectible
- tone or announcement to play (resource)
- whether or not tones and announcements are interruptible
- number of digits to collect (MINDIGS and MAXDIGS)
- permanent signal timer

Table 4-1 provides the parameters the **Send_To_Resource** and **Connect_To_Resource** messages can contain for Virtual IP interactions.

Table 4-1
Send_To_Resource and Connect_To_Resource message parameters for Virtual IP

Parameter	Usage	Definition
ResourceType	Required	This parameter contains the resource type ***FlexParameterBlock***.
StrParameterBlock	Required	This parameter contains the ***FlexParameterBlock*** choice option which contains the following information: • Resource encoding authority, which defines how to interpret the subsequent data within the ***FlexParameterBlock*** choice. • ***FlexParameterBlock*** format choice: NT • Flextag: VIP for this feature. Specifies the specific functionality. • Collectible type: ANNC, ADDR, AUTH, CARD, ACCT, PIN, and UNKNOWN • Resource information, such as the tone or announcement to play and whether tones and announcements are interruptible. • Optionally, minimum and maximum digits to collect (0 to 24), used for fast interdigital timing, and to determine end of dialing for a collectible when multiple collectibles are entered without interruption. (ADDR, AUTH, CARD, ACCT, PIN, UNKNOWN) • Timer, which identifies the permanent signal timer value (0 to 15 seconds) for the initial digit. The trunk group partial dial timer controls the interdigital timing for the remaining digits. (ADDR, AUTH, CARD, ACCT, PIN, UNKNOWN)

—continued—

Table 4-1
Send_To_Resource and Connect_To_Resource message parameters for Virtual IP (continued)

Parameter	Usage	Definition
AnswerIndicator	Optional	This parameter instructs the UCS DMS-250 switch to provide answer supervision to the originating agent while the caller is connected to the resource. The UCS DMS-250 switch sends answer indication to the caller in response to the Play Announcement request if answer indication has not already been sent. ***Note 1:*** ***AnswerIndicator*** is not used for the **Connect_To_Resource** message. ***Note 2:*** ***AnswerIndicator*** is only used for SS7 and PRI originators. ***Note 3:*** ***AnswerIndicator*** does not affect billing (internal resources only) at the querying switch.
AMAMeasure	Optional	This parameter instructs the UCS DMS-250 switch to start call duration timing if the value is connectTimeRecordedDestinationSSP, so the time spent simulating an IP is captured in the total call duration time. The total call duration time value is stored in the CALLDUR field of the CDR. If the value is connectTimeRecordedDestinationSCP or connectTimeNotRecorded, no timing is begun. If more than one **Send_To_Resource** or **Connect_To_Resource** is sent during a single call, subsequent ***AMAMeasure*** parameters will not reset nor stop timing, but will start timing if it has not already begun.
ExtensionParameter	Optional	***ExtensionParameter***s require the CAIN0200 SOC option.
pretranslator Name	Optional	This extension parameter contains an index into table CNPREXLA, used with the ADDR collectible.
—end—		

The pretranslatorName extension parameter is used to minimize ambiguous dialing. This optional parameter contains an index into the table CNPREXLA. Table CNPREXLA provides a pretranslator name used by the UCS DMS-250 switch, through table STDPRTCT, to pretranslate the collected address digits. All pretranslators found in table STDPRTCT supported by CAIN are allowed (OFFNET, ONNET, FORCED_ONNET, VIRTUAL_ONNET). The pretranslator performs call typing and defines the minimum and maximum number of digits to collect. Once the address digits are collected, the UCS DMS-250 switch buffers any remaining digits for the next collectible. The ***FlexParameterBlock*** minimum and maximum digit information defines the number of billing digits to collect. When it can be determined (based on the pretranslator name), the ADDRESS digits returned to the SCP are assigned the appropriate nature of address (NOA), not UNKNOWN. Any misdialing errors or other dialing ambiguities continue to be handled by the SCP.

Note: The new pretranslator name returned in the pretranslatorName extension parm is used for the rest of the call, including reorigination.

When a **Send_To_Resource** or **Connect_To_Resource** message is received, the UCS DMS-250 switch must determine if it should simulate an IP or route the call to an IP for processing. The UCS DMS-250 switch simulates an IP when the following criteria are met, however, as soon as one of the criteria fails, the rest are ignored and appropriate "error handling" occurs.

- The IP ***DestinationAddress*** is not received.
- The **Send_To_Resource** or **Connect_To_Resource** message is received in a conversation package.
- The resource type of the **Send_To_Resource** or **Connect_To_Resource** message is ***FlexParameterBlock***.
- The VIP SOC, CAIN0607, is set to "ON".
- The 'NetworkBuilder' encoding authority is used to code the ***FlexParameterBlock***.
- The VIP tag is present in the ***FlexParameterBlock***.

If all of the above conditions do not exist, the following occurs:

If an IP ***DestinationAddress*** is received in the **Send_To_Resource** or **Connect_To_Resource** message, the UCS DMS-250 switch immediately forwards the SCP request to the designated IP for processing. The call is handled by the IP as a STR- or CTR-Connection. Refer to the chapters on STR- and CTR-Connections in this volume for more information.

If the VIP SOC, CAIN0607 is not on, then a **Resource_Clear** or **CTR_Clear** message is returned with a ***ClearCause*** value of taskRefused.

If the **Send_To_Resource** or **Connect_To_Resource** message (with a resource type of ***FlexParameterBlock***) is received in a response package, a **Report_Error** message is sent in a uni-directional package with an ApplicationErrorString of 'ErrorCause' and an ***ErrorCause*** of unexpectedCommunication.

If the resource type of the **Send_To_Resource** or **Connect_To_Resource** message is not ***FlexParameterBlock***, the **Send_To_Resource** or **Connect_To_Resource** message is handled appropriately for the resource type (play announcement or tones, or play announcement or tones and collect digits).

If the encoding authority used to encode the ***FlexParameterBlock*** is not the ‘NetworkBuilder’ encoding authority, then a **Resource_Clear** or **CTR_Clear** message will be returned with a ***ClearCause*** parameter value of failure and a ***FailureCause*** of applicationError.

If the VIP tag is not present in the ***FlexParameterBlock*** (no other value is currently supported), then a **Resource_Clear** message will be returned with a ***ClearCause*** value of taskRefused.

For more information on the handling of errors by the Virtual IP, please refer to Table 4-9.

Virtual IP data collectibles

The following data collectibles are supported for Virtual IP:

- announcement
- address
- authcode
- card
- account code
- PIN
- UNKNOWN

With VIP, the UCS DMS-250 switch can identify the data being requested and can populate the appropriate CDR fields upon collecting the subscriber data. As this data is collected and identified, it can be stored in call processing data for future use in processing the call. The data can be re-sent to the SCP in future TCAP query messages through their various parameters. When the CAIN_PROTOCOL_VERSION parameter in table CAINPARM is set to V2 or lower, the authcode and card are sent in the ***ChargeNumber*** parameter with an NOA of AUTH; address in the ***CollectedAddressInfo*** parameter; PIN is sent in the ***CollectedDigits*** parameter with an NOA of PIN; and account code is sent up in the ***AccessCode*** parameter. This ability of the SSP to return this collected data to the SCP at later points in the call can aid the SCP in its service processing of future queries.

Note: If the CAIN_PROTOCOL_VERSION parameter is set to V3 or higher, the ***AccessCode*** parameter is not populated and therefore is not sent.

When deciding what collectibles to request, the SCP service creators must determine what the collectible will be used for, taking into consideration how the switch will interpret and use the subscriber collected digits in its processing.

Note: Each of the collectibles will overwrite any digits of the same type, both in the in-switch call processing logic and in the applicable CDR fields (whether the digits were previously determined or subscriber collected, either through in-switch dialing plans or as a Virtual IP collectible). For a list of applicable CDR fields, refer to "Billing" in this chapter. For more information on AMA digits overwriting, refer to Volume 3, Chapter 12, "Incoming CAIN message parameters."

The ADDRESS collectible is always treated as the originally dialed number populating the DIALEDNO CDR field and ***CollectedAddressInfo*** parameter of SCP outgoing messages. If this ADDRESS collectible is the first address collected, as with the *Off_Hook_Immediate* trigger, there is no conflict. If this ADDRESS is a subsequent address, as with the triggers found at the ***O_Feature_Requested***, ***Info_Collected*** and ***Info_Analyzed*** TDPs, then the previously collected address (through in-switch digit collection and perhaps used to trigger) will be moved from the orig_dialed_number to the ua_number (universal access), as is done with *O_Feature_Requested* address processing. The Virtual IP ADDRESS collectible digits populate the orig_dialed_number. This functionality can be used to re-subscribe using the address. For example, a call can subscribe using the first address, query the SCP through an ***Info_Analyzed*** trigger, receive a Virtual IP **Send_To_Resource** response that has an ADDRESS collectible, and then this new address could be re-evaluated and subscribe to the Busy triggers.

Note: The address is the only subscriber dialed digits that re-evaluates subscription. Authcodes and ANIs are not used to re-evalutate subscription.

If the action of replacing the orig_dialed_number with the newly collected address digits is not the desired action, then the UNKNOWN collectible can be used instead.

The UNKNOWN collectible also can be used by the SCP when it needs to collect subscriber digits that do not fall into the set of digit collectibles provided through Virtual IP. Whether the UNKNOWN collectible is used for specially handled address digits or some SCP specific digits, the subscriber dialed digits are not used by call processing logic, nor will they populate CDR fields (unless they are returned in an **Analyze_Route** message for population of the SCPBILL CDR field).

Virtual IP will allow the following combinations of collectibles:

- ANNC only – only the specified announcements will be played, no digits will be collected.
- ANNC and others – any combination and number of announcements and the collectibles will be collected. (ADDR, AUTH, CARD, ACCT, PIN, UNKNOWN) – up to 5 digit collectibles.

Special Considerations for Virtual IP

If the UCS DMS-250 switch previously collected authcode, card, address, PIN, and account code information and forwarded it to the SCP, the SCP can return a **`Send_To_Resource`** message with a ***`FlexParameterBlock`*** that instructs the switch to collect any or all of these same data elements. If this occurs, the new subscriber dialed digits will overwrite the pre-existing data stored in call processing and the CDR.

As stated earlier, the SCP can specify whether a resource (tone or announcement) is or is not interruptible. If the resource is interruptible and the subscriber dials through all tones and announcements, then ambiguous dialing can occur. In this case it cannot discern where a collectible digit string ends and another begins.

For example, let's assume the ***`FlexParameterBlock`*** instructs the UCS DMS-250 switch to play an announcement and collect between 3 and 15 address digits, then play another announcement and collect between 14 and 16 Travel Card Number (TCN) digits. In this example, the subscriber is familiar with the dial plan and enters a 10-digit address followed by a 14-digit TCN during the first announcement. The first announcement is interrupted by the initial digit, however, the continuous string of digits thereafter prevents the second announcement from playing, which is normally used to separate collectibles.

Based on the ***`FlexParameterBlock`*** minimum and maximum digit instructions, the switch assumes the first 15 digits are the address, and the remaining digits are the TCN. This information is reported to the SCP, and it is the SCP's responsibility to resolve ambiguous dialing that occurs.

To avoid ambiguous dialing situations, subscribers should wait for announcements to begin before entering digits or enter an end of dialing digit between collectibles; or the operating company can define the resource as uninterruptible for collectible types, or send the UCS DMS-250 switch the `pretranslatorName` extension parameter to identify address digit types.

Once the UCS DMS-250 switch has collected all information defined in the ***`FlexParameterBlock`*** choice, it returns the data to the SCP in the ***`IPReturnBlock`*** of the **`Resource_Clear`** message. The ***`IPReturnBlock`*** contains response information for all collectibles received in the ***FlexParameterBlock***.

When the permanent signal time expires no digits are sent for the particular collectible.

Resource_Clear and CTR_Clear messages

The UCS DMS-250 switch reports the collected information to the SCP in an ***IPReturnBlock***, which is sent in a **Resource_Clear** or **CTR_Clear** message.

The ***IPReturnBlock*** contained in a **Resource_Clear** message is a duplicate of one contained in a **CTR_Clear** message.

Virtual IP supports all currently supported **Resource_Clear** ***ClearCause*** parameter values. Refer to Volume 3, Chapter 12, “Incoming CAIN message parameters,” for a list of supported ***ClearCause*** values.

Table 4-2 provides a list of **Resource_Clear** and **CTR_Clear** message parameters used for Virtual IP.

Table 4-2
Resource_Clear and CTR_Clear message parameters for Virtual IP

Parameter	Usage	Definition
IPReturnBlock	Required	This parameter contains the collected digits requested by the ***FlexParameterBlock*** of the **Send_To_Resource** message. Digits are returned by collectible type.
ClearCause	Required	This parameter indicates the reason a connection between a caller and a Virtual IP resource was terminated.
FailureCause	Optional	This parameter is sent when a ***ClearCause*** parameter is encoded as failure.

Table 4-3 provides a mapping of returned NOA and numbering plans to collectible types.

Table 4-3
NOAs and numbering plans returned by collectible type

Collectible	Nature of address (NOA)	Numbering plan
ADDR	UNKNOWN	UNKNOWN
	Note: If the actual NOA can be determined it will be populated as such, otherwise UNKNOWN will be returned for both the NOA and numbering plan will be ISDN.	
AUTH	authcode	private
CARD	mccs	private
ACCT	acct	private
PIN	pin	private
UNKNOWN	UNKNOWN	UNKNOWN

Virtual IP messaging

Table 4-4 provides an example of a Virtual IP `Send_To_Resource` message containing a request for multiple collectibles.

Table 4-4
Send_To_Resource with request for multiple collectibles

Parameters						
ResourceType	FLEX_PARAMETER_BLOCK					
StrParameter Block	NT DMS250 RESOURCE ENCODING AUTHORITY					
	VIP					
	COLL_TYPE	**RSRC**	**INTERUPT**	**MIN**	**MAX**	**TIMER**
	ANNC	2	FALSE			
	AUTH	7	TRUE	7	10	5

—continued—

Table 4-4
Send_To_Resource with request for multiple collectibles (continued)

Parameters						
	ADDR	8	TRUE	10	10	5
	PIN	9	FALSE	4	4	5
Answer Indicator	TRUE					
AMAMeasure	connectTimeDestinationRecordedSSP					
—end—						

Based on information in Table 4-4, the UCS DMS-250 switch can process a call in the followng manner:

- An uninterruptible branding announcement: “Thank you for using Long Distance Mail-R-Us.”

  Note: These are sample announcements only.

- An interruptible announcement: “Please enter your authcode.”
- Between 7 and 10 authcode digits are collected: ‘12341234’
  - These digits populate the BILLNUM field of the CDR

  Note: The collected digits are stored in call processing data to populate the CDR and for future use by call processing.

- An interruptible announcement: “Please enter your mailbox number.”
- 10 address digits are collected: ‘5432154321’
  - These digits populate the DIALEDNO field of the CDR
- An uninterruptible announcement: “Enter the PIN for your mailbox.”
- 4 PIN digits are collected: 7676
  - These digits populate the PINDIGS field of the CDR

Note: In this example there isn’t any validation in-switch. All of the above digit collectibles are sent to the SCP for validation and processing.

Table 4-5 corresponds with the example shown in Figure 4-1 of a **Resource_Clear** with the ***IPReturnBlock***.

Table 4-5
Resource_Clear with the IPReturnBlock

Parameters			
ClearCause	NORMAL		
IPReturnBlock	NT DMS250 RESOURCE ENCODING AUTHORITY		
	NOA	**NUMPLAN**	**DIGITS**
	AUTH	PRIV	12341234
	UNKNOWN	UNKNOWN	5432154321
	Note: If the actual NOA can be determined it will; otherwise, UNKNOWN will be returned for both the NOA and numbering plan.		
	PIN	PRIV	7676
AMAMeasure	connectTimeDestinationRecordedSSP		

If the ***FlexParameterBlock*** contains multiple collectibles, the switch processes them in the order received. Figure 4-1 provides a messaging example based on the above FlexParameterBlock example.

Figure 4-1
Request for multiple collectibles using Virtual IP conversational messaging

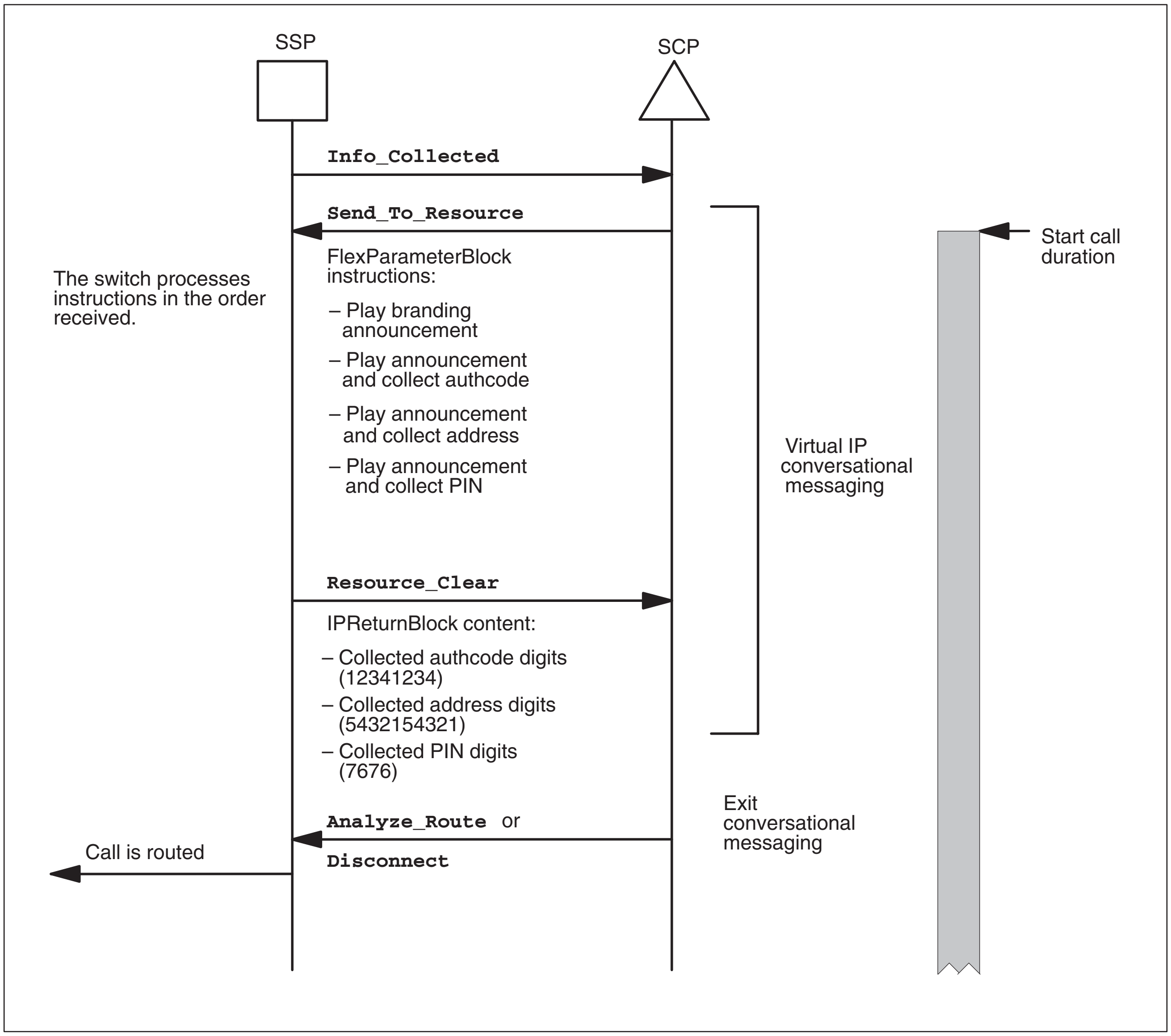

SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
The switch processes instructions in the order received.
Start call duration
Virtual IP conversational messaging
Resource_Clear
IPReturnBlock content:
– Collected authcode digits (12341234)
– Collected address digits (5432154321)
– Collected PIN digits (7676)
Analyze_Route or
Disconnect
Call is routed
Exit conversational messaging

Table 4-6 provides an example of a Virtual IP **Send_To_Resource** message containing a request for multiple collectibles including UNKNOWN.

Table 4-6
Send_To_Resource with request for multiple collectibles including UNKNOWN

Parameters						
ResourceType	FLEX_PARAMETER_BLOCK					
StrParameter Block	NT DMS250 RESOURCE ENCODING AUTHORITY					
	VIP					
	COLL_TYPE	**RSRC**	**INTERUPT**	**MIN**	**MAX**	**TIMER**
	ANNC	2	FALSE			
	AUTH	7	TRUE	7	10	5
	UNKNOWN	8	TRUE	10	10	5
	UNKNOWN	9	FALSE	4	4	5
Answer Indicator	TRUE					
AMAMeasure	connectTimeDestinationRecordedSSP					

Based on the example in Table 4-6, the UCS DMS-250 switch can process a call in the followng manner:

- An uninterruptible branding announcement: “Thank you for using Long Distance Mail-R-Us.”

  Note: These are sample announcements only.

- An interruptible announcement: “Please enter your authcode.”
- Between 7 and 10 authcode digits are collected:‘12341234’

  — These digits populate the BILLNUM field of the CDR

  Note: The collected digits are stored in call–processing data to populate the CDR and possibly be used later.

- An interruptible announcement: “Please enter your mailbox number.”
- 10 digits are collected: ‘5432154321’

  — These digits do not populate any CDR fields

- An uninterruptible announcement: “Enter the PIN for your mailbox.”

- 4 digits are collected: ‘7676’
  - — These digits do not populate any CDR fields

Note: In this example there isn’t any validation in-switch. All of the above digit collectibles are sent to the SCP for validation and processing.

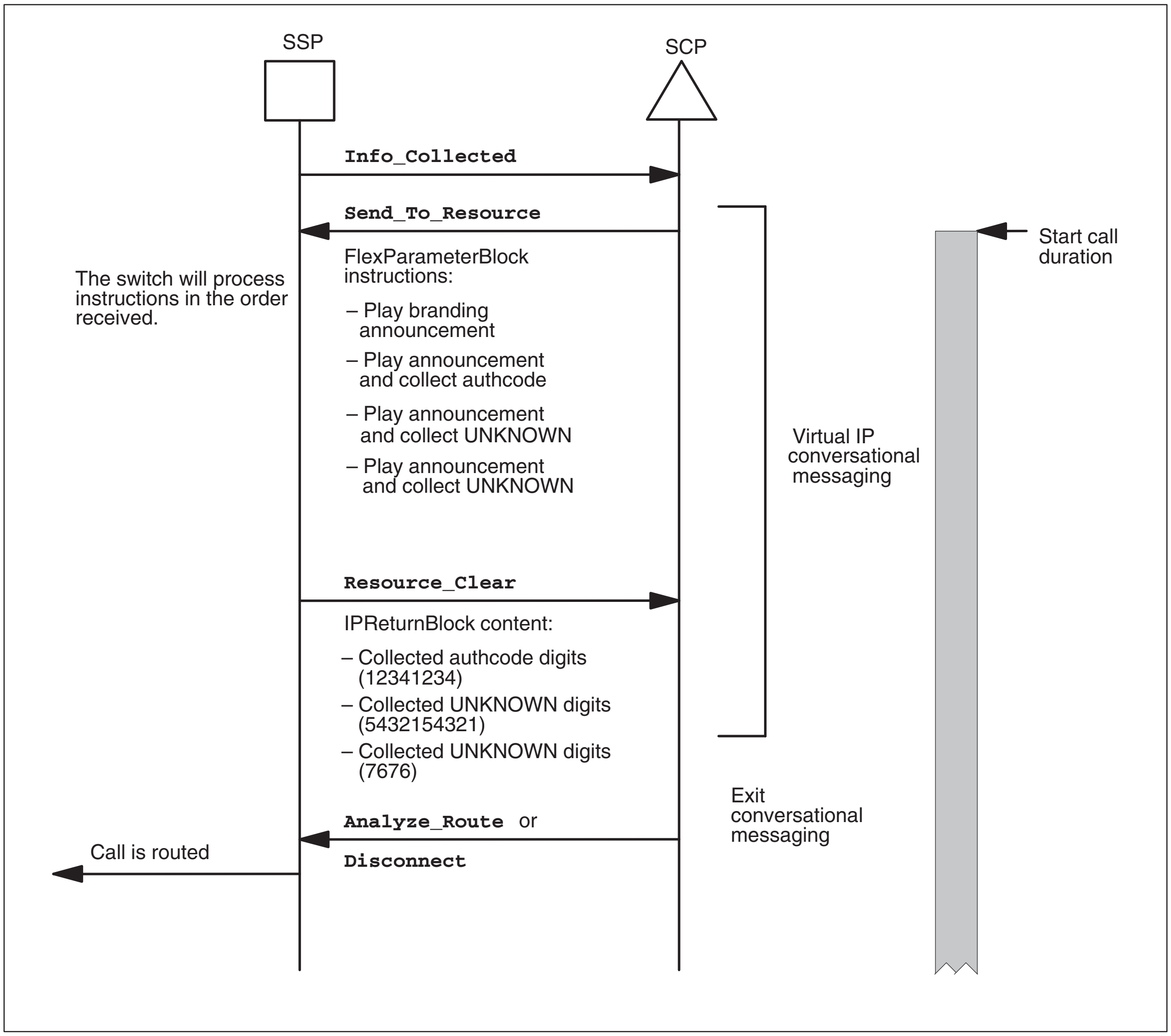
With the exception of AUTHCODE digits, collected digits are not saved in call processing or CDR fields in this example.

Table 4-7 corresponds with the example shown in Figure 4-2 of a **Resource_Clear** that contains a ***IPReturnBlock*** with an UNKNOWN collectible.

Table 4-7
Resource_Clear with the IPReturnBlock

Parameters			
ClearCause	NORMAL		
IPReturnBlock	NT DMS250 RESOURCE ENCODING AUTHORITY		
	NOA	**NUMPLAN**	**DIGITS**
	AUTH	PRIV	12341234
	UNKNOWN	UNKNOWN	5432154321
	UNKNOWN	UNKNOWN	7676
AMAMeasure	connectTimeDestinationRecordedSSP		

Figure 4-2
Request for multiple collectibles including UNKNOWN

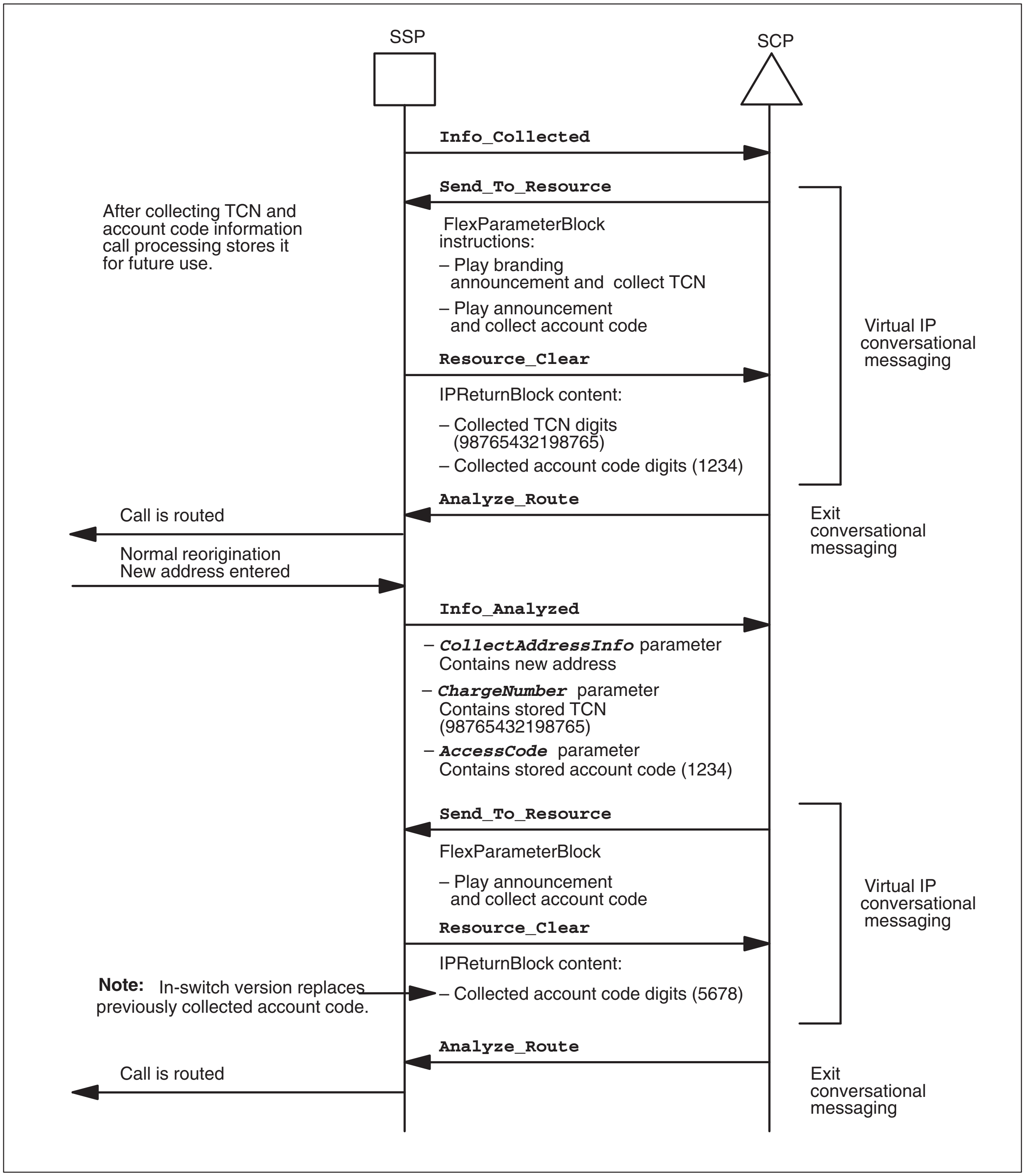

SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect UNKNOWN
– Play announcement and collect UNKNOWN
The switch will process instructions in the order received.
Virtual IP conversational messaging
Start call duration
Resource_Clear
IPReturnBlock content:
– Collected authcode digits (12341234)
– Collected UNKNOWN digits (5432154321)
– Collected UNKNOWN digits (7676)
Exit conversational messaging
Analyze_Route or
Disconnect
Call is routed

Using collected subscriber data

The information collected through Virtual IP conversational messaging is stored in call processing data. This data can be re-sent to the SCP in TCAP query messages that have parameters to support Virtual IP collectible data. This is possible because the ***FlexParameterBlock*** provides data identifiers for the switch to logically store the data once it's received from the subscriber.

Figure 4-3 provides an example of a Virtual IP call flow (when the CAIN_PROTOCOL_VERSION parameter in table CAINPARM is set to V2 or lower) where the collectible information is re-sent to the SCP. It shows that during prior Virtual IP messaging, the UCS DMS-250 switch collected and stored a TCN and account code. The subscriber invokes reorigination. The UCS DMS-250 switch traverses through the CAIN originating call model and at the **Analyze_Information** PIC is instructed to query the SCP. The UCS DMS-250 switch forwards the previously stored Virtual IP information to the SCP to aid in processing the call. The SCP requests the UCS DMS-250 switch to collect a different account code from the subscriber before allowing the call to route. If the CAIN_PROTOCOL_VERSION parameter is set to V3 or higher, numbers will not populate any V3 controlled parameters (for example, ***AccessCode***); therefore, they are not sent.

Figure 4-3
Virtuall IP subscriber data re-use call flow with CAIN_PROTOCOL_VERSION is V2 or lower


SSP
SCP
Info_Collected
Send_To_Resource
After collecting TCN and account code information call processing stores it for future use.
FlexParameterBlock instructions:
– Play branding announcement and collect TCN
– Play announcement and collect account code
Virtual IP conversational messaging
Resource_Clear
IPReturnBlock content:
– Collected TCN digits (98765432198765)
– Collected account code digits (1234)
Analyze_Route
Call is routed
Exit conversational messaging
Normal reorigination New address entered
Info_Analyzed
– CollectAddressInfo parameter Contains new address
– ChargeNumber parameter Contains stored TCN (98765432198765)
– AccessCode parameter Contains stored account code (1234)
Send_To_Resource
FlexParameterBlock
– Play announcement and collect account code
Virtual IP conversational messaging
Resource_Clear
IPReturnBlock content:
Note: In-switch version replaces previously collected account code.
– Collected account code digits (5678)
Analyze_Route
Call is routed
Exit conversational messaging

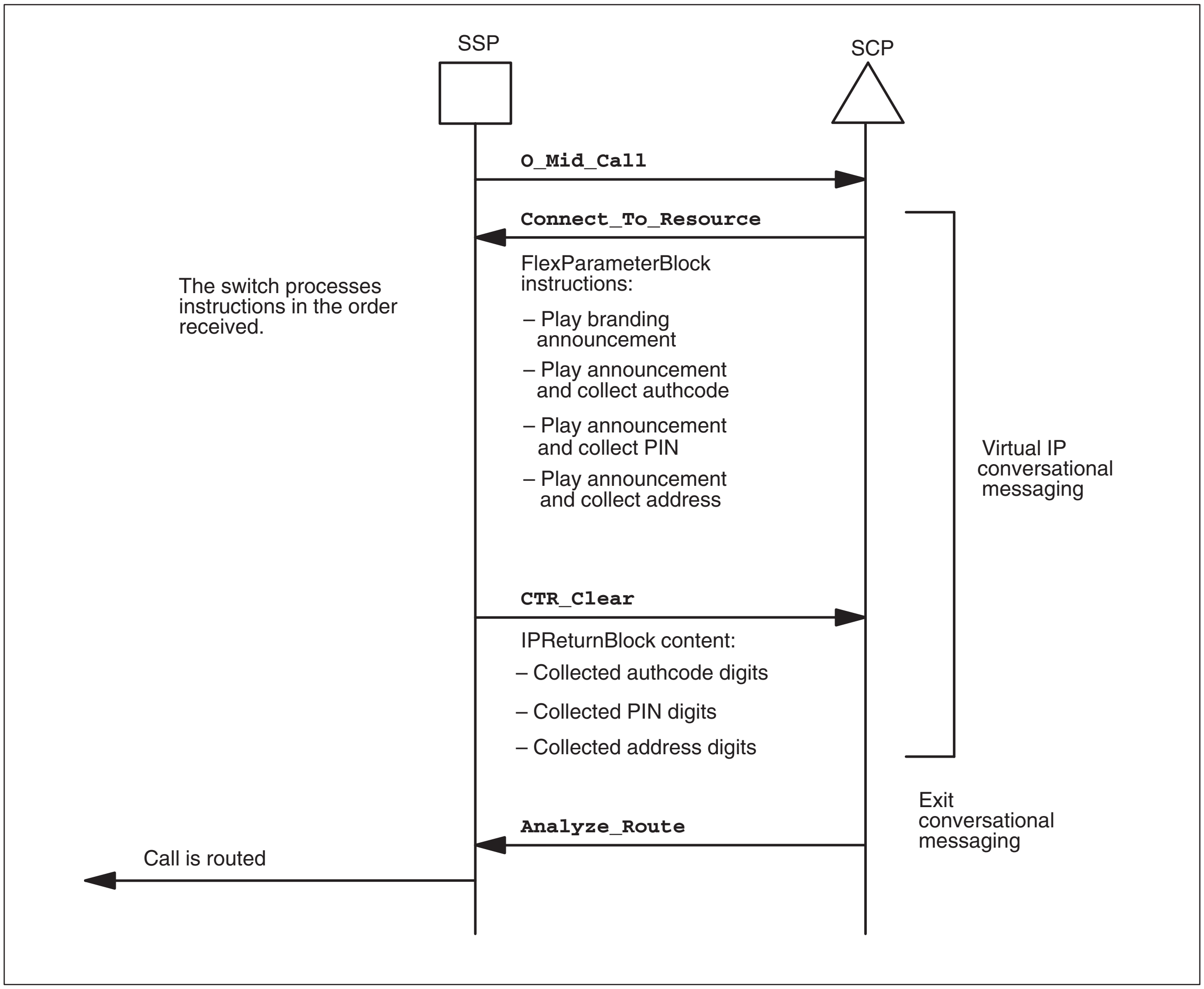
Virtual IP defined dial plans

Virtual IP conversational messaging can be used to define the dial plan in response to *O_Feature_Requested*, *Offhook_Delay*, and *O_IEC_Reorigination* trigger information. For example, if a call reorigination is received at the UCS DMS-250 switch and the *O_IEC_Reorigination* trigger criteria is met, the switch can query the SCP for processing instructions. The SCP can respond with dialing plan collectibles. This functionality also applies to calls that query the SCP based on *O_Feature_Requested* or *Offhook_Delay* trigger criteria.

Figure 4-4 provides an example of the call flow for reorigination using the *O_IEC_Reorigination* where authcode, PIN, and address digits are to be collected. Notice that **`Connect_To_Resource`** and **`CTR_Clear`** conversational messages are used for ***O_Mid_Call*** processing.

For *Offhook_Delay*, Virtual IP conversational messaging is done using **`Send_To_Resource`** and **`Resource_Clear`** messages.

Figure 4-4
Virtual IP O_IEC_Reorigination trigger reorigination call flow


SSP
SCP
O_Mid_Call
Connect_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect PIN
– Play announcement and collect address
The switch processes instructions in the order received.
Virtual IP conversational messaging
CTR_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected PIN digits
– Collected address digits
Exit conversational messaging
Analyze_Route
Call is routed

Virtual IP support for TCAP queries

This feature is supported for all triggers (and their associated query messages) that support the **`Send_To_Resource`** or **`Connect_To_Resource`** response message and can receive an **`Analyze_Route`** in response, with the exception of the *Termination_Attempt* trigger at the ***Termination_Attempt*** TDP. *Termination_Attempt* is not supported because it occurs at a point in the call where new address or billing information cannot affect the routing of the call. Because subscription is not re-evaluated, CAIN does not support a **`Continue`** or a **`Collect_Information`** message returned after a Virtual IP interaction. If one is received, then it is a Fatal Application Error and a **`Report_Error`** message is returned.

Table 4-8 lists the NetworkBuilder detection points (DPs) and triggers that can receive `Send_To_Resource` and `Connect_To_Resource` response messages from the SCP.

Table 4-8
SCP Virtual IP response message

PIC	Detection Point	Trigger/Event	SCP VIP response
O_Null	***Origination_Attempt***	*Off_Hook_Immediate*	`Send_To_Resource`
Collect_ Information	***O_Feature_ Requested***	*O_Feature_ Requested*	`Send_To_Resource`
	Info_Collected	*Offhook_Delay*	
		Shared_Interoffice_ Trunk	
		PRI_B-Channel	
Analyze_ Information	***Info_Analyzed***	*Specific_Feature_ Code*	`Send_To_Resource`
		Customized_Dialing_ Plan	
		Specific_Digit_String	
Select_Route	***Network_Busy***	*Network_Busy* (trigger or event)	`Send_To_Resource`
Send_Call	***O_Called_Party_ Busy***	*O_Called_Party_Busy* (trigger or event)	`Send_To_Resource`
	O_Mid_Call	*O_IEC_Reorigination*	`Connect_To_Resource`
O_Alerting	***O_No_Answer***	*O_No_Answer* (trigger or event)	`Send_To_Resource`
	O_Mid_Call	*O_IEC_Reorigination*	`Connect_To_Resource`
O_Active	***O_Mid_Call***	*O_IEC_Reorigination*	`Connect_To_Resource`
O_Suspended	***O_Mid_Call***	*O_IEC_Reorigination*	`Connect_To_Resource`

—end—

Virtual IP error scenarios

Table 4-9 contains a brief overview of the handling of error cases by Virtual IP.

Table 4-9
Virtual IP error scenarios

Error	Virtual IP response
DestinationAddress is received in the **Send_To_Resource** or **Connect_To_Resource** message	Switch to IP Forwards SCP request to the designated IP for processing
VIP SOC CAIN0607 not ON	Switch to SCP **Resource_Clear** message; ***ClearCause*** = taskRefused
Send_To_Resource or **Connect_To_Resource** message is not received in a conversation package (RESP package instead)	Switch to SCP **Report_Error**message; ApplicationErrorString = ErrorCause; ErrorCause = unexpectedCommunication
Resource type of the **Send_To_Resource** or **Connect_To_Resource** message is not ***FlexParameterBlock***	Switch processing **Send_To_Resource** or **Connect_To_Resource** message is handled appropriately for the resource type
Encoding authority used to encode the ***FlexParameterBlock*** is not the 'NetworkBuilder' encoding authority	Switch to SCP **Resource_Clear** message; ***ClearCause*** = failure; ***FailureCause*** = applicationError
VIP tag is not present in the ***FlexParameterBlock***	Switch to SCP **Resource_Clear** message; ***ClearCause*** = taskRefused
Continue message received after a Virtual IP interaction	Switch to SCP **Report_Error**message; FatalApplicationError

—continued—

Table 4-9
Virtual IP error scenarios (continued)

Error	Virtual IP response
`Collect_Information` message received after a Virtual IP interaction	Switch to SCP **`Report_Error`**message; FatalApplicationError
A user misdials or does not enter enough digits.	Switch to SCP **`Resource_Clear`** message; ***`IPReturnBlock`*** will contain the digits that were dialed and will return a ***`ClearCause`*** = `invalidCode`.
A **`Cancel_Resource_Event`** message is received during collectible processing.	Switch to SCP **`Resource_Clear`** message; ***`IPReturnBlock`*** will contain the digits that were dialed and will return a ***`ClearCause`*** = `resourceCancelled`.
—end—	

Billing

Billing information for Virtual IP interactions is captured in the following CDR fields:

- BILLNUM: authcode and card
- DIALEDNO: address
- PINDIGS: PIN
- ACCTCD: account code
- CALLDUR:
  - — Virtual IP affects the CALLDUR field in the sense that it can optionally store the amount of time the UCS DMS-250 switch simulates an IP. If the UCS DMS-250 switch receives the ***`AMAMeasure`*** parameter in a **`Send_To_Resource`** or **`Connect_To_Resource`** message, the switch starts timing at this point in the call. The time spent processing the Virtual IP portion of the call is captured in the CALLDUR field of the CDR. If the ***`AMAMeasure`*** parameter is not received, the Virtual IP portion of the call is not included in the total call duration time.

> ***Note:*** If the FlexCDR template is used, the Virtual IP portion of the call can optionally be recorded in the CALLDUR field of the FlexCDR template.

Figures 4-5, 4-6, 4-7, 4-8, and 4-9 provide examples of how the call duration is recorded in various scenarios.

Figure 4-5
Call duration with AMAMeasure of connectTimeNotRecorded

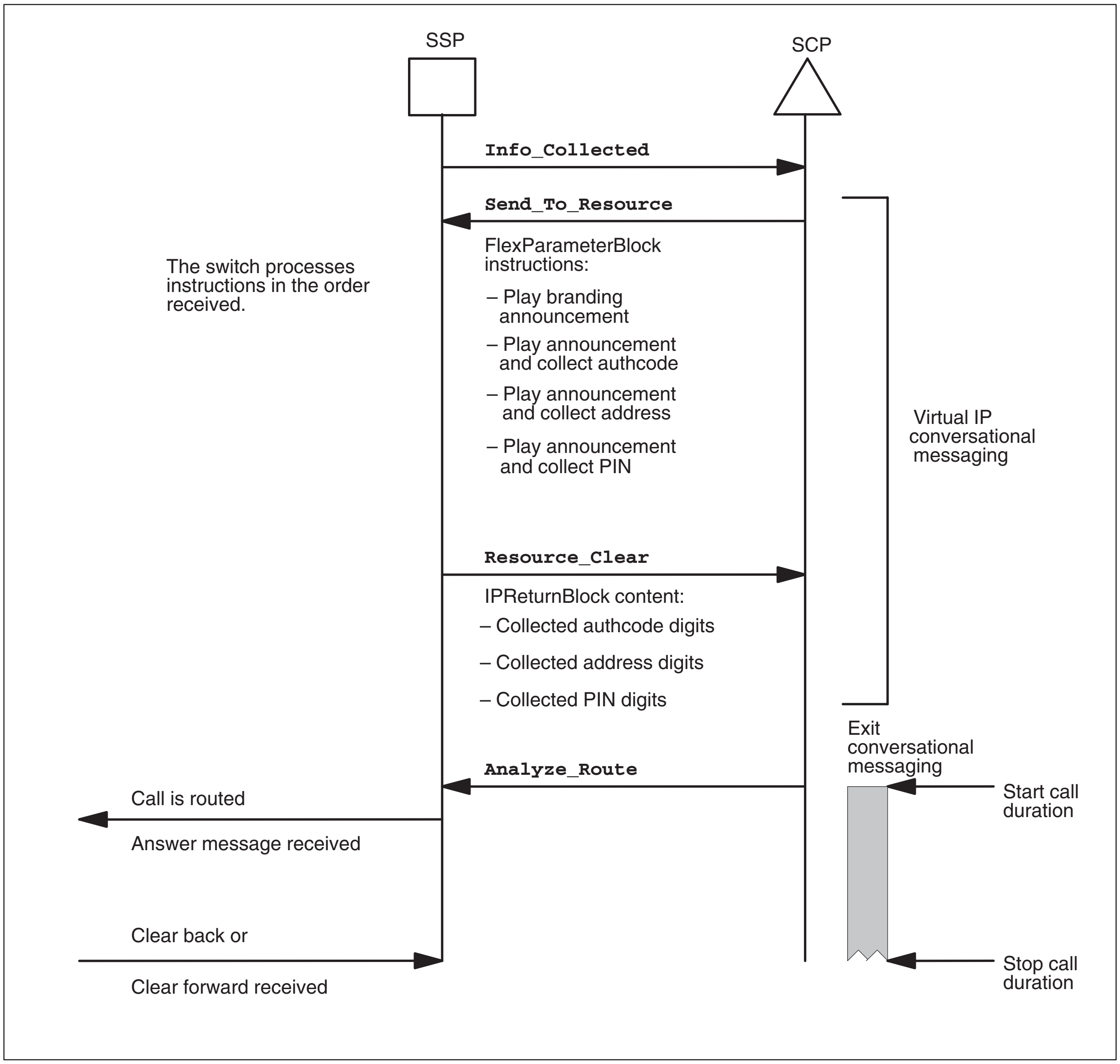

SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
The switch processes instructions in the order received.
Virtual IP conversational messaging
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Exit conversational messaging
Analyze_Route
Call is routed
Answer message received
Start call duration
Clear back or
Clear forward received
Stop call duration

Figure 4-6
Call duration with AMAMeasure of connectTimeRecorded DestinationSCP

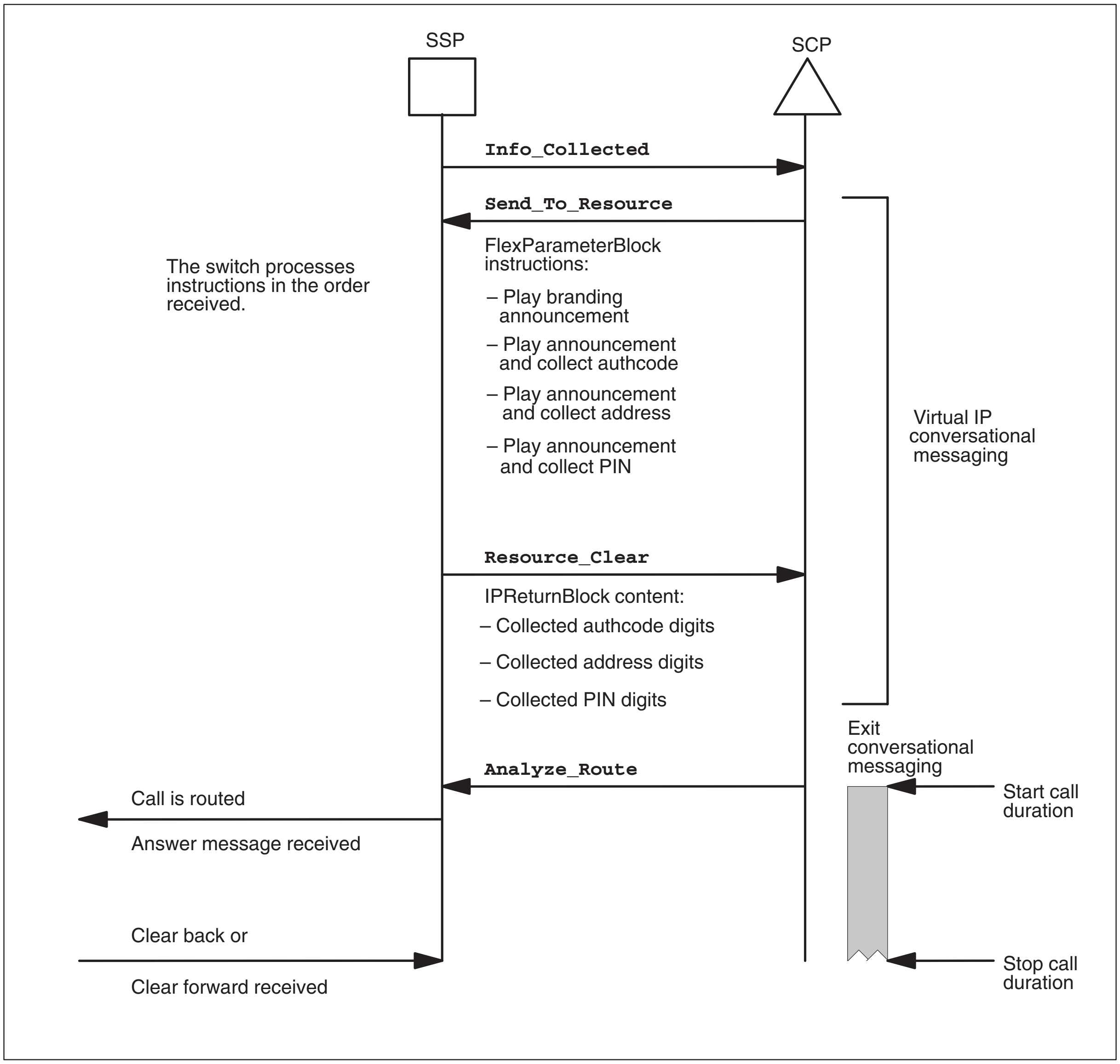

SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
The switch processes instructions in the order received.
Virtual IP conversational messaging
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Exit conversational messaging
Analyze_Route
Call is routed
Answer message received
Start call duration
Clear back or
Clear forward received
Stop call duration

Figure 4-7
Call duration with AMAMeasure of connectTimeRecordedDestinationSSP

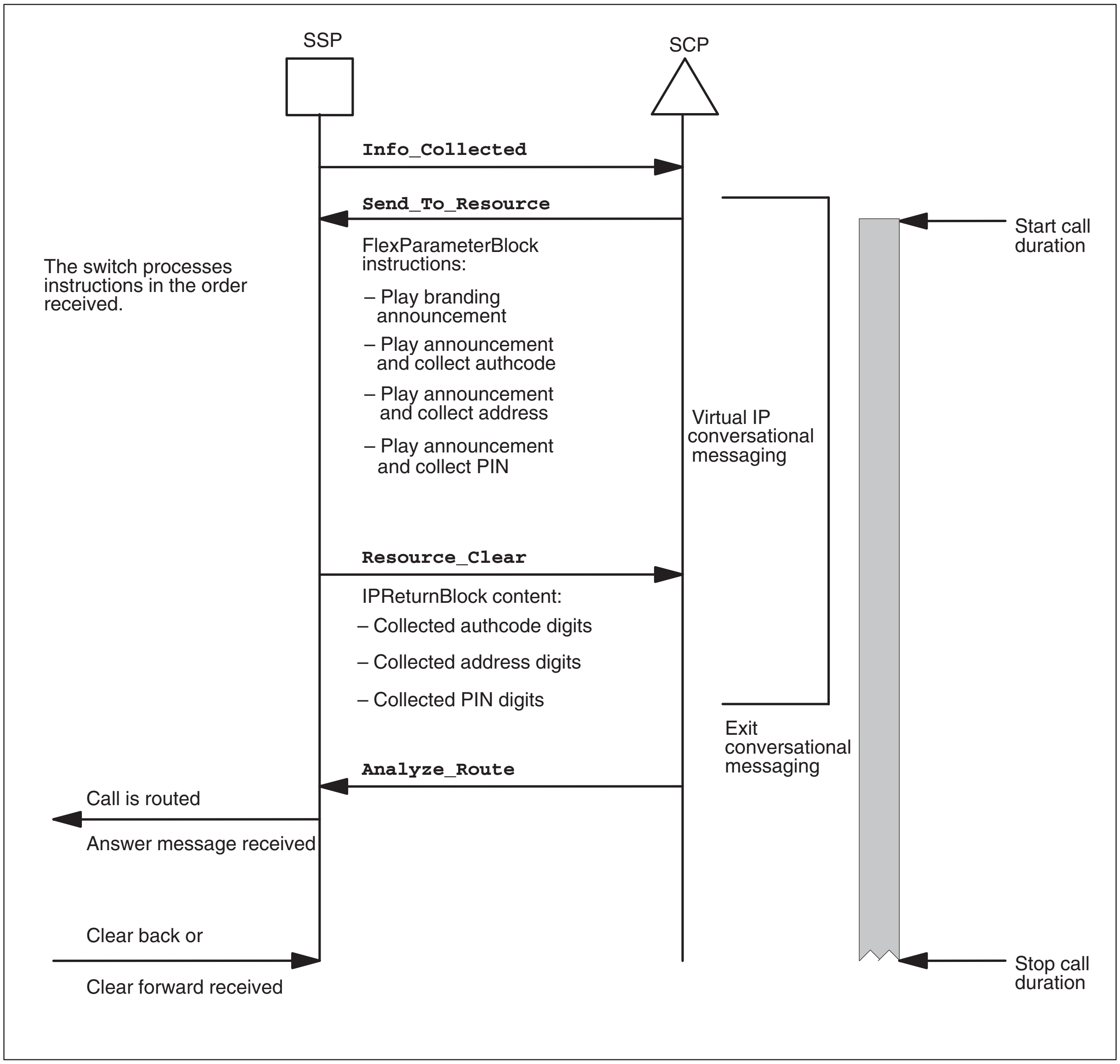

SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
The switch processes instructions in the order received.
Virtual IP conversational messaging
Start call duration
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Exit conversational messaging
Analyze_Route
Call is routed
Answer message received
Clear back or
Clear forward received
Stop call duration

Figure 4-8
Call duration for multiple Virtual IP interaction with first AMAMeasure of connectTimeRecordedDestinationSSP, followed by any AMAMeasure value (or no AMAMeasure value)

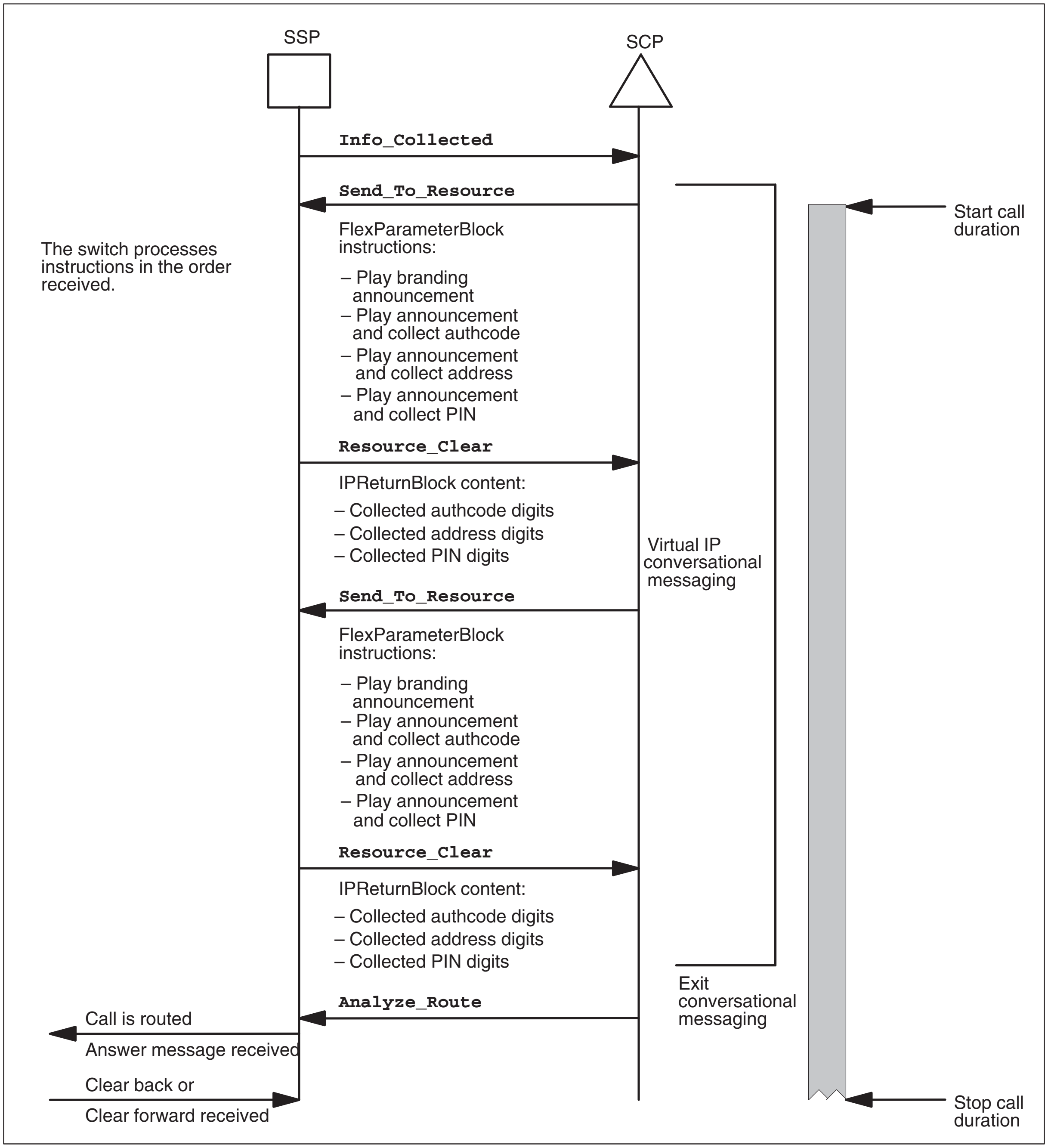
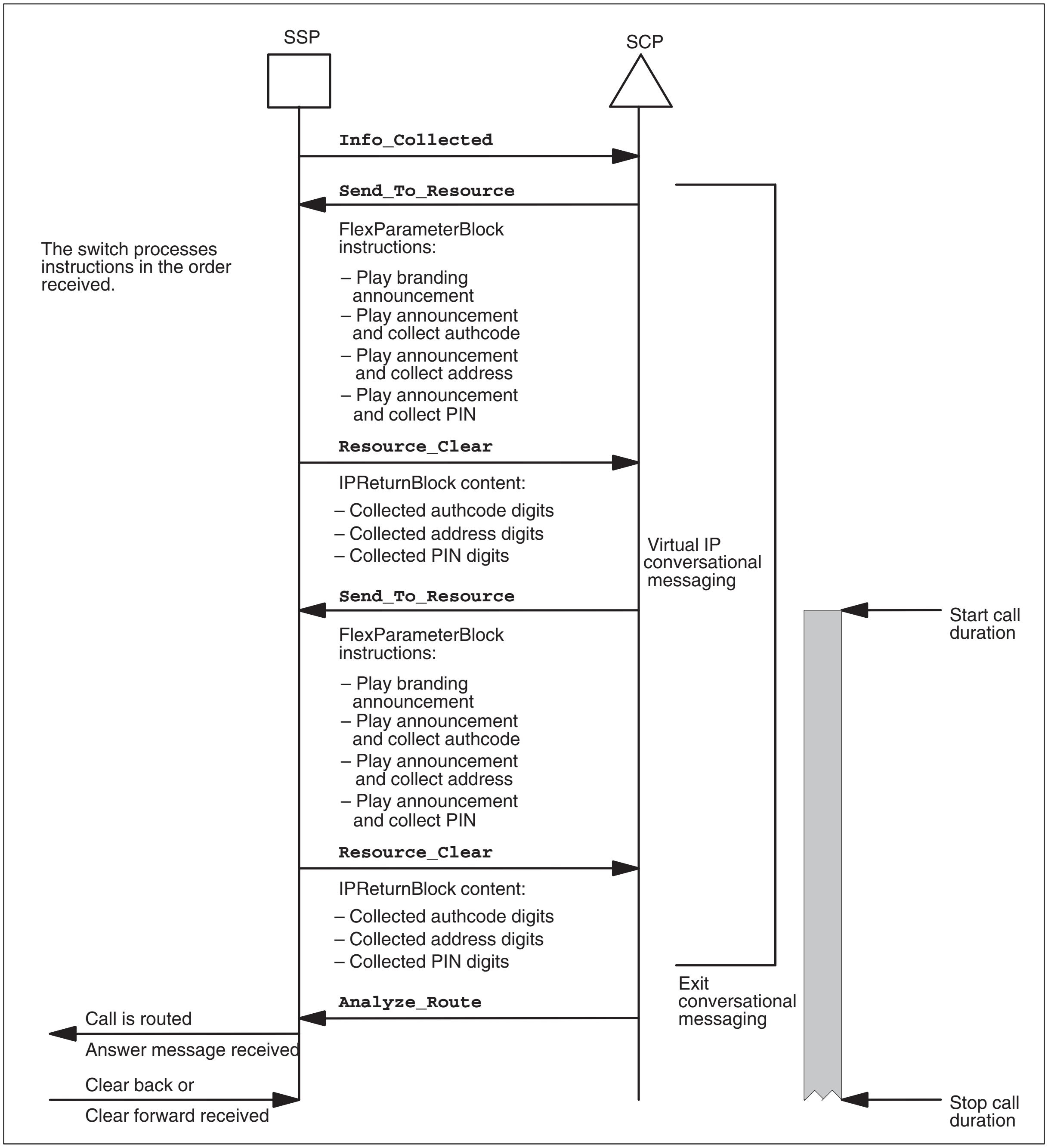

SSP
SCP
Info_Collected
Send_To_Resource
Start call duration
FlexParameterBlock instructions:
The switch processes instructions in the order received.
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Virtual IP conversational messaging
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Exit conversational messaging
Analyze_Route
Call is routed
Answer message received
Clear back or
Clear forward received
Stop call duration

Figure 4-9
Call duration with first Send_To_Resource or Connect_To_Resource with either no AMAMeasure or AMAMeasure of connectTimeNotRecorded or connectTimeRecordedDestinationSCP, followed by connectTimeRecordedDestinationSSP


SSP
SCP
Info_Collected
Send_To_Resource
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
The switch processes instructions in the order received.
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Virtual IP conversational messaging
Send_To_Resource
Start call duration
FlexParameterBlock instructions:
– Play branding announcement
– Play announcement and collect authcode
– Play announcement and collect address
– Play announcement and collect PIN
Resource_Clear
IPReturnBlock content:
– Collected authcode digits
– Collected address digits
– Collected PIN digits
Exit conversational messaging
Analyze_Route
Call is routed
Answer message received
Clear back or
Clear forward received
Stop call duration

Digital Switching Systems

UCS DMS-250

NetworkBuilder Application Guide,
Volume 4 of 5

Product Documentation–Dept 3423
Nortel Networks
P.O. Box 13010
RTP, NC 27709–3010
1–877-662-5669

NORTEL NETWORKS CONFIDENTIAL: The information contained herein is the property of Nortel Networks and is strictly confidential. Except as expressly authorized in writing by Nortel Networks, the holder shall keep all information contained herein confidential, shall disclose the information only to its employees with a need to know, and shall protect the information, in whole or in part, from disclosure and dissemination to third parties with the same degree of care it uses to protect its own confidential information, but with no less than reasonable care. Except as expressly authorized in writing by Nortel Networks, the holder is granted no rights to use the information contained herein.

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Publication number: 297-2621–370
Product release: UCS17
Document release: Standard 10.01
Date: July 2002
Printed in the United States of America

NORTEL
NETWORKS™

How the world shares ideas.