SIP +
(Inter MGC Protocol)

Edition 0.0
DRAFT 0.1

Last Modified December 4, 1998

Eric Zimmerer
+1-303-926-3142
eric.zimmerer@Level3.com

1 Overview

This document describes Inter Media Gateway Controller communication using SIP+. This protocol facilitates the exchange of information between an Originating Call-Half Controller (referred to as a Media Gateway Controller in this document) and a Terminating Call-Half Controller, such that the call may be completed. The space occupied by SIP+ is similar to the space occupied by H.323 Gatekeeper to Gatekeeper communications. SIP+ is based on SIP:Session Initiation Protocol [1] with extensions described in this document. SIP+ provides the methods for Media Gateway Controllers to set up, tear down and manage voice and data calls. These extensions make SIP+ flexible enough to allow for a variety of in bound and out bound signaling types including but not limited to SS7, ISDN, and CAS.

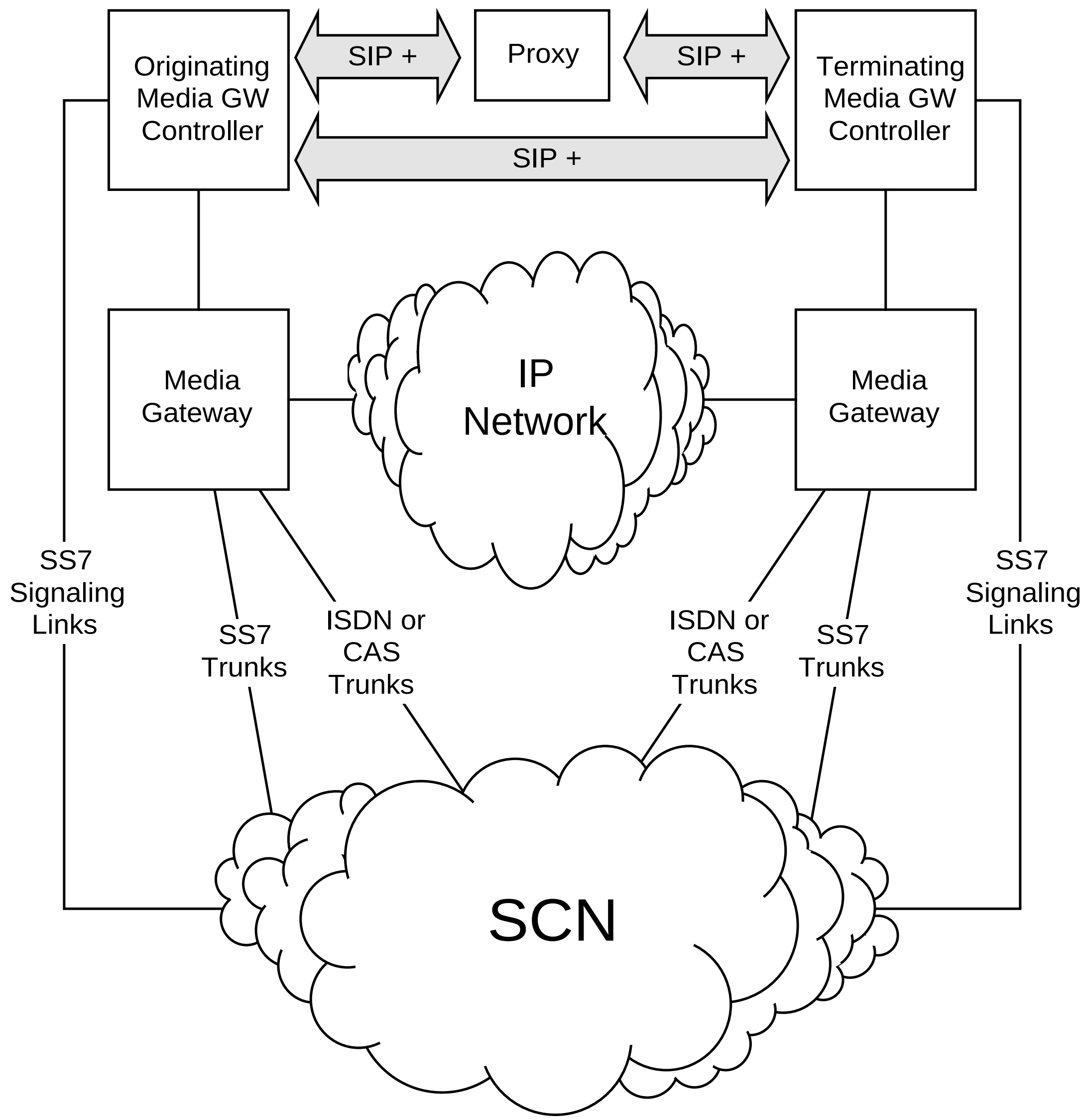


Figure 1 - Use of the SIP+ protocol

Figure 1 shows a basic network configuration using SIP+. In this example Media Gateways are connected to the Switched Circuit Network (SCN) via SS7 trunks, ISDN trunks, and CAS trunks. The Originating Media Gateway Controller (O_MGC) may receive a call over any of these trunks. The signaling information from these trunks must be processed by the O_MGC to establish the originating call half, and to determine the identity of the T_MGC required to complete the call. The O_MGC uses SIP+ to communicate the necessary information to the T_MGC to complete the call. The T_MGC must be able to establish the terminating call half on any of the supported trunk types.

2 Extensions to SIP

SIP must be extended for the Inter MGC Protocol application to handle PSTN signaling messages between the O_MGC and the T_MGC and the networks they connect to. The use of MIME [2] encoding with content type of APPLICATION allows PSTN signaling messages to be tunneled between Media Gateway Controllers. The use of content SUBTYPE enables ISUP messages, Q.931 messages, and other types of signaling messages to be differentiated by the receiving MGC. A basic mapping of ISUP messages to SIP+ messages has been done in the call flows below. With these extensions, SIP+ includes ISUP for compatibility with the PSTN, and has SDP [3] functionality for future VOIP services.

2.1 SIP+ Mapping of SS7 ISUP Messages

IAM = INVITE
ACM = 180 RINGING
ANM = 200 OK
RLS = BYE
RLC = 200 OK

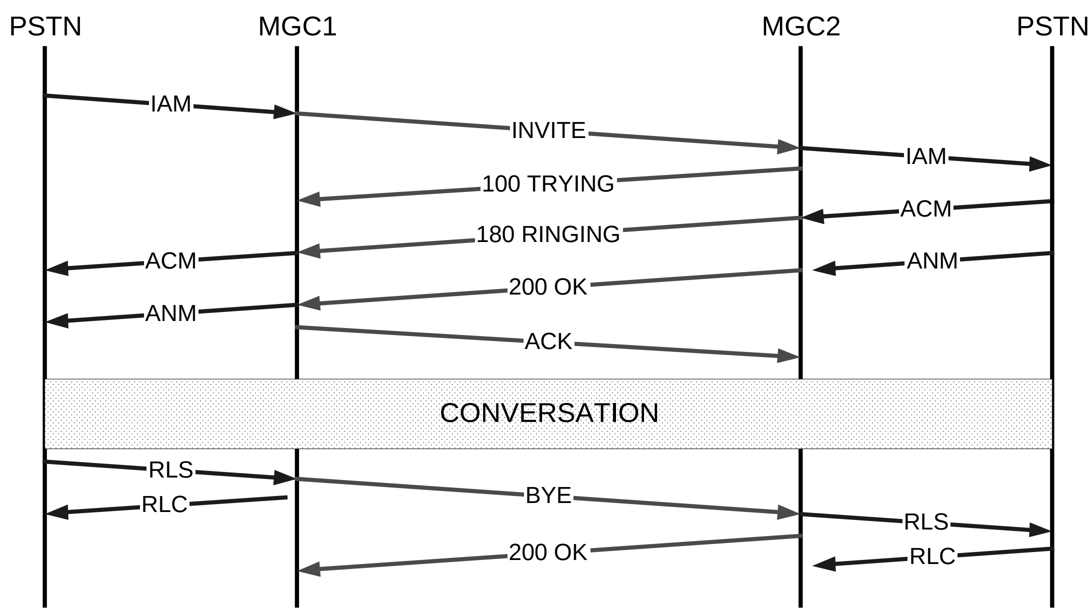


Figure 2 - SS7 Mapping to SIP+

2.2 SIP+ Mapping of Q.931 Messages

Under construction…

2.3 SIP+ Mapping of CAS Messages

Under construction…

2.4 MIME Encoding of PSTN Signaling messages

Extending SIP+ by encoding SS7 ISUP and other PSTN signaling messages allows MGCs using SIP+ to be compatible with the PSTN. SIP+ encodes and transmits the native signaling messages from one SCN to another. To do this, SIP+ has been extended with MIME encoding of signaling messages. The PSTN signaling messages are appended to the SIP+ messages (INVITE, ACK, BYE, CANCEL) using binary encoding. The use of MIME encoding with content type: APPLICATION allows PSTN signaling messages to be tunneled between Media Gateway Controllers. The use of content SUBTYPE enables ISUP messages, Q.931 messages, and other types of signaling messages to be differentiated by the receiving MGC.

An example of MIME encoding is when a call arrives on an SS7 trunk, the IAM is encoded and forwarded to the T_MGC as part of the SIP+ INVITE message body. The subsequent ACM and ANM messages are passed back to the O_MGC along with other necessary information via 180 RINGING, and 200 OK messages. For ISDN Q.931 messages (setup, connect, etc.) and other signaling types that the MGC must process, SIP+ encodes the messages in the same way. The SUBTYPE tells the receiving MGC which type of messaging has been encoded. This can be ISUP, Q931, CAS, etc.

If the native signaling message from the originating network is compatible with the terminating network, (i.e. ISUP to ISUP) the T_MGC simply modifies the message as necessary and sends it to the terminating network. If the native signaling message from the originating network is not compatible (i.e. CAS to ISUP), the T_MGC must generate appropriate signaling messages for the terminating network. However, protocol interworking is beyond the scope of SIP+.

3 SIP+ Usage

SIP+ 0.0 can be used to set up calls from the PSTN via IP network to the PSTN. SIP+ provides the messaging for Media Gateway Controllers to communicate all necessary information to collaborate on a call. Scaling up for larger networks is handled by the use of proxies to abstract network coverage areas. A proxy can be used to abstract coverage areas such as East Coast, West Coast, Northern-Europe, etc. SIP+ is also able to use proxies as Network to network interfaces. Peer networks can establish calls through a peer network's proxies, allowing each network to treat the other network as a Terminating MGC. The peer network proxy handles the routing of the INVITE requests to the correct MGC.

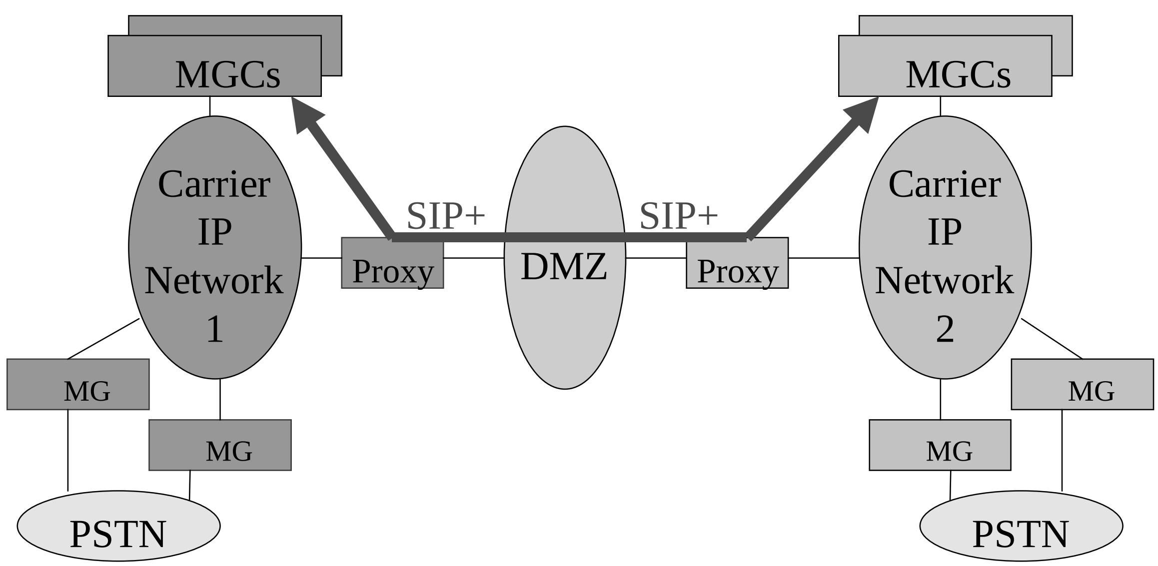
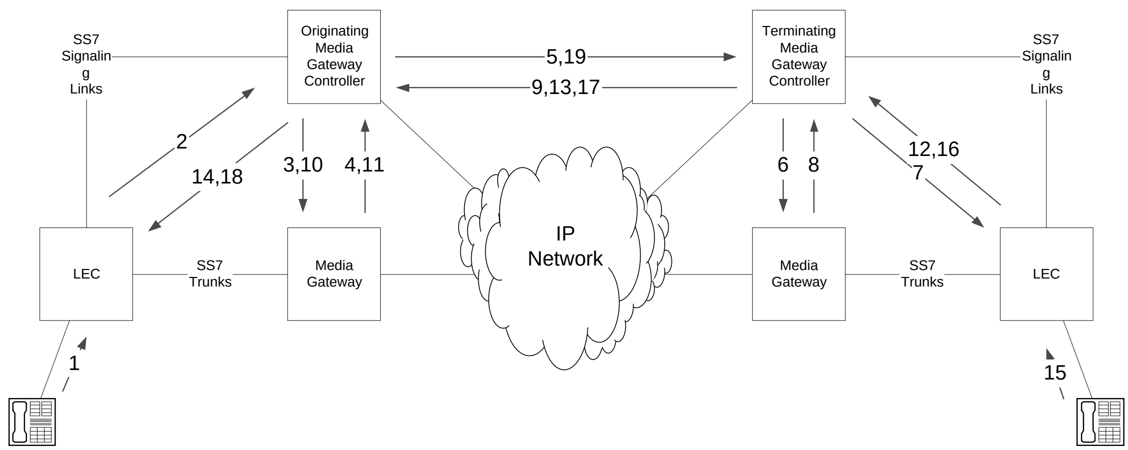


Figure 3 – Network to Network Interface

4 Call Flow Examples

The figure below represents a call from the PSTN via an IP network terminating back to the PSTN. The Media Gateways are controlled by Media Gateway Controllers using MGCP or IPDC. The call setup steps are shown as numbered arrows, which are detailed below. It is assumed that the Media Gateways are equipped to send and receive RTP packet streams.



-

Figure 4 - Basic Call Setup

1. Caller goes off hook
2. LEC switch sends SS7 IAM to O_MGC
3. O_MGC sends create connection command to O_MG
4. O_MG acknowledges the command and replies with O_MG RTP port number
5. O_MGC determines where call is destined for and sends INVITE to T_MGC with encoded IAM
6. T_MGC receives INVITE; sends create connection command to T_MG
7. T_MGC extracts IAM from INVITE, updates and forwards IAM to terminating network
8. T_MG establishes RTP connection to O_MG and sends connection acknowledge to T_MGC including T_MG RTP port information
9. T_MGC sends 100 TRYING to O_MGC including the RTP port information from the T_MG
10. O_MGC sends modify connection command to O_MG
11. O_MG establishes RTP connection to T_MG and sends connection acknowledge to O_MGC
12. LEC sends ACM to T_MGC
13. T_MGC sends 180 RINGING to O_MGC with encoded ACM
14. O_MGC receives 180 RINGING; extracts, updates and forwards ACM to originating LEC
15. Called Party answers phone
16. Terminating LEC sends ANM message to T_MGC
17. T_MGC sends 200 OK to O_MGC with encoded ANM
18. O_MGC receives 200 OK; extracts, updates and forwards ANM to Originating LEC
19. O_MGC sends ACK to T_MGC

4.1 Basic Call

For two-party VoIP phone call originating from an SS7 signaled trunk, transiting the IP network and terminating via another SS7 signaled trunk, the MGCs must communicate the location and description of the ports which will send and receive the IP data. In the example below, calling party A in Arvada calls

called party B in Boulder. The call begins on the PSTN, arrives at the Originating MGC (O_MGC) in Denver and is routed to the Terminating MGC (T_MGC) in Denver which then terminates the call to the PSTN. The O_MGC indicates it is equipped to receive RTP audio codings 0 (PCMU), 3 (GSM), and 4 (G.723).
In this example, Denver MGC #01 contacts Denver MGC #09 to set up a call on Media Gateway #122 port 9092 using PCMU, GSM or G.723. Denver MGC #09 responds that the call will be set up between Media Gateway #122 and Media Gateway #89. MG #89 will send audio data via RTP to MG #122 port 9092. MG #122 will send audio data via RTP to MG #89 port 4148 using PCMU. MG #89 will receive RTCP packets on port 4149 , and MG #122 will receive them on port 9093.
The call flow above illustrates the message order. Below are the detailed SIP+ messages from the call flow above.

Step 5: O_MGC sends INVITE to T_MGC with encoded IAM

```
INVITE sip:Denver_09.Denver.CarrierX.com SIP+/0.0
From: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
To: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 228
                /* number of bytes */
        v=0
                /* protocol version */
        o=- 199812150740 0 IN IP4 123.45.67.122
                /* username, session id, version, network type, address type, address */
        s=1998121507401234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_122.Denver.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 9092 RTP/AVP 0 3 4
                /* media type, transport port, transport protocol, media formats */
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
        <isup IAM encoded here>
```

Step 9: T_MGC sends 100 TRYING to O_MGC

```
SIP+/0.0 100 Trying
From: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
To: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 233
                /* length in bytes */
        v=0
                /* protocol version */
        o=- 199812150740 0 IN IP4 123.45.67.89
                /* username, session id, version, network type, address type, address */
        s=1998121507401234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_089.Denver.CarrierX.com
                /* network type, address type, originating MG address */
```

m=audio 4148 RTP/AVP 0
/* media type, transport port, transport protocol, media formats */

Step 13: T_MGC sends 180 RINGING to O_MGC with ACM

SIP+/0.0 180 Ringing
From: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
To: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ACM encoded here>

Note: If the terminating LEC returns the ACM in time for the T_MGC to include it in the 100 TRYING, it may be encoded in the 100 TRYING, eliminating the need to send the 180 RINGING.

Step 17: T_MGC sends 200 OK to O_MGC with ANM

SIP+/0.0 200 OK
From: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
To: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ANM encoded here>

Step 19: O_MGC sends ACK to T_MGC confirming the call

ACK sip:Denver_09.Denver.CarrierX.com SIP+/0.0
From: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
To: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 1 ACK

4.2 Tear Down

In this example, the call has already been set up as in 3.1. To terminate the call, either calling party or called party can go on hook. When the caller goes on hook, the MGC sends a BYE request. To continue the example call above, the calling party hangs up the phone and the O_MGC, Denver #01 sends a BYE request to the T_MGC Denver #09.

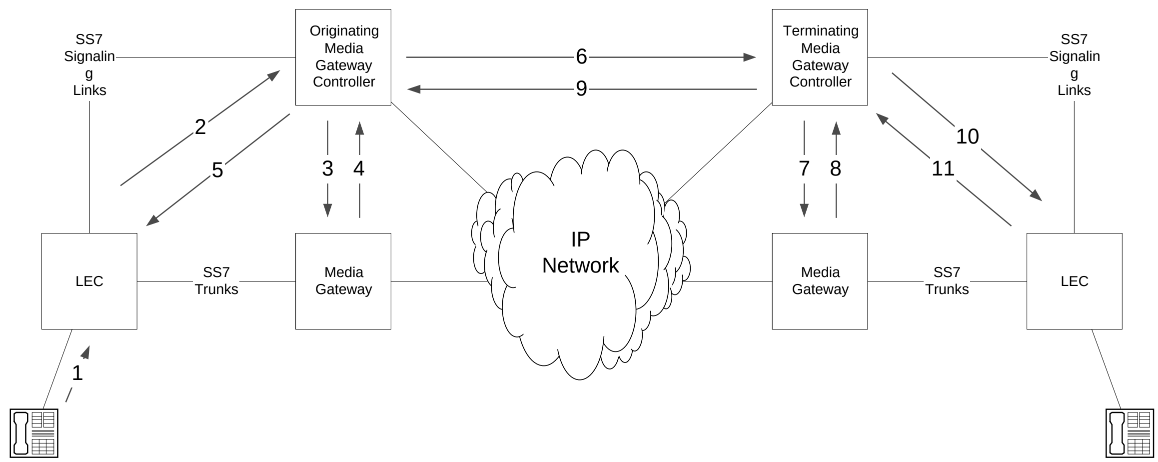


Figure 5 – Call Tear Down

1. Caller goes on hook (hangs up)
2. LEC sends Release message to O_MGC
3. O_MGC sends delete connection command to O_MG
4. O_MG sends delete connection ACK to O_MGC
5. O_MGC sends Release Complete to LEC
6. O_MGC sends BYE to T_MGC
7. T_MGC sends delete connection command to T_MG
8. T_MG sends delete connection ACK to T_MGC
9. T_MGC sends 200 OK to O_MGC
10. T_MGC sends Release message to LEC
11. LEC sends Release complete message to T_MGC

Step 6: O_MGC sends BYE to T_MGC:

```
BYE sip:Denver_09.Denver.CarrierX.com SIP+/0.0
From: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
To: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 2 BYE
```

Step 9: T_MGC sends 200 OK to O_MGC:

```
SIP+/0.0 200 OK
From: Denver_09 <sip:Denver_09.Denver.CarrierX.com>
To: Denver_01 <sip:Denver_01.Denver.CarrierX.com>
```

Call-ID: 1998122516401234@Denver_01.Denver.CarrierX.com
CSeq: 2 BYE

4.3 Ring No Answer

In this example, a call from New_York MGC #7 to New_York #3 has progressed up to step 16 in section 4.1, the receipt of ACM message at the originating LEC. Instead of the call being answered, however, there is no answer. In the case of Ring-no-answer, the calling party eventually hangs up the phone. When this happens, the call is released just as in 3.2 above.

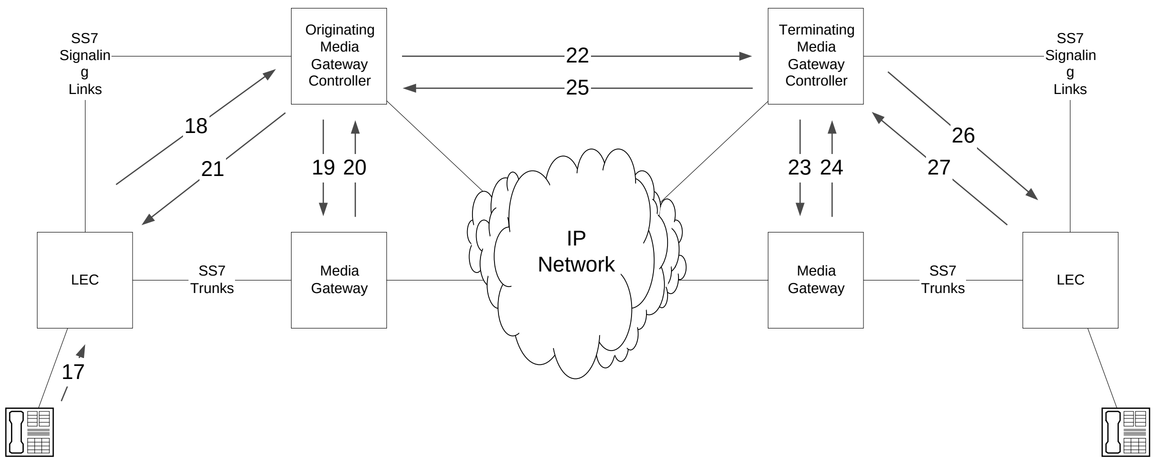


Figure 6 – Ring No Answer

17. Ring-No-Answer so calling party goes on hook
18. LEC sends Release message to O_MGC
19. O_MGC sends delete connection command to O_MG
20. O_MG sends delete connection ACK to O_MGC
21. O_MGC sends Release Complete to LEC
22. O_MGC sends BYE to T_MGC
23. T_MGC sends delete connection command to T_MG
24. T_MG sends delete connection ACK to T_MGC
25. T_MGC sends 200 OK to O_MGC
26. T_MGC sends Release message to LEC
27. LEC sends Release complete message to T_MGC

Step 22: O_MGC sends BYE to T_MGC:

BYE sip:New_York_03.New_York.CarrierX.com SIP+/0.0
From: New_York_07 <sip:New_York_07.New_York.CarrierX.com>
To: New_York_03 <sip:New_York_03.New_York.CarrierX.com>
Call-ID: 1998110509271234@New_York_07.New_York.CarrierX.com
CSeq: 2 BYE

Step 25: T_MGC sends 200 OK to O_MGC:

SIP+/0.0 200 OK

From: New_York_03 <sip:New_York_03.New_York.CarrierX.com>
To: New_York_07 <sip:New_York_07.New_York.CarrierX.com>
Call-ID: 1998110509271234@New_York_07.New_York.CarrierX.com
CSeq: 2 BYE

4.4 Call via Proxy

To illustrate how SIP+ may use a proxy, calling party A in Denver calls called party B in New York. The call begins on the PSTN, arrives at the Originating MGC (O_MGC) in Denver and is routed to the Terminating MGC (T_MGC) in New York via the New York proxy. The New York T_MGC then terminates the call to the PSTN.

The O_MGC indicates it is equipped to receive RTP audio codings 0 (PCMU), 3 (GSM), and 4 (G.723). In this example, Denver MGC #05 contacts New York MGC #11 to set up a call on Denver MG #124 port 7254 using PCMU, GSM or G.723. New York MGC #11 responds that the call will be set up between Denver MG #124 and New York MG #68. MG #68 will send audio data via RTP to MG #124 port 7254. MG #124 will send audio data via RTP to MG #68 port 5632 using PCMU. MG #68 will receive RTCP packets on port 5633 , and MG #124 will receive them on port 7255.

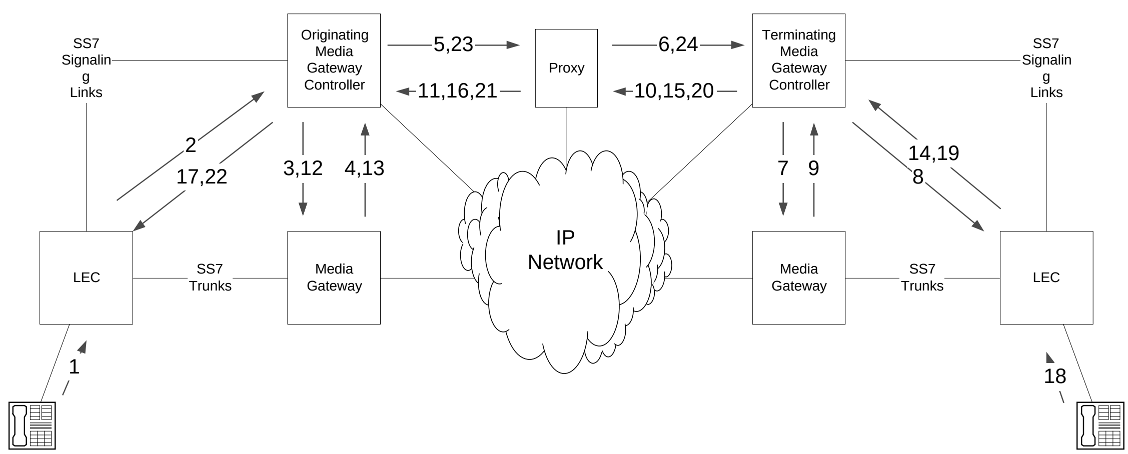


Figure 7 – Call Via Proxy

1. Caller goes off hook
2. LEC switch sends SS7 IAM to O_MGC
3. O_MGC sends create connection command to O_MG
4. O_MG acknowledges command with O_MG RTP port number
5. O_MGC determines where call is destined for and sends INVITE to T_MGC proxy with encoded IAM
6. T_MGC proxy forwards INVITE with encoded IAM to T_MGC
7. T_MGC receives INVITE; sends create connect command to T_MG
8. T_MGC extracts IAM from INVITE; updates and forwards IAM to terminating network
9. T_MG establishes RTP connection to O_MG and sends connection acknowledge to T_MGC including T_MG RTP port number
10. T_MGC sends 100 TRYING to T_MGC proxy including RTP information from the T_MG
11. T_MGC proxy sends 100 TRYING to O_MGC including RTP information from the T_MG
12. O_MGC sends modify connection command to O_MG
13. O_MG establishes RTP connection to T_MG and sends connection acknowledge to O_MGC
14. LEC sends ACM to T_MGC
15. T_MGC sends 180 RINGING to T_MGC proxy with encoded ACM
16. T_MGC proxy sends 180 RINGING to O_MGC with encoded ACM

17. O_MGC receives 180 RINGING; extracts, updates and forwards ACM to originating LEC
18. Called Party phone goes off hook
19. Terminating LEC sends ANM message to T_MGC
20. T_MGC sends 200 OK to T_MGC proxy with encoded ANM
21. T_MGC proxy sends 200 OK to O_MGC with encoded ANM
22. O_MGC receives 200 OK; extracts, updates and forwards ANM to Originating LEC
23. O_MGC sends ACK to T_MGC proxy
24. T_MGC proxy sends ACK to T_MGC

Step 5: O_MGC sends INVITE to T_MGC proxy with encoded IAM

```
INVITE sip:New_York.CarrierX.com SIP+/0.0
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York <sip:New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 248
                /* number of bytes */
        v=0
                /* protocol version */
        o=- 199809250815 0 IN IP4 123.45.67.124
                /* username, session id, version, network type, address type, address */
        s=1998092508151234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_124.Denver.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 7254 RTP/AVP 0 3 4
                /* media type, transport port, transport protocol, media formats */
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
        <isup IAM encoded here>
```

The INVITE to the New_York.CarrierX.com proxy specifies no MGC so that the proxy can substitute for the correct MGC in the SIP+ address.

Step 6: T_MGC proxy forwards INVITE with encoded IAM to T_MGC

```
INVITE sip:New_York_11.New_York.CarrierX.com SIP+/0.0  /*the proxy assigns MGC 11*/
Via: SIP+/0.0/UDP New_York.CarrierX.com
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York_11 <sip:New_York_11.New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 248
                /* number of bytes */
        v=0
                /* protocol version */
        o=- 199809250815 0 IN IP4 123.45.67.124
                /* username, session id, version, network type, address type, address */
        s=1998092508151234
                /* session name, aka call tag */
```

```
        c=IN IP4 Media_Gateway_124.Denver.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 7254 RTP/AVP 0 3 4
                /* media type, transport port, transport protocol, media formats */
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
        <isup IAM encoded here>
```

In this step, the New York Proxy assigns a specific T_MGC to handle the call. The assignment is based on configurable parameters at the Proxy.

Step 10: T_MGC sends 100 TRYING to T_MGC proxy

```
SIP+/0.0 100 Trying
Via: SIP+/0.0/UDP New_York.CarrierX.com
From: New_York_11 <sip:New_York_11.New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>  ###
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 255
                /* length in bytes */
        v=0
                /* protocol version */
        o=- 199809250815 0 IN IP4 198.76.54.68
                /* username, session id, version, network type, address type, address */
        s=1998092508151234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_068.New_York.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 5632 RTP/AVP 0
                /* media type, transport port, transport protocol, media formats */
```

Step 11: T_MGC proxy sends 100 TRYING to O_MGC

```
SIP+/0.0 100 Trying
From: New_York <sip:New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 255
                /* length in bytes */
        v=0
                /* protocol version */
        o=- 199809250815 0 IN IP4 198.76.54.68
                /* username, session id, version, network type, address type, address */
        s=1998092508151234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_068.New_York.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 5632 RTP/AVP 0
                /* media type, transport port, transport protocol, media formats */
```

Step 15: T_MGC sends 180 RINGING to T_MGC proxy with encoded ACM

SIP+/0.0 180 Ringing
Via: SIP+/0.0/UDP New_York.CarrierX.com
From: New_York_11 <sip:New_York_11.New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ACM encoded here>

Step 16: T_MGC proxy sends 180 RINGING to O_MGC with encoded ACM

SIP+/0.0 180 Ringing
From: New_York <sip:New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ACM encoded here>

Step 20: T_MGC sends 200 OK to T_MGC proxy with encoded ANM

SIP+/0.0 200 OK
Via: SIP+/0.0/UDP New_York.CarrierX.com
From: New_York_11 <sip:New_York_11.New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ANM encoded here>

Step 21: T_MGC proxy sends 200 OK to O_MGC with encoded ANM

SIP+/0.0 200 OK
From: New_York <sip:New_York.CarrierX.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ANM encoded here>

Step 23: O_MGC sends ACK to T_MGC proxy

ACK sip: New_York.CarrierX.com SIP+/0.0
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York <sip:New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com

CSeq: 2 ACK

Step 24: T_MGC proxy sends ACK to T_MGC

ACK sip: New_York_11.New_York.CarrierX.com SIP+/0.0
Via: SIP+/0.0/UDP New_York.CarrierX.com
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York_11 <sip:New_York_11.New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 2 ACK

4.5 Redirected Call

In the case of redirection, the MGC, or a proxy determines that it is not the correct controller for the terminating call half. The MGC or proxy replies to the INVITE with a 301 (moved permanently) or 302 (moved temporarily) message containing another MGC location where the call may be terminated.
In the previous example, if the New York Proxy had been alerted to the fact that called party B was no longer a the New York address, but was instead reachable in Boston, the New York Proxy could have responded to the initial INVITE as follows:

Step 6: O_MGC sends INVITE to T_MGC proxy with encoded IAM

INVITE sip:New_York.CarrierX.com SIP+/0.0
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York <sip:New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 248
…

Step 7: T_MGC proxy redirects INVITE to Boston proxy

SIP+/0.0 302 Moved temporarily
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: New_York <sip:New_York.CarrierX.com>
Call-ID: 1998092508151234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Location: sip:Boston.CarrierX.com

The O_MGC in Denver would then send an INVITE to the Boston T_MGC via the proxy at Boston.CarrierX.com.

4.6 Network to Network Call

The following call flow illustrates how SIP+ may be used between networks. Calling party A in Denver subscribed to CarrierX calls called party B in London who is serviced by ZCarrier. The call begins on the PSTN, arrives at the Originating MGC in Denver and is routed to the Terminating MGC in London via the London proxy. The London T_MGC then terminates the call to the PSTN.
The O_MGC indicates it is equipped to receive RTP audio codings 0 (PCMU) and 3 (GSM).
In this example, Denver MGC #05 contacts London MGC #21. The call is set up between Denver MG #42 port 6132 using PCMU and London MG #101 port 7686. The ZCarrier proxy will mask the actual IP addresses, substituting RTP ports on the Proxy. The O_MGC will have an RTP connection from CarrierX MG #42 IP address= 123.45.67.42 port 6132 to Zcarrier Proxy IP address= 226.123.22.22 port 6002. The T_MGC will have an RTP connection from Zcarrier MG 101 IP address= 226.123.111.101 port 7686 to Zcarrier Proxy IP address = 226.123.22.22 port 3002.

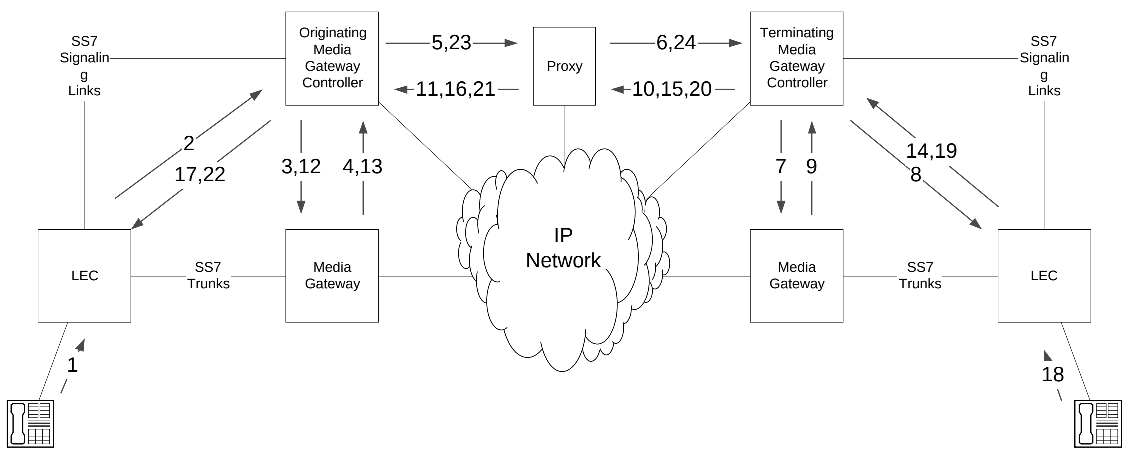


Figure 8 – Network to Network Call

1. Caller goes off hook
2. LEC switch sends SS7 IAM to O_MGC
3. O_MGC sends create connection command to O_MG
4. O_MG acknowledges command with O_MG RTP port number
5. O_MGC determines where call is destined for and sends INVITE to T_MGC proxy with encoded IAM
6. T_MGC proxy forwards INVITE with encoded IAM to T_MGC
7. T_MGC receives INVITE; sends create connect command to T_MG
8. T_MGC extracts IAM from INVITE; updates and forwards IAM to terminating network
9. T_MG establishes RTP connection to O_MG via the T_MGC proxy and sends connection acknowledge to T_MGC including T_MG RTP port number
10. T_MGC sends 100 TRYING to T_MGC proxy
11. T_MGC proxy sends 100 TRYING to O_MGC
12. O_MGC sends modify connection command to O_MG
13. O_MG establishes RTP connection to T_MG via the T_MGC Proxy and sends connection acknowledge to O_MGC
14. LEC sends ACM to T_MGC
15. T_MGC sends 180 RINGING to T_MGC proxy with encoded ACM
16. T_MGC proxy sends 180 RINGING to O_MGC with encoded ACM
17. O_MGC receives 180 RINGING; extracts, updates and forwards ACM to originating LEC

18. Called Party phone goes off hook
19. Terminating LEC sends ANM message to T_MGC
20. T_MGC sends 200 OK to T_MGC proxy with encoded ANM
21. T_MGC proxy sends 200 OK to O_MGC with encoded ANM
22. O_MGC receives 200 OK; extracts, updates and forwards ANM to Originating LEC
23. O_MGC sends ACK to T_MGC proxy
24. T_MGC proxy sends ACK to T_MGC

Step 5: O_MGC sends INVITE to T_MGC proxy with encoded IAM

```
INVITE sip:London.ZCarrier.com SIP+/0.0
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: London <sip:London.ZCarrier.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 248
                /* number of bytes */
        v=0
                /* protocol version */
        o=- 199902231900 0 IN IP4 123.45.67.42
                /* username, session id, version, network type, address type, address */
        s=1999022319001234
                /* session name, aka call tag */
        c=IN IP4 Media_Gateway_42.Denver.CarrierX.com
                /* network type, address type, originating MG address */
        m=audio 6132 RTP/AVP 0 3
                /* media type, transport port, transport protocol, media formats */
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
        <isup IAM encoded here>
```

The INVITE to the London.ZCarrier.com proxy specifies no MGC so that the proxy can substitute for the correct MGC in the SIP+ address.

Step 6: T_MGC proxy forwards INVITE with encoded IAM to T_MGC

```
INVITE sip:London_21.London.ZCarrier.com SIP+/0.0  /*the proxy assigns MGC 21*/
Via: SIP+/0.0/UDP London.ZCarrier.com
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: London_21 <sip:London_21.London.ZCarrier.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Subject: IAM
Content-Type: application/sdp
Content-Length: 248
                /* number of bytes */
        v=0
                /* protocol version */
        o=- 199902231900 0 IN IP4 226.123.22.22
                /* username, session id, version, network type, address type, address */
        s=1999022319001234
                /* session name, aka call tag */
        c=IN IP4 RTP_London.ZCarrierX.com
```

/* network type, address type, originating MG address */
m=audio 3002 RTP/AVP 0 3
/* media type, transport port, transport protocol, media formats */
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup IAM encoded here>

In this step, the London Proxy assigns a specific T_MGC and substitute RTP port to handle the call. The assignment is based on configurable parameters at the Proxy.

Step 10: T_MGC sends 100 TRYING to T_MGC proxy

SIP+/0.0 100 Trying
Via: SIP+/0.0/UDP London.ZCarrier.com
From: London_21 <sip:London_21.London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 255
/* length in bytes */
v=0
/* protocol version */
o=- 199902231900 0 IN IP4 226.123.111.101
/* username, session id, version, network type, address type, address */
s=1999022319001234
/* session name, aka call tag */
c=IN IP4 Media_Gateway_101.London.ZCarrier.com
/* network type, address type, originating MG address */
m=audio 7686 RTP/AVP 0
/* media type, transport port, transport protocol, media formats */

Step 11: T_MGC proxy sends 100 TRYING to O_MGC

SIP+/0.0 100 Trying
From: London <sip:London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 255
/* length in bytes */
v=0
/* protocol version */
o=- 199902231900 0 IN IP4 226.123.22.22
/* username, session id, version, network type, address type, address */
s=1999022319001234
/* session name, aka call tag */
c=IN IP4 RTP_London.ZCarrier.com
/* network type, address type, originating MG address */
m=audio 6002 RTP/AVP 0
/* media type, transport port, transport protocol, media formats */

Step 15: T_MGC sends 180 RINGING to T_MGC proxy with encoded ACM

SIP+/0.0 180 Ringing
Via: SIP+/0.0/UDP London.ZCarrier.com
From: London_21 <sip:London_21.London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ACM encoded here>

Step 16: T_MGC proxy sends 180 RINGING to O_MGC with encoded ACM

SIP+/0.0 180 Ringing
From: London <sip:London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ACM encoded here>

Step 20: T_MGC sends 200 OK to T_MGC proxy with encoded ANM

SIP+/0.0 200 OK
Via: SIP+/0.0/UDP London.ZCarrier.com
From: London_21 <sip:London_21.London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ANM encoded here>

Step 21: T_MGC proxy sends 200 OK to O_MGC with encoded ANM

SIP+/0.0 200 OK
From: London <sip:London.ZCarrier.com>
To: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 1 INVITE
MIME-Version: 1.0
Content-Type: application / isup
Content-Transfer-Encoding: binary
<isup ANM encoded here>

Step 23: O_MGC sends ACK to T_MGC proxy

ACK sip: London.ZCarrier.com SIP+/0.0
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: London <sip:London.ZCarrier.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 2 ACK

Step 24: T_MGC proxy sends ACK to T_MGC

ACK sip: London_21.London.ZCarrier.com SIP+/0.0
Via: SIP+/0.0/UDP London.ZCarrier.com
From: Denver_05 <sip:Denver_05.Denver.CarrierX.com>
To: London_21 <sip:London_21.London.ZCarrier.com>
Call-ID: 1999022319001234@Denver_05.Denver.CarrierX.com
CSeq: 2 ACK

4.7 COT Call Flow

Under construction…

4.8 ISDN Call Flow

Under construction…

4.9 CAS Call Flow

Under construction…

4.10 Network transit Call Flow

This example illustrates how a call may traverse a second carrier's network to terminate on a third carrier's MGC. Under construction…

4.11 H.323 Terminal Call Flow

Under construction…

5 Acknowledgments

Many people have contributed greatly to this document. In particular Ike Elliott, Andrew Dugan, John Wetherbie, Rich Terpstra, Jonathan Mitchell, Rob Maidhof, and Dent Steve Dent.

6 References

[1] Handley, Schulzrinne, Schooler and Rosenberg, ``Session Initiation Pro-
tocol (SIP)" Internet Draft, July 1997.

[2] N. Borenstein and N. Freed, "Multipurpose Internet Mail Extensions Part One: Format of Internet Message Bodies," RFC 2045, Internet Engineering Task Force, November 1996.

[3] M. Handley and V. Jacobson, "SDP: Session Description Protocol, " RFC 2327, Internet Engineering Task Force, April 1998