Internet Engineering Task Force
INTERNET DRAFT
<draft-huitema-megaco-mgcp-flows-01.txt>
January 20, 1999
Expires: June 18, 1999

Christian Huitema
Flemming Andreasen
Bellcore
Mauricio Arango
RSL COM
Prakash K
TELESOFT INC

Media Gateway Control Protocol (MGCP) Call Flows

Christian Huitema, Flemming Andreasen, Mauricio Arango, Prakash K

Status of this document

Abstract

The Media Gateway Control Protocol (MGCP) organizes the communication between a Media Gateway controller, or call agent, and a Media Gateway, e.g. a Voice over IP gateway or a Network Access Server. MGCP is defined in a companion document [1]. This document provides example of MGCP usage by providing a variety of call flows, in the case of telephony and network access servers.

Table of Contents Page

1. Introduction

In order to understand the way the MGCP interface will be used, we have described here several possible call flows between a TGW, which is a trunking gateway that implements MGCP, and an RGW, which is a residential gateway that implements MGCP, as well as several call flows describing how MGCP could be used to control a network access service. For each of these call flows it is assumed that the default event packages are as follows:

TGW	Trunk package
RGW	Line package
NAS	Network Access Server package

The diagrams also show a Common Database (CDB) that can be queried for authorization and routing information, and an Accounting Gateway (ACC) that collects accounting information at the start and the end of calls.

These diagrams are solely meant to exhibit the behavior of the MGCP, and to help understanding this protocol. They are not meant as a tutorial on the implementation of a Call Agent. They may very well include miscellaneous errors and imprecisions.

2. Internet Telephony Call Flows.

We present seven Internet Telephony call flows:

- A basic call between two "trunking gateways",

- A basic call from a "residential gateway" to a "trunking gateway",

- A basic call from a "trunking gateway" to a "residential gateway".

- A basic call from an R2 trunk in a TGW to an SS7 trunk in a TGW.

- A basic call from an ISDN trunk in a business gateway to an SS7 trunk in a TGW.

- A basic call with continuity test, from a "trunking gateway" to a "residential gateway".

- A "hairpin" connection between two endpoints on a trunking gateway, using regular call set-up procedures.

- A "hairpin" connection between two endpoints on a residential gateway, using accelerated procedures.

2.1. Connection from a TGW to another TGW

The figure below gives the flow that results in a connection between two trunking gateways.

```
 _____________________________________
| sw1|  SG1|  TGW1|   CA |  TGW2|  SG2|
|____|_____|______|______|______|_____|
| IAM|  -> |      |      |      |     |
|    |  IAM|  - - |  ->  |      |     |
|    |     |   <- |  CRCX|      |     |
|    |     |  ACK |  ->  |      |     |
|    |     |      |  CRCX|  ->  |     |
|    |     |      |    <-|  ACK |     |
|    |     |      |  IAM |  - - |  -> |
|    |     |      |      |      |  IAM|
|    |     |      |      |      |   <-|
|    |     |      |    <-|  - - |  ACM|
|    |     |   <- |  ACM |      |     |
|  <-|  - -|  ACM |      |      |     |
|    |  ...|      |      |      |     |
|    |     |      |      |      |   <-|
|    |     |      |    <-|  - - |  ACM|
|    |     |   <- |  MDCX|      |     |
|    |     |  ACK |  ->  |      |     |
|    |   <-|  - - |  ANM |      |     |
|  <-|  ANM|      |      |      |     |
| REL|  -> |      |      |      |     |
|    |  REL|  - - |  ->  |      |     |
|    |     |   <- |  DLCX|      |     |
|    |     |  Perf|  ->  |      |     |
|    |     |  data|      |      |     |
|    |   <-|  - - |  RLC |      |     |
|  <-|  RLC|      |      |      |     |
|    |     |      |  DLCX|  ->  |     |
|    |     |      |    <-|  Perf|     |
|    |     |      |      |  data|     |
|    |     |      |  REL |  - - |  -> |
|    |     |      |      |      |  REL|
|    |     |      |      |      |   <-|
|    |     |      |    <-|  - - |  RLC|
|____|_____|______|______|______|_____|
```

During these exchanges the MGCP is used by the Call Agent to control the two endpoints located on the two TGW.

The exchanges start with the arrival from the first switch (SW1) of an

SS7/ISUP "IAM" message, relayed by the signalling gateway to the Call Agent. The call agent performs the routing, and determines that the call will have to be relayed towards the second switch (SW2), using a trunk located on TGW2.

The call agent starts the exchange by seizing the endpoint referenced in the IAM message:

```
        CRCX 1204 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711
        M: recvonly
```

Upon reception of that command, the trunking gateway prepares a connection description.

```
        200 1204 OK
        I:FDE234C8

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0
```

The call agent, upon reception of this acknowledgement, sends a connection request to the trunking gateway, asking the gateway to reserve a trunk in the group connected to the second switch:

```
        CRCX 1205 trunk-group-2/$@tgw2.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        M: sendrecv

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0
```

The call agent selects a trunk in the group, and acknowledges the creation:

```
        200 1204 OK
        I:abc0

        v=0
        c=IN IP4 128.96.63.25
        m=audio 1296 RTP/AVP 0
```

The two commands have created a one way path, suitable for forwarding ring tones and announcements to the calling party. The call agent can now relay the call by sending an IAM message to the second switch. When the ACM is received, it is immediately relayed to the first switch.

After some time, the call is answered, and an ANM message is relayed from the second switch to the call agent. The call agent will first validate the call by asking the first end-point to place the connection in duplex mode:

```
MDCX 1206 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0
```

The trunking gateway executes and acknowledges the modification:

```
200 1206 OK
```

The call agent can now relay the ANM message to the calling switch, and both parties can talk.

At the end of the call, in our example, the calling party hangs up. The first switch sends a release message to the call agent, through the signalling gateway. The call agent releases the connection on the first endpoint:

```
DLCX 1207 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1217 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48
```

The call agent can now confirm the release of the trunk, sending an RLC message to the first switch.

In parallel, the call agent releases the connection to the second endpoint:

```
DLCX 1208 trunk-group-2/13@tgw2.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:abc0
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1218 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27,LA=48
```

After receiving the acknowledgement, the call agent can relay the release message to the second switch. The switch will in turn acknowledge the release.

2.2. Basic call, RGW to TGW

```
 _____________________________________________________________________
|  Usr  |    RGW   |       CA     |  CDB|  ACC|    TGW   |  SS7/|  CO |
|       |          |              |     |     |          |  ISUP|     |
|_______|__________|______________|_____|_____|__________|______|_____|
|       |     <-   |      RQNT    |     |     |          |      |     |
|       |    Ack   |       ->     |     |     |          |      |     |
|  Off  |          |              |     |     |          |      |     |
| -hook |   (local |              |     |     |          |      |     |
| (Dial |  action) |              |     |     |          |      |     |
| -tone)|          |              |     |     |          |      |     |
| Digits|   Notify |       ->     |     |     |          |      |     |
|       |     <-   |      Ack     |     |     |          |      |     |
| (pro- |     <-   |   CRCX+RQNT  |     |     |          |      |     |
| gress)|    Ack   |       ->     |     |     |          |      |     |
|       |          |     Query    |     |     |          |      |     |
|       |          |  (E.164 S,D) |  -> |     |          |      |     |
|       |          |       <-     |  IP |     |          |      |     |
|       |          |      CRCX    |  - -|  - -|     ->   |      |     |
|       |          |              |     |     |  (cut in)|      |     |
|       |          |       <-     |  - -|  - -|    ack   |      |     |
|       |          |      IAM     |  - -|  - -|    - -   |   -> |     |
|       |     <-   |     MDCX     |     |     |          |  IAM |  -> |
|       |    Ack   |       ->     |     |     |          |   <- |  ACM|
|       |          |       <-     |  - -|  - -|    - -   |  ACM |     |
|       |     <-   |  Notification|     |     |          |      |     |
|       |          |    Request   |     |     |          |      |     |
|       |    Ack   |       ->     |     |     |          |      |     |
|       |          |              |     |     |          |   <- |  ANM|
|       |          |       <-     |  - -|  - -|    - -   |  ANM |     |
|       |     <-   |   MDCX+RQNT  |     |     |          |      |     |
|       |    Ack   |       ->     |     |     |          |      |     |
|       |  (cut in)|   Call start |  - -|  -> |          |      |     |
|_______|__________|______________|_____|_____|__________|______|_____|
   ___________________________________________________________________
  |   Usr   |   RGW   |      CA      |  CDB|  ACC|   TGW |  SS7/|  CO |
  |         |         |              |     |     |       |  ISUP|     |
  |_________|_________|______________|_____|_____|_______|______|_____|
  |         |         |              |     |     |       |   <- |  REL|
  |         |         |      <-      |  - -|  - -|  - -  |  REL |     |
  |         |    <-   |    Delete    |     |     |       |      |     |
  |         |         |  Connection  |     |     |       |      |     |
  |         |         |    Delete    |     |     |       |      |     |
  |         |         |  Connection  |  - -|  - -|   ->  |      |     |
  |         |   Perf  |              |     |     |       |      |     |
  |         |   Data  |      ->      |     |     |       |      |     |
  |         |         |      <-      |  - -|  - -|  perf |      |     |
  |         |         |              |     |     |  data |      |     |
  |         |         |   Call end   |  - -| ->  |       |      |     |
  | On-hook |  Notify |      ->      |     |     |       |      |     |
  |         |    <-   |      Ack     |     |     |       |      |     |
  |         |    <-   | Notification |     |     |       |      |     |
  |         |         |    Request   |     |     |       |      |     |
  |         |   Ack   |      ->      |     |     |       |      |     |
  |_________|_________|______________|_____|_____|_______|______|_____|
```

During these exchanges the MGCP is used by the Call Agent to control both the TGW and the residential gateway. The exchanges occur on two sides.

The first command is a NotificationRequest, sent by the Call Agent to the residential gateway. The request will consist of the following lines:

```
RQNT 1201 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hd(E(dl;hu, D/[0-9#*T](D);)
D: (0T|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
S:
```

That transaction illustrates the power of the "embedded" action. The gateway is instructed to look for an off-hook event, and, upon its detection, to provide a dial-tone and to start looking for DTMF digits. The gateway immediately acknowledges the command, repeating in the acknowledgement message the transaction id that the Call Agent attached to the query.

```
200 1201 OK
```

When the off hook event is noticed, the gateway provides the dial tone to the line (the delay between off-hook and dialtone is thus minimal.)

The gateway will start accumulating digits according to that digit map. When it has noticed a sufficient set of values, it will notify the observed string to the Call Agent:

```
NTFY 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: 912018294266
```

The Call Agent immediately acknowledges that notification.

```
200 2002 OK
```

The Call Agent will then seize the incoming circuit, creating a connection. The create connection commands piggybacks a notification request, to stop collecting digits yet continue watch for an on-hook transition:

```
CRCX 1204 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: recvonly
X: D/0123456789AD
R: hu
S:
```

> We should note at this point that the call agent could send the acknowledgement of the Notify and the CRCX in a single UDP packet, as explained in the "piggy backing" section of [1]. There are many possible groupings of that nature in the various examples.

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data:

```
200 1204 OK
I:FDE234C8

v=0
c=IN IP4 128.96.41.1
```

```
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The SDP announcement, in our example, specifies the address at which the gateway is ready to receive audio data (128.96.41.1), the transport protocol (RTP), the RTP port (3456) and the audio profile (AVP). The audio profile refers to RFC 1890, which defines that the payload type 0 has been assigned for G.711 transmission. The gateway is also ready to use ADPCM encoding at 32 kbps (G.726 -4). There is no standard payload type associated to ADPCM, so the gateway mentions its readiness to use a non standard payload associated to the dynamic type 96. The "rtpmap" attribute entry associates the payload type 96 to G726-32/4.

The Call Agent, having seized the incoming trunk and completed a routing look up to identify the outgoing gateway, must now seize the outgoing trunk. It does so by sending a connection command to the e-gress gateway:

```
CRCX 1205 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: sendrecv

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The CreateConnection command has the same parameters as the command sent to the ingress gateway, with two differences:

- The EndpointId points towards the outgoing trunk,

- The message carries the session description returned by the ingress gateway,

- Because the session description is present, the "mode" parameter is set to "send/receive".

We observe that the call identifier is identical for the two connections. This is normal: the two connections belong to the same call, which has a global identifier in our system.

The trunking gateway will acknowledge the connection command, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1205 OK
I:32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The Call Agent will relay the information to the ingress gateway, using a ModifyConnection command:

```
MDCX 1206 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: recvonly

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The residential gateway immediately acknowledges the modification:

```
200 1206 OK
```

At this stage, the Call Agent has established a half duplex transmission path. The phone attached to the residential gateway will be able to receive the signals, such as tones or announcements, that the remote switch may send through the trunking gateway.

When the call progresses, the Call Agent will receive from the remote switch progress messages, for example an "address complete" message (ACM). The Call Agent will analyze the message to determine whether signal are transmitted in band. If this is not the case, the Call Agent will instruct the RGW to generate ringing tones by sending a NotificationRequest:

```
RQNT 1207 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AE
R: hu
S: v
```

The gateway immediately acknowledges the command:

```
200 1207 OK
```

After the called user answers the call, the Call Agent will receive an answering message (ANM) from the CO switch. At that point, it will send a ModifyConnection command, to the residential gateway, to place the connection in full duplex mode. The command embeds NotificationRequest to stop the ringing tones.

```
MDCX 1209 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv
X: 0123456789AF
R: hu
S:
```

The residential gateway will acknowledge this command:

```
200 1209 OK
```

At this point, the connection is established.

When the Call Agent receives the REL message from the CO switch, it will have to tear down the call. It will do so by sending to both gateways a DeleteConnection command:

```
DLCX 1210 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8

DLCX 1211 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:32F345E2
```

The gateways will respond with acknowledgements that should include a "call parameters" header fields:

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48

250 1211 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27,LA=48
```

At this point, the phone attached to the residential gateway, in our scenario, goes on-hook. This event is notified to the Call Agent, according to the policy received in the last NotificationRequest by sending a Notify command:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AF
O: hu
```

After this notification, the Call Agent should send an acknowledgement:

```
200 2005 OK
```

It should then issue a new NotificationRequest, to be ready to receive the next off-hook detected by the residential gateway:

```
RQNT 1212 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B0
R: hd(E(dl;hu, [0-9#*T](D);)
D: (0T|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
S:
```

The gateway will acknowledge this message:

```
200 1212 OK
```

Both gateways, at this point, are ready for the next call.

2.3. Basic call, TGW to RGW

```
 __________________________________________________________________
| CO |  SS7/|    TGW   |       CA      |  CDB|  ACC|   RGW  |  Usr |
|    |  ISUP|          |               |     |     |        |      |
|____|______|__________|_______________|_____|_____|________|______|
| IAM|   -> |          |               |     |     |        |      |
|    |  IAM |    - -   |       ->      |     |     |        |      |
|    |      |          |      Check    |  -> |     |        |      |
|    |      |          |       <-      |  IP |     |        |      |
|    |      |     <-   |     Create    |     |     |        |      |
|    |      |          |   Connection  |     |     |        |      |
|    |      |    Ack   |       ->      |     |     |        |      |
|    |      |          |     Create    |     |     |        |      |
|    |      |          |   Connection  |  - -|  - -|    ->  |      |
|    |      |          |       <-      |  - -|  - -|   Ack  |      |
|    |      |     <-   |     Modify    |     |     |        |      |
|    |      |          |   Connection  |     |     |        |      |
|    |      |    Ack   |       ->      |     |     |        |      |
|    |      |          |  Notification |     |     |        |      |
|    |      |          |     Request   |  - -|  - -|    ->  |  ring|
|    |      |          |       <-      |  - -|  - -|   Ack  |      |
|    |   <- |    - -   |       ACM     |     |     |        |      |
| <- |  ACM |          |               |     |     |        |      |
|    |      |          |               |     |     |        |  off |
|    |      |          |               |     |     |        |  hook|
|    |      |          |       <-      |  - -|  - -|  Notify|      |
|    |      |          |       Ack     |  - -|  - -|    ->  |      |
|    |      |          |  Notification |     |     |        |      |
|    |      |          |     Request   |  - -|  - -|    ->  |      |
|    |      |          |       <-      |  - -|  - -|   Ack  |      |
|    |      |     <-   |    Modify     |     |     |        |      |
|    |      |          |   Connection  |     |     |        |      |
|    |      |    Ack   |       ->      |     |     |        |      |
|    |      |  (cut-in)|   Call start  |  - -|  -> |        |      |
|    |   <- |    - -   |       ANM     |     |     |        |      |
| <- |  ANM |          |               |     |     |        |      |
|____|______|__________|_______________|_____|_____|________|______|
    ____________________________________________________________
   | CO|  SS7/|  TGW |      CA      |  CDB|  ACC|  RGW   |  Usr |
   |   |  ISUP|      |              |     |     |        |      |
   |___|______|______|______________|_____|_____|________|______|
   |   |      |      |              |     |     |        |  on  |
   |   |      |      |              |     |     |        |  hook|
   |   |      |      |      <-      |  - -|  - -|  Notify|      |
   |   |      |      |      Ack     |  - -|  - -|    ->  |      |
   |   |      |      |    Delete    |     |     |        |      |
   |   |      |      |   Connection |  - -|  - -|    ->  |      |
   |   |      |   <- |    Delete    |     |     |        |      |
   |   |      |      |   Connection |     |     |        |      |
   |   |   <- |  - - |      REL     |     |     |        |      |
   | <-|  REL |      |              |     |     |        |      |
   |   |      |  Perf|              |     |     |        |      |
   |   |      |  data|      ->      |     |     |        |      |
   |   |      |      |      <-      |  - -|  - -|  perf  |      |
   |   |      |      |              |     |     |   data |      |
   |   |      |      |    Call end  |  - -|  -> |        |      |
   |   |      |      |  Notification|     |     |        |      |
   |   |      |      |    Request   |  - -|  - -|    ->  |      |
   |   |      |      |      <-      |  - -|  - -|   Ack  |      |
   |___|______|______|______________|_____|_____|________|______|
```

This diagram shows the various exchange of messages during a call from a telephone user on the circuit-switched PSTN to a residential user connected to a residential gateway. During these exchanges the Call Agent uses MGCP to control both the TGW and the residential gateway. The exchanges occur on two sides.

Upon reception of the IAM message, the Call Agent immediately sends a CreateConnection request to the trunking gateway to connect to the incoming trunk, creating a connection:

```
        CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: recvonly
```

The trunking gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data:

```
        200 1237 OK
        I: FDE234C8

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The SDP announcement, in our example, specifies the address at which the gateway is ready to receive audio data (128.96.41.1), the transport protocol (RTP), the RTP port (3456) and the audio profile (AVP). The audio profile refers to RFC 1890, which defines that the payload type 0 has been assigned for G.711 transmission. The gateway is also ready to use ADPCM encoding at 32 kbps (G.726 -4). There is no standard payload type associated to ADPCM, so the gateway mentions its readiness to use a non standard payload associated to the dynamic type 96. The "rtpmap" attribute entry associates the payload type 96 to G726/4.

The Call Agent, having seized the incoming trunk, must now reserve the outgoing circuit. It does so by sending a connection command to the residential gateway:

```
        CRCX 1238 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: sendrecv

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The CreateConnection command has the same parameters as the command sent to the ingress gateway, with two differences:

* The EndpointId points towards the outgoing trunk,

* The message carries the session description returned by the ingress gateway,

* Because the session description is present, the "mode" parameter is set to "send/receive".

We observe that the call identifier is identical for the two connections. This is normal: the two connection belong to the same call, which has a global identifier in our system.

The residential gateway will acknowledge the connection command, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1238 OK
I:32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The Call Agent will relay the information to the ingress gateway, using a ModifyConnection command:

```
MDCX 1239 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: recvonly

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The trunking gateway immediately acknowledges the modification:

```
200 1239 OK
```

At this stage, the Call Agent has established a half-duplex transmission path. The Call Agent must now tells the residential gateway to ring the called line. It will send a NotificationRequest, consisting of the following lines:

```
RQNT 1240 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B1
R: hd
S: rg
```

The residential gateway, at that point, is instructed to look for an off-hook event, and to report it. It will first acknowledge the command, repeating in the acknowledgement message the transaction id that the Call Agent attached to the query.

```
        200 1240 OK
```

Upon reception of this message, the Call Agent sends an address complete message (ACM) to the calling switch, which we assume will generate ringing tones for the calling user.

When the gateway notices the off hook event, it sends a Notify command to the Call Agent:

```
        NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        X: 0123456789B0
        O: hd
```

The Call Agent immediately acknowledges that notification.

```
        200 2001 OK
```

The Call Agent now asks the residential gateway to send a Notify command on the occurrence of an on-hook event. It does so by sending a NotificationRequest to the residential gateway:

```
        RQNT 1241 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        X: 0123456789B1
        R: hu
```

The gateway acknowledges that command:

```
200 1241 OK
```

In parallel, the Call Agent will send a ModifyConnection command to the trunking gateway, to place the connection in full duplex mode:

```
        MDCX 1242 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:FDE234C8
        M: sendrecv
```

The trunking gateway will acknowledge that command:

```
200 1242 OK
```

The Call Agent can now send an answer message (ANM) to the calling switch.

After some time, the Call Agent will have to tear down the call. In our example, this is triggered by the residential user, who hangs up. The Notify command is sent to the Call Agent:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B1
O: hu
```

The Call Agent acknowledges the notification.

```
200 2005 OK
```

It will then send to both gateways a DeleteConnection command:

```
DLCX 1243 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:32F345E2

DLCX 1244 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
```

The gateways will respond with a message that should include a "call parameters" header fields:

```
250 1243 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48

250 1244 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27, LA=48
```

The Call Agent should now issue a new NotificationRequest to the residential gateway to detect the next off-hook event:

```
RQNT 1245 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B2
R: hd
```

The residential gateway will acknowledge this command:

```
200 1245 OK
```

Both gateways, at this point, are ready for the next call.

2.4. Basic call from an R2 trunk in a TGW to an SS7 trunk in a TGW

```
 ________________________________________________________________
|   CO   |    R2     |      CA       |    TGW    |   SS7/|  CO  |
|        |    TGW    |               |           |   ISUP|      |
|________|___________|_______________|___________|_______|______|
|        |    <-     |     NTRQ      |           |       |      |
|        |    Ack    |      ->       |           |       |      |
| Trunk  |           |               |           |       |      |
| seizure|           |               |           |       |      |
|    &   |           |               |           |       |      |
| Called |           |               |           |       |      |
| &callng|           |               |           |       |      |
| number |           |               |           |       |      |
| digit  |           |               |           |       |      |
| collec-|           |               |           |       |      |
|  tion  |    ->     |               |           |       |      |
|        |   Notify  |      ->       |           |       |      |
|        |    <-     |      Ack      |           |       |      |
|        |    <-     |   Notification|           |       |      |
|        |           |     Request   |           |       |      |
|        |    Ack    |      ->       |           |       |      |
|        |    <-     |     Create    |           |       |      |
|        |           |   Connection  |           |       |      |
|        |    Ack    |      ->       |           |       |      |
|        |           |     Create    |           |       |      |
|        |           |   Connection  |    ->     |       |      |
|        |           |               |   (cut in)|       |      |
|        |           |      <-       |    ack    |       |      |
|        |           |      IAM      |    - -    |   ->  |      |
|        |    <-     |     Modify    |           |   IAM |  ->  |
|        |           |   Connection  |           |       |      |
|        |    Ack    |      ->       |           |   <-  |   ACM|
|        |           |      <-       |    - -    |   ACM |      |
|        |           |               |           |   <-  |   ANM|
|        |           |      <-       |    - -    |   ANM |      |
|        |    <-     |     Modify    |           |       |      |
|        |           |   Connection  |           |       |      |
|        |    Ack    |      ->       |           |       |      |
|        |   (cut in)|               |           |       |      |
|________|___________|_______________|___________|_______|______|
  ____________________________________________________________
 |    CO   |    R2    |      CA       |   TGW  |   SS7/|   CO |
 |         |    TGW   |               |        |   ISUP|      |
 |_________|__________|_______________|________|_______|______|
 |         |          |               |        |   <-  |   REL|
 |         |          |      <-       |   - -  |   REL |      |
 |         |          |    Delete     |        |       |      |
 |         |    <-    |   Connection  |        |       |      |
 |         |   Clear- |               |        |       |      |
 |    <-   |    back  |    Delete     |        |       |      |
 |  Clear- |          |   Connection  |    ->  |       |      |
 |   fwd   |    ->    |               |        |       |      |
 |         |    Rel-  |               |        |       |      |
 |    <-   |   guard  |               |        |       |      |
 |         |    Perf  |               |        |       |      |
 |         |    Data  |      ->       |        |       |      |
 |         |          |      <-       |   Perf |       |      |
 |         |          |               |   data |       |      |
 |         |    <-    |   Notification|        |       |      |
 |         |          |     Request   |        |       |      |
 |         |    Ack   |      ->       |        |       |      |
 |_________|__________|_______________|________|_______|______|
```

The above diagram describes the call flow between a trunk with R2 signaling in a trunking gateway and an SS7 trunk in a trunking gateway. R2 is a type of Channel Associated Signaling (CAS) and this call flow assumes the use of an R2 package. The following diagram describes a simplified R2 package. In this package digit arrival events are not observed by the Call Agent and therefore cannot be part of a NotificationRequest. The first event that the Call Agent can observe in an R2 trunk is a "call-in" event which is a combination of the arrival at the gateway of a trunk seizure signal and collection of the called (destination) and calling (source) and calling party numbers from the R2 registers. The notification for a call-in event occurs when digit collection is completed. The clear forward event indicates that the exchange at the other side released the trunk.

Symbol	Definition	R	S	Duration
ci	Call in	x		
cf	Clear forward	x		
dn	Destination number			
sn	Source number			

```
        |         |                       |     |                       |
```

The first command is a NotificationRequest, sent by the Call Agent to the R2 trunking gateway. The request will consist of the following lines:

```
        RQNT 1201 trunk-group-1/*@r2tgw-2567.whatever.net MGCP 0.1
        N: ca@ca1.whatever.net:5678
        X: 0123456789AB
        R: ci
```

The gateway, at that point, is instructed to look for a call-in event in any of the trunks corresponding to a trunk group named trunk-group-1. The gateway responds with the acknowledgement message:

```
        200 1201 OK
```

When the call-in event is detected, the gateway sends a Notify to the Call Agent:

```
        NTFY 2001 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
        N: ca@ca1.whatever.net:5678
        X: 0123456789AB
        O: ci, dn(5313456789), sn(5845430978)
```

This notification indicates the occurrence of a call-in event in trunk #5 in trunk-group-1. It also contains the collected destination (dn) and source (sn) party numbers. The Call Agent immediately acknowledges that notification.

```
        200 2001 OK
```

At this stage, the Call Agent sends a NotificationRequest to wait for a trunk release signal: RQNT 1203 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1 X: 0123456789AD R: cf The Call Agent immediately acknowledges that command.

```
        200 1203 OK
```

The Call Agent will then seize the incoming circuit, creating a connec- tion:

```
        CRCX 1204 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: recvonly
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data:

```
200 1204 OK
I:FDE234C8

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The Call Agent, having seized the incoming trunk and completed a routing look up to identify the outgoing gateway, seizes the outgoing trunk. It does so by sending a connection command to the egress gate- way:

```
CRCX 1205 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: sendrecv

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The SS7 trunking gateway acknowledges the connection command, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1205 OK
I:32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1297 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The Call Agent relays the information to the ingress gateway, using a ModifyConnection command:

```
MDCX 1206 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: recvonly
```

```
v=0
c=IN IP4 128.96.63.25
m=audio 1297 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The R2 trunking gateway immediately acknowledges the modification:

```
200 1206 OK
```

At this stage, the Call Agent has established a half duplex transmission path. The subscriber that originated the call will be able to receive signals, such as tones or announcements, that the remote switch may send through the trunking gateways.

After the called user answers the call, the Call Agent will receive an answering message (ANM) from the CO switch. At that point, it sends a ModifyConnection command, to place the connection in full-duplex mode:

```
MDCX 1209 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv
```

The R2 trunking gateway acknowledges the modify command:

```
200 1209 OK
```

At this point, the connection is established.

When the Call Agent receives the REL message from the CO switch, it will have to tear down the call. It will do so by sending to both gateways a DeleteConnection command:

```
DLCX
1210 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8

DLCX 1211 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:32F345E2
```

The gateways will respond with acknowledgements that should include a "call parameters" header fields:

```
250 1210 OK
```

```
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,
LA=48

250 1211 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27,
LA=48
```

Finally, the Call Agent issues a new NotificationRequest, to be ready to receive the next call-in event detected by the trunking gateway at the specified trunk:

```
RQNT 1212 trunk-group-1/5@r2tgw-2567.whatever.net MGCP 0.1
X: 0123456789B0
R: hd
```

The gateway will acknowledge this message:

```
200 1212 OK
```

Both gateways, at this point, are ready for the next call.

2.5. Basic call, from an ISDN gateway to an SS7/TGW

PBX	Business	CA	TGW	SS7/	CO
	GW			ISUP	
SETUP	->	->			
<-	<-	CALLPROC.			
	<-	CRCX			
	Ack	->			
		CRCX	->		
			(cut in)		
		<-	ack		
		IAM	->		
	<-	MDCX		IAM	->
	Ack	->		<-	ACM
		<-	- -	ACM	
<-	<-	ALERT			
	<-	RQNT			
	Ack	->			
<-	Ringing				
				<-	ANM
		<-	- -	ANM	
	<-	RQNT+MDCX			
	Ack	->			
				<-	REL
		<-	- -	REL	
	<-	DLCX			
		DLCX	->		
	Perf				
	Data	->			
		<-	Perf		
			data		
		RLC	- -	->	->
<-	<-	RLSE			
RLCOM	->	->			

The above diagram describes the call flow between a trunk with ISDN signaling in a business gateway and an SS7 trunk in a trunking gateway.

This call flow assumes that the ISDN Q.931 signaling messages are "back-hauled" to the call agent. The tunnelling protocol, together with configuration tables, allows the call agent to associate the signalling messages to the endpoint corresponding to the B-channel. The specifics of the tunnelling protocol are being worked on by the IETF.

The call is triggered by the arrival of a SETUP command, which is relayed to the Call Agent. The call agent analyzes teh command, to obtain the destination (dn) and source (sn) party numbers. The call agent sends a call progress message, which is tunneled to the gateay and relayed over the D-Channel. The Call Agent then seizes the incoming circuit, creating a connection:

```
CRCX 1204 isdn-trunk-group-1/3@isdngw-45.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: recvonly
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session descrip- tion used to receive audio data:

```
200 1204 OK I:FDE234C8

v=0 c=IN IP4 128.96.41.1 m=audio 3456 RTP/AVP 0 96 a=rtpmap:96
G726-32/8000
```

The Call Agent, having seized the incoming trunk and completed a routing look up to identify the outgoing gateway, seizes the outgoing trunk. It does so by sending a connection command to the egress gate- way:

```
CRCX 1205 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: sendrecv

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The SS7 trunking gateway acknowledges the connection command, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1205 OK
I:32F345E2
```

```
        v=0
        c=IN IP4 128.96.63.25
        m=audio 1297 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The Call Agent relays the information to the ingress gateway, using a ModifyConnection command:

```
        MDCX 1206 isdn-trunk-group-1/3@isdngw-45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:FDE234C8
        M: recvonly

        v=0
        c=IN IP4 128.96.63.25
        m=audio 1297 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The business gateway immediately acknowledges the modification:

```
        200 1206 OK
```

At this stage, the Call Agent has established a half duplex transmission path. The subscriber that originated the call will be able to receive signals, such as tones or announcements, that the remote switch may send through the trunking gateways.

When the call progresses, the Call Agent will receive from the remote switch progress messages, for example an "address complete" message (ACM). The Call Agent will tunnel an ALERT message to the originating PBX. It may, if needed, send a notification request command to the gateway, to generate alerting tones over the B-channel:

```
        RQNT 1207 isdn-trunk-group-1/3@isdngw-45.whatever.net MGCP 0.1
        X: 0123456789AE
        S: rt
```

The gateway immediately acknowledges the command:

```
        200 1207 OK
```

After the called user answers the call, the Call Agent will receive an answering message (ANM) from the CO switch. At that point, it will send a ModifyConnection command to the business gateway, to place the connection in full duplex mode, and an embedded NotificationRequest to stop the ringing tones:

```
        MDCX 1209 isdn-trunk-group-1/3@isdngw-45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I: FDE234C8
        M: sendrecv
        X: 0123456789AF
        S:
```

The business gateway will acknowledge the command:

```
        200 1209 OK
```

At this point, the connection is established.

When the Call Agent receives the REL message from the CO switch, it will have to tear down the call. It will do so by sending to both gateways a DeleteConnection command:

```
        DLCX 1210 isdn-trunk-group-1/3@isdngw-45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:FDE234C8

        DLCX 1211 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:32F345E2
```

The gateways will respond with acknowledgements that should include a "call parameters" header fields:

```
250 1210 OK P: PS=1245, OS=62345, PR=780, OR=45123, PL=10,
JI=27, LA=48

250 1211 OK P: PS=790, OS=45700, PR=1230, OR=61875, PL=15,
JI=27, LA=48
```

Having freed the local resource, the call agent will confirm the release by sending a RLC message to the next switch, and will also send a release message through the Q.931 tunnel to the PBX. The PBX will send back a release confirmation.

Both gateways, at this point, are ready for the next call.

2.6. Basic call with continuity test, TGW to RGW

```
 _______________________________________________________
| CO |  SS7/|  TGW|     CA     |  CDB|  ACC|  RGW|  Usr|
|    |  ISUP|     |            |     |     |     |     |
|____|______|_____|____________|_____|_____|_____|_____|
| IAM|   -> |     |            |     |     |     |     |
|    |  IAM |  - -|      ->    |     |     |     |     |
|    |      |     |    Check   |  -> |     |     |     |
|    |      |     |      <-    |  IP |     |     |     |
|    |      |  <- |   Create   |     |     |     |     |
|    |      |     |  Connection|     |     |     |     |
|    |      |  Ack|      ->    |     |     |     |     |
| COT|   -> |     |            |     |     |     |     |
|    |  COT |  - -|      ->    |     |     |     |     |
|    |      |  <- |   Modify   |     |     |     |     |
|    |      |     |  Connection|     |     |     |     |
|    |      |     |   Create   |     |     |     |     |
|    |      |     |  Connection|  - -|  - -|  -> |     |
|    |      |     |      <-    |  - -|  - -|  Ack|     |
|____|______|_____|____________|_____|_____|_____|_____|
```

This diagram shows the various exchange of messages during the beginning of a call from a telephone user on the circuit-switched PSTN to a residential user connected to a residential gateway. During these exchanges the Call Agent uses MGCP to control both the TGW and the residential gateway. The circuit switch decides to execute a continuity test during the call establishment. The exchanges occur on two sides.

Upon reception of the IAM message, the Call Agent recognizes that a continuity test has been requested. It immediately sends a CreateConnection request to the trunking gateway to connect to the incoming trunk, creating a connection:

```
CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: conttest
```

The trunking gateway recognizes that the mode is set to "conttest". It places the circuit in "continuity test" mode, ready to send back the 2010 Hz return tone if it receives a 1780 Hz tone. The gateway acknowledges the creation of the connection, sending back the identification of the newly created connection and the session description used to receive audio data:

```
200 1237 OK
I: FDE234C8

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

At this point, the call agent is waiting for the result of the continuity test. The calling switch is sending the test tone (1780 Hz); if it receives the return tone (2010 Hz), it will send a "continuity passed" message (COT). At this point, the call agent will send a modify connection message to the gateway, in order to place the connection in "recvonly" mode:

```
MDCX 1238 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
```

The gateway will immediately acknowledge that command:

```
200 1238 OK
```

The Call Agent will then proceed with the establishment of the call.

2.7. Hairpin connection, regular set-up

The figure below gives the flow that results in an hairpin connection on the same gateway. In this flow, we assume that the exchange of messages is exactly comparable to what would happen in a call relayed between two trunking gateways, with the sole difference that the call will be relayed on a local connection.

```
 _______________________________________
| sw1|  sw2|  SG |   CA |  TGW-1|  TGW-2|
|____|_____|_____|______|_______|_______|
| IAM|  - -|  -> |      |       |       |
|    |     |  IAM|  ->  |       |       |
|    |     |     |  CRCX|   ->  |       |
|    |     |     |    <-|   ACK |       |
|    |     |     |  CRCX|   - - |   ->  |
|    |     |     |    <-|   - - |   ACK |
|    |     |   <-|  IAM |       |       |
|    |   <-|  IAM|      |       |       |
|    |  ACM|  -> |      |       |       |
|    |     |  ACM|  ->  |       |       |
|    |     |   <-|  ACM |       |       |
|  <-|  - -|  ACM|      |       |       |
|    |  ...|     |      |       |       |
|    |  ANM|  -> |      |       |       |
|    |     |  ANM|  ->  |       |       |
|    |     |     |  MDCX|   ->  |       |
|    |     |     |    <-|   ACK |       |
|    |     |   <-|  ANM |       |       |
|  <-|  - -|  ANM|      |       |       |
| REL|  - -|  -> |      |       |       |
|    |     |  REL|  ->  |       |       |
|    |     |     |  DLCX|   ->  |       |
|    |     |     |    <-|   ACK |       |
|    |     |   <-|  RLC |       |       |
|  <-|  - -|  RLC|      |       |       |
|    |     |     |  DLCX|   ->  |       |
|    |     |     |    <-|  Perf |       |
|    |     |     |      |  data |       |
|    |     |   <-|  REL |       |       |
|    |   <-|  REL|      |       |       |
|    |  RLC|  -> |      |       |       |
|    |     |  RLC|  ->  |       |       |
|____|_____|_____|______|_______|_______|
```

During these exchanges the MGCP is used by the Call Agent to control two endpoints located on the same TGW.

The exchanges start with the arrival from the first switch (SW1) of an SS7/ISUP "IAM" message, relayed by the signalling gateway to the Call Agent. The call agent performs the routing, and determines that the call will have to be relayed towards the second switch (SW2), using a trunk located on the same gateway.

The call agent starts the exchange by seizing the endpoint referenced in the IAM message:

```
CRCX 1204 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: nt=LOCAL
M: recvonly
```

Upon reception of that command, the trunking gateway prepares a "local" connection description.

```
200 1204 OK
I:FDE234C8

v=0
c=IN tgw.whatever.net trunk-group-1/17
m=audio 0 LOCAL 0
```

The call agent, upon reception of this acknowledgement, sends a connection request to the trunking gateway, asking the gateway to reserve a trunk in the group connected to the second switch:

```
CRCX 1205 trunk-group-2/$@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: nt=LOCAL
M: sendrecv

v=0
c=IN tgw.whatever.net trunk-group-1/17
m=audio 0 LOCAL 0
```

The call agent selects a trunk in the group, and acknowledges the creation:

```
200 1204 OK
I:abc0

v=0
c=IN tgw.whatever.net trunk-group-2/13
```

```
m=audio 0 LOCAL 0
```

The two commands have created a one way path, suitable for forwarding ring tones and announcements to the calling party. The call agent can now relay the call by sending an IAM message to the second switch. When the ACM is received, it is immediately relayed to the first switch.

After some time, the call is answered, and an ANM message is relayed from the second switch to the call agent. The call agent will first validate the call by asking the first end-point to place the connection in duplex mode:

```
MDCX 1206 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv

v=0
c=IN tgw.whatever.net trunk-group-2/13
m=audio 0 LOCAL 0
```

The trunking gateway executes and acknowledges the modification:

```
200 1206 OK
```

The call agent can now relay the ANM message to the calling switch, and both parties can talk.

At the end of the call, in our example, the calling party hangs up. The first switch sends a release message to the call agent, through the signalling gateway. The call agent releases the connection on the first end-point:

```
DLCX 1207 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1217 OK
P: OS=62345, OR=62345
```

The call agent can now confirm the release of the trunk, sending an RLC message to the first switch.

In parallel, the call agent releases the connection to the second endpoint:

```
DLCX 1208 trunk-group-2/13@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:abc0
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1218 OK
P: OS=62345, OR=62345
```

After receiving the acknowledgement, the call agent can relay the release message to the second switch. The switch will in turn acknowledge the release.

2.8. Hairpin connection, accelerated set-up

The figure below gives the flow that results in an hairpin connection on the same gateway. Contrary to the previous example, we will assume that the call agent uses a special-case software, and takes full benefit of the call’s locality to accelerate the call set-up.

User1	User2	RGW	CA
		<--	RQNT
		ACK	-->
OFF HOOK	---	-->	
<--	---	dialtone	
digits	---	NTFY	-->
		<--	ACK
		<--	CRCX+RQNT,
	Ring	<--	CRCX+RQNT
		ACK	-->
		ACK	-->
	OFF HOOK	NTFY	-->
		<--	ACK
		<--	RQNT
		ACK	-->
on-hook	---	NTFY	-->
		<--	ACK
		<--	DLCX+RQNT
		<--	DLCX
		ACK	-->
		ACK	-->
	on-hook	NTFY	-->
		<--	ACK
		<--	RQNT
		ACK	-->

The first command is a NotificationRequest, sent by the Call Agent to the residential gateway. The request will consist of the following lines:

```
RQNT 1201 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hd(E(dl;hu, D/[0-9#*T](D);)
        D: (0T|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
```

The gateway immediately acknowledges the command, repeating in the acknowledgement message the transaction id that the Call Agent attached to the query.

```
        200 1201 OK
```

When the off hook event is noticed, the gateway provides the dial tone to the line (the delay between off-hook and dialtone is thus minimal.) The gateway will then start accumulating digits according to that digit map. When it has noticed a sufficient set of values, it will notify the observed string to the Call Agent:

```
        NTFY 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        N: ca@ca1.whatever.net:5678
        X: 0123456789AC
        O: 912018294266
```

The Call Agent immediately acknowledges that notification.

```
        200 2002 OK
```

The call agent analyzes the called number and determines that this is an hairpin connection: the called party is located on the same gateway, on endpoint/2. The CallAgent can prepare two simultaneous CreateConnection commands, creating the two legs of the connection. Because we are not too concerned at this stage with tax evasion, the The Call Agent will then seize the incoming circuit, creating a connection. The create connection sent to the first endpoint piggybacks a notification request, to stop collecting digits yet continue watch for an on-hook transition. The CreateConnection sent to the second endpoint piggybacks a request to generate ringing and look for off-hook. Both commands can be sent in a single UDP packet:

```
          CRCX 1204 endpoint/1@rgw-2567.whatever.net MGCP 0.1
          C: A3C47F21456789F0
          X: 0123456789AD
          M: sendrecv
          R: hu
          S:

          v=0
          c=LOCAL rgw-2567.whatever.net endpoint/2
          m=audio 0 LOCAL 0
          .
          CRCX 1205 endpoint/2@rgw-2567.whatever.net MGCP 0.1
          C: A3C47F21456789F0
          X: 9875659876
          M: sendrecv
          R: hd
          S: rg

          v=0
          c=LOCAL rgw-2567.whatever.net endpoint/1
          m=audio 0 LOCAL 0
```

> We should note that the call agent does not send the local connection options since it knows that it is a local (a.k.a. "hairpin") connection: the connections are entirely described by the SDP text.

The gateway immediately acknowledges the creations, sending back in two messages the identification of the newly created connections:

```
          200 1204 OK
          I:FDE234C8
          .
          200 1204 OK
          I:9867659A
```

The residential gateway, at that point, is instructed to look for an off-hook event on the second endpoint, and to report it. When the gateway notices the off hook event, it sends a Notify command to the Call Agent:

```
          NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1
          X: 9875659876
          O: hd
```

The Call Agent immediately acknowledges that notification:

```
          200 2001 OK
```

The Call agent will now send a NotificationRequest command to the gateway, asking to look for an off-hook event on the second end-point:

```
RQNT 1206 endpoint/2@rgw-2567.whatever.net MGCP 0.1
X: 987565989A
R: hu
S:
```

The gateway acknowledges that command:

```
200 1206 OK
```

At this point the call is active between the two residential gateway users.

When the first user goes off hook, it sends a notification to the call agent:

```
NTFY 2010 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 987565989A
O: hu
```

The call agent acknowledges the notification. It can, in a single UDP message, send the acknowledgement and the DeleteConnection commands that will clear the call. For the first gateway, the command embeds a notification request that readies that gateway for the next call:

```
200 2010 OK
.
DLCX 1210 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
N: ca@ca1.whatever.net:5678
X: 012345673FDE
R: hd(E(dl;hu, D/[0-9#*T](D);)
S:
.
DLCX 1211 endpoint/2@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: 9867659A
X: A3C5F0
R: hu
S:
```

The gateway will acknowledge the commands in a single UDP message that will carry the "local connection" version of the connection parameters.

```
250 1243 OK
P: OS=62345, OR=62345
.
250 1244 OK
P: OS=62345, OR=62345
```

When the second user goes off hook, the gateway sends a Notify commands

```
NTFY 2020 endpoint/2@rgw-2567.whatever.net MGCP 0.1
X: A3C5F0
O: hu
```

The Call agent follows with a notification requests, transmitted in the same packet as the acknowledgement, in order to ready the line for the next call:

```
200 2020 OK
.
RQNT 1220 enpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456793E5
R: hd(E(dl;hu, D/[0-9#*T](D);)
S:
```

The gateway acknowledges the command, signalling that the second endpoint is now ready.

```
200 1220 OK
```

2.9. Hairpin connection, double end-point model

The figure below gives the flow that results in an hair-pin connection on the same gateway. In this flow, we assume that we use the "double endpoint" extensions to the create-connection command.

```
 ________________________________________
| sw1|  sw2|  SG |   CA |  TGW-1|  TGW-2|
|____|_____|_____|______|_______|_______|
| IAM|  - -|  -> |      |       |       |
|    |     |  IAM|  ->  |       |       |
|    |     |     |  CRCX|   ->  |  (->) |
|    |     |     |    <-|   ACK |  (ACK)|
|    |     |   <-|  IAM |       |       |
|    |   <-|  IAM|      |       |       |
|    |  ACM|  -> |      |       |       |
|    |     |  ACM|  ->  |       |       |
|    |     |   <-|  ACM |       |       |
|  <-|  - -|  ACM|      |       |       |
|    |  ...|     |      |       |       |
|    |  ANM|  -> |      |       |       |
|    |     |  ANM|  ->  |       |       |
|    |     |   <-|  ANM |       |       |
|  <-|  - -|  ANM|      |       |       |
| REL|  - -|  -> |      |       |       |
|    |     |  REL|  ->  |       |       |
|    |     |     |  DLCX|   ->  |       |
|    |     |     |    <-|  Perf |       |
|    |     |     |      |  data |       |
|    |     |   <-|  RLC |       |       |
|  <-|  - -|  RLC|      |       |       |
|    |     |     |  DLCX|   ->  |       |
|    |     |     |    <-|  Perf |       |
|    |     |     |      |  data |       |
|    |     |   <-|  REL |       |       |
|    |   <-|  REL|      |       |       |
|    |  RLC|  -> |      |       |       |
|    |     |  RLC|  ->  |       |       |
|____|_____|_____|______|_______|_______|
```

During these exchanges the MGCP is used by the Call Agent to control two endpoints located on the same TGW.

The exchanges start with the arrival from the first switch (SW1) of an SS7/ISUP "IAM" message, relayed by the signalling gateway to the Call Agent. The call agent performs the routing, and determines that the call will have to be relayed towards the second switch (SW2), using a trunk located on the same gateway.

The call agent starts the exchange by seizing the endpoint referenced in the IAM message:

```
CRCX 1204 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: nt=LOCAL
M: sendrecv
E2: trunk-group-2/$
```

Upon reception of that command, the trunking gateway prepares a "local" connection description between the specified endpoint (trunk-group-1/17) and an endpoint that it selects within the secund trunk group (trunk-group-2/13). The gateway acknowledges the two creations in a single message:

```
200 1204 OK
I:FDE234C8
Z2:trunk-group-2/13@tgw.whatever.net
I2:abc0
```

The command has created a path, between the two endpoints. The call agent can now relay the call by sending an IAM message to the second switch. When the ACM is received, it is immediately relayed to the first switch.

After some time, the call is answered, and an ANM message is relayed from the second switch to the call agent. The call agent immediately relays the ANM message to the calling switch, and both parties can talk.

At the end of the call, in our example, the calling party hangs up. The first switch sends a release message to the call agent, through the signalling gateway. The call agent releases the connection on the first endpoint:

```
DLCX 1207 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1217 OK
P: OS=62345, OR=62345
```

The call agent can now confirm the release of the trunk, sending an RLC message to the first switch.

In parallel, the call agent releases the connection to the second endpoint:

```
DLCX 1208 trunk-group-2/13@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:abc0
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1218 OK
P: OS=62345, OR=62345
```

After receiving the acknowledgement, the call agent can relay the release message to the second switch. The switch will in turn acknowledge the release.

3. Interaction between an MGCP gateway and an H.323 entity

MGCP is not intended to replace H.323, or even to compete with it. In fact, we should mostly consider it as a tool to enable distributed implementations of H.323. The combination of gateways and call agent behaves as a distributed H.323 system, using MGCP for internal communication. This system appears to H.323 users as a larger H.323 system, or, if one prefers, as an H.323 gatekeeper that implements the Gatekeeper routed call model.

In order to demonstrate the compatibility between MGCP and H.323, we provide here a step by step demonstration of 2 call set up scenarios:

- Call from an MGCP controlled residential gateway to an H.323 entity,

- Call from an H.323 entity to an MGCP controlled residential gateway.

We suppose, in these scenarios, that the H.323 entity is capable of using the fast start procedure defined in H.323v2.

3.1. Call from a residential gateway (RGW) to an H.323 user

```
 ______________________________________________________________________
|    Usr     |    RGW    |      CA      |   H.323   |   Usr    | GK  |
|____________|___________|______________|___________|__________|_____|
|            |    <-     |     RQNT     |           |          |     |
|            |    Ack    |      ->      |           |          |     |
|            |           |              |           |          |     |
|  Off-hook  |  Notify   |      ->      |           |          |     |
|     <-     |    Ack    |              |           |          |     |
|(Dial-tone) |    <-     |     RQNT     |           |          |     |
|            |    Ack    |      ->      |           |          |     |
|            |           |              |           |          |     |
|   Digits   |  Notify   |      ->      |           |          |     |
|     <-     |    Ack    |              |           |          |     |
| (progress) |    <-     |  CRCX+RQNT   |           |          |     |
|            |    Ack    |      ->      |           |          |     |
|            |           | (processing) |           |          |     |
|            |           |   TCP-SYN    |    ->     |          |     |
|            |           |      <-      | SYN, ACK  |          |     |
|            |           |   Set-up+    |           |          |     |
|            |           |  faststart   |    ->     |          |     |
|            |           |              |    ARQ    |  - - -   | ->  |
|            |           |              |    <-     |  - - -   | ACF |
|            |           |      <-      | alerting  |   ring   |     |
|            |           |   TCP ACK    |    ->     |          |     |
|(ring back) |    <-     |     RQNT     |           |          |     |
|            |    Ack    |      ->      |           |          |     |
|            |           |      <-      | connect + | off hook |     |
|            |           |              | faststart |          |     |
|            |           |   TCP ACK    |    ->     |          |     |
|            |    <-     |  MDCX+RQNT   |           |          |     |
|            |    Ack    |      ->      |           |          |     |
|            |           | (call est.)  |           |          |     |
|  on hook   |  Notify   |      ->      |           |          |     |
|     <-     |    Ack    |              |           |          |     |
|     <-     | DLCX+RQNT |              |           |          |     |
| perf data  |    ->     |              |           |          |     |
|            |           |   Rel. C.    |    ->     |          |     |
|            |           |   TCP-FIN    |    ->     |          |     |
|            |           |      <-      | FIN, ACK  |          |     |
|            |           |   TCP ACK    |    ->     |          |     |
|            |           |              | (signal)  | On-hook  |     |
|            |           |              |    DRQ    |  - - -   | ->  |
|            |           |              |    <-     |  - - -   | DCF |
|____________|___________|______________|___________|__________|_____|
```

During these exchanges the MGCP is used by the call agent to control the residential gateways. The call will be routed to an H.323 entity. The first command is a notification request, sent by the call agent to the residential gateway. The request will consist of the following lines:

```
RQNT 1201 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AB
R: hd
```

The gateway, at that point, is instructed to look for an off-hook event, and to report it. It will first acknowledge the request, repeating in the acknowledgement message the transaction id that the call agent attached to the query.

```
200 1201 OK
```

Note that this command is not actually simultaneous with the call. It can be issued long before the actual call, for example when the gateway is turned on, or after the end of a previous call.

When the off hook event is noticed, the gateway sends a Notify command to the call agent:

```
NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AB
O: hd
```

The call agent immediately acknowledges that notification.

```
200 2001 OK
```

The call agent examines the services associated to an off hook action (it could take special actions in the case of a direct line). In most cases, it will send a notification request, asking for digits. The current example provides the gateway with a digit map, and requests the gateway to play a dialtone:

```
RQNT 1202 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hu, [0-9#*T](D)
D: (0T | 00T | [1-7]xxx | 8xxxxxxx | #xxxxxxx | *xx | 91xxxxxxxxxx | 9011x.T)
S: dl
```

The gateway immediately acknowledges that request.

```
 200 1202 OK
```

The gateway will start accumulating digits according to that digit map. When it has noticed a sufficient set of values, it will notify the observed string to the call agent:

```
NTFY 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: 912018294266
```

The call agent immediately acknowledges that notification.

```
200 2002 OK
```

At this stage, the call agent seizes the incoming circuit, creating a connection. It will also send together with that creation request a notification request, to stop collecting digits yet continue watch for an on-hook transition:

```
CRCX 1204 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: recvonly
X: 0123456789AD
R: hu
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data:

```
200 1204 OK
I:  FDE234C8

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0
```

The SDP announcement, in our example, specifies the address at which the gateway is ready to receive audio data (128.96.41.1), the transport protocol (RTP), the RTP port (3456) and the audio profile (AVP). The audio profile refers to RFC 1890, which defines that the payload type 0 has been assigned for G.711 transmission.

The call agent, having seized the incoming trunk, proceeds with call routing. Using local databases, it determines that the dialed digits (912018294266) correspond to a H.323 entity. It will set up a TCP-IP connection and send an H.225/Q.931 "set-up" message to the H.323 entity. In this message, the "faststart" element carries a set of open logical channel proposals, derived from the SDP description received from the calling gateway:

```
                        faststart-1 OpenLogicalChannel ::= {
                           forwardLogicalChannelNumber   1,
                           forwardLogicalChannelParameters {
                              dataType   g711Ulaw64k 160, -- 20 ms G.711 frame
                              multiplexParameters
                                 h2250LogicalChannelParameters H2250LogicalChannelParameters  {
                                     sessionID 1,
                                     silenceSuppression   FALSE
                                 }
                           }
                        }

                        faststart-2 OpenLogicalChannel ::= {
                           forwardLogicalChannelNumber   2,
                           forwardLogicalChannelParameters {
                              dataType              nullData, -- pro forma
                              multiplexParameters   none
                           },
                           reverseLogicalChannelParameters   {
                              dataType   g711Ulaw64k 160, -- 20 ms frame
                              multiplexParameters
                                 h2250LogicalChannelParameters H2250LogicalChannelParameters {
                                    sessionID 2,
                                    mediaChannel unicastAddress iPAddress {
                                       network  ’80602901’H, -- 128.96.41.1
                                       tsapIdentifier 3456 -- port
                                    },
                                    mediaControlChannel unicastAddress iPAddress {
                                       network  ’80602901’H, -- 128.96.41.1
                                       tsapIdentifier 3457 -- port
                                    },
                                    silenceSuppression FALSE
                                 }
                           }
                        }
```

The H.323 alerts the user, and sends an H.225/Q.931 "alerting" message. On reception of this message, the call agent will send a notification request that instruct the RGW to generate alerting tones:

```
RQNT 1206 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AE
R: hu
S: v
```

When the called user accepts the call, the H.323 entity sends an H.225/Q.931

set-up
 message to the call agent. If the H.323 entity accepted the fast start procedure, the faststart
 parameter will contain the confirmation of the open logical channel creations in the two directions of communication:

```
        faststart-1 OpenLogicalChannel ::= {
           forwardLogicalChannelNumber   1,
           forwardLogicalChannelParameters {
              dataType   g711Ulaw64k 160, -- 20 ms frame
              multiplexParameters h2250LogicalChannelParameters {
                 sessionID 1,
                 mediaChannel unicastAddress iPAddress {
                                 network  ’80603F19’H,
                                 tsapIdentifier 3456, -- port
                 } ,
                 mediaControlChannel unicastAddress iPAddress {
                                 network  ’80603F19’H,
                                 tsapIdentifier 3457, -- port
                 } ,
                 silenceSuppression   FALSE
              }
          }
        }
        faststart-2 OpenLogicalChannel ::= {
           forwardLogicalChannelNumber   2,
           forwardLogicalChannelParameters {
              dataType   g711Ulaw64k 160, -- 20 ms frame
                    multiplexParameters h2250LogicalChannelParameters  {
                        sessionID 2,
                        silenceSuppression   FALSE
                    }
              }
        }
```

The call agent will send a modification request to the residential gateway, in order to establish a full duplex connection. The SDP payload, in that request, is derived from the parameters of the logical channel for transmission from the gateway to the H.323 entity.

```
MDCX 1209 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:  FDE234C8
M: sendrecv
X: 0123456789AF
R: hu

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0
```

The gateway will acknowledge this request:

```
200 1209 OK
```

At this point, the connection is established.

In our example, the call is terminated when the calling party hangs up. This triggers a Notify command to the call agent:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AF
O: hu
```

After this notification, the call agent should send an acknowledgement:

```
200 2005 OK
```

The call agent will then clear the call, by sending a delete connection request to the calling gateway. This request is combined with a notification request, to be ready to detect the next call issued by the residential gateway:

```
DLCX 1210 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
X: 0123456789B0
R: hd
```

The gateway will respond with a message that should include a "call parameters" header field:

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

The call agent will in parallel sends an H.225/Q.931 "release complete" message to the H.323 entity. It will then tear down the TCP-IP connection. The gateway, at this point, is ready for the next call. The H.323 user will be ready as soon as the H.323 entity notices that the phone is back on hook.

3.2. Basic call, H.323 to RGW

User	H.323	GK	CA	RGW	Usr
call	->				
	ARQ	->			
	<-	ACF			
	TCP SYN	- - -	->		
	<-	- - -	SYN+ACK		
	set-up+				
	fast start	- - -	->		
			(call		
			processing)		
			CRCX+RQNT	->	ring
			<-	Ack	
	<-	- - -	alerting		
	TCP ACK	- - -	->		
	(ringing)				
			<-	Notify	off hook
			Ack	->	
			RQNT	->	
			<-	Ack	
	<-	- - -	connect +		
			fast start		
	TCP ACK	- - -	->		
			(call		
			established)		
			<-	Notify	on hook
			Ack	->	
			(no		
			suspension)		
			RQNT	->	
			<-	Ack	
hangup	detected				
	Rel. C.				
	TCP FIN	- - -	->		
	<-	- - -	FIN ACK		
			DLCX+RQNT	->	
			<-	perf data	
	DRQ	->			
	<-	DCF			

This diagram shows the various exchange of messages during a call from an H.323 user to a residential user.

During these exchanges the MGCP is used by the call agent to control the residential gateway.

When the user initiates the call, the H.323 entity will perform a RAS transaction with its designated Gatekeeper. As a result of this transaction, it will learn the TCP-IP address of the call agent, and will set up a TCP-IP connection with the call agent. Once the TCP-IP connection is established, the H.323 entity sends a Q.931/H.225 connect message to the call agent. The message, in its user-to-user parameter, includes a "fast start" parameter that lists a set of OpenLogicalChannel proposals, such as for example:

```
faststart-1 OpenLogicalChannel ::= {
   forwardLogicalChannelNumber   1,
   forwardLogicalChannelParameters {
      dataType   g711Ulaw64k 160, -- 20 ms G.711 frame
      multiplexParameters h2250LogicalChannelParameters {
         sessionID 1,
         silenceSuppression   FALSE
      }
   }
}
faststart-2 OpenLogicalChannel ::= {
   forwardLogicalChannelNumber   2,
   forwardLogicalChannelParameters {
      dataType   nullData, -- pro forma
      multiplexParameters   none
   },
   reverseLogicalChannelParameters   {
      dataType   g711Ulaw64k 160, -- 20 ms frame
      multiplexParameters h2250LogicalChannelParameters {
         sessionID 2,
         mediaChannel unicastAddress iPAddress {
                     network  ’80602901’H, -- 128.96.41.1
                     tsapIdentifier 3456, -- port
         },
         mediaControlChannel unicastAddress iPAddress {
                     network  ’80602901’H, -- 128.96.41.1
                     tsapIdentifier 3457, -- port
         },
         silenceSuppression FALSE
      }
   }
}
```

Upon reception of the set-up message, the call agent will perform called routing functions and determine the end point that correspond to the called party number. It will reserve the outgoing circuit. It does so and at the same time it requests ringing, by sending to the residential gateway a create connection request combined with a notification request:

```
CRCX 1238 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711
M: sendrecv
X: 0123456789B1
R: hd
S: rg

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0
```

In this command, the SDP announcement is obtained by translating the "faststart" parameters corresponding to the receive channel announced by the caller - channel 2 in our case. The IP address, RTP port and authorized payload are derived from the "reverseLogicalChannel-Parameters" data elements. We derive the authorized payload type from the "dataType" element. We have however to check that this value is proposed in at least one of the forward channels.

The gateway will acknowledge the connection request, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1238 OK
I:  32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0
```

The phone is ringing, and the gateway, is instructed to look for an off-hook event, and to report it. The call agent sends an alerting message to the calling switch, which we assume will generate ringing tones for the calling user.

When the off hook event is noticed, the gateway sends a Notify command to the call agent:

```
NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1X: 0123456789B0
O: hd
```

The call agent immediately acknowledges that notification.

```
200 2001 OK
```

The call agent must now ask the residential gateway to notify the occurrence of an on-hook event. It does so by sending a notification request to the residential gateway:

```
RQNT 1241 endpoint/1@rgw-2567.whatever.net MGCP 0.1X: 0123456789B1
R: hu
```

The gateway acknowledges that request:

```
200 1241 OK
```

In parallel, the call agent will send a "connect" message to the H.323 agent. The message includes the "fast start" parameter, which will validate and complement the proposals that the caller sent:

```
faststart-1 OpenLogicalChannel ::= {
   forwardLogicalChannelNumber   1,
   forwardLogicalChannelParameters {
      dataType   g711Ulaw64k 160, -- 20 ms frame
      multiplexParameters h2250LogicalChannelParameters {
         sessionID 1,
         mediaChannel unicastAddress iPAddress {
                         network  ’80603F19’H,
                         tsapIdentifier 3456, -- port
         } ,
         mediaControlChannel unicastAddress iPAddress {
                         network  ’80603F19’H,
                         tsapIdentifier 3457, -- port
         } ,
         silenceSuppression   FALSE
      }
   }
}
faststart-2 OpenLogicalChannel ::= {
   forwardLogicalChannelNumber   2,
   forwardLogicalChannelParameters {
      dataType   g711Ulaw64k 160, -- 20 ms frame
      multiplexParameters h2250LogicalChannelParameters (
         sessionID 1,
         silenceSuppression   FALSE
      }
   }
}
```

After some time, in our example, the residential user hangs up. The notify request is sent to the call agent:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B1
O: hu
```

The call agent acknowledges the notification.

```
200 2005 OK
```

Upon reception of that notification, the call agent should send a "suspend" message to the calling H.323 entity, but the Q.931 suspend message should not be sent in H.225. In order to preserve the user experience, the call agent will simply initiate a timer, after which it would actually release the call. (In North-America, the call is not actually terminated if the called party hangs up. If it hangs down in a short interval, the call will be resumed.) The call agent, in any case, sends a notification request to the gateway, to look for an off-hook event.

```
RQNT 1243 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B2
R: hd
```

The gateway will acknowledge this request:

```
   200 1243 OK
```

In our example, the calling user releases the call immediately. The H.323 agent sends a "release complete" message to the call agent, which will then send a delete connection request to the residential gateway. The request sent to the gateway is combined with a request to detect a off-hook event, which will be used to detect rare conditions where the user would have gone off hook simultaneously with the release on the other side:

```
DLCX 1244 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
X: 0123456789B3
R: hd
I:  FDE234C8
```

The gateway will respond with a message that should include a "call parameters" header fields:

```
200 1244 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

The gateway, at this point, is ready for the next call.

4. Interworking between SIP and MGCP

The use of SDP in both MGCP and SIP makes interworking between these protocols very easy. In order to demonstrate this interworking, we provide here a step by step demonstration of 2 call set-up scenarios:

* Call from an MGCP controlled residential gateway to a SIP agent,

* Call from a SIP agent to an MGCP controlled residential gateway.

4.1. Call from a residential gateway (RGW) to a SIP user

Usr	RGW	CA	SIP	Usr
	<-	RQNT		
	Ack	->		
Off-hook	Notify	->		
	<-			
Ack				
(Dial-tone)	<-	RQNT		
	Ack	->		
Digit	Notify	->		
	<-	Ack		
(progress)	<-			
CRCX+RQNT				
	Ack	->		
		(processing)		
		INVITE	->	
ring				
		<-	resp. 180 (ringing)	
(ringing)	<-	RQNT		
	Ack	->		
		<-	resp. 200 (OK)	
off hook				
		Ack	->	
	<-	MDCX+RQNT		
	Ack	->		
		(call established)		
on hook	Notify	->		
	<-	Ack		
	<-	DLCX+RQNT		
	perf data	->		
		BYE	->	
		<-	resp. 200 (OK)	
			(local)	On-hook

During these exchanges the MGCP is used by the call agent to control the residential gateways. The call will be routed to a SIP agent. The first command is a notification request, sent by the call agent to the residential gateway. The request will consist of the following lines:

```
RQNT 1201 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AB
R: hd
```

The gateway, at that point, is instructed to look for an off-hook event, and to report it. It will first acknowledge the request, repeating in the acknowledgement message the transaction id that the call agent attached to the query.

```
200 1201 OK
```

> Note that this command is not actually simultaneous with the call. It can be issued long before the actual call, for example when the gateway is turned on, or after the end of a previous call.

When the off hook event is noticed, the gateway initiates a notification request to the call agent:

```
NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AB
O: hd
```

The call agent immediately acknowledges that notification.

```
200 2001 OK
```

The call agent examines the services associated to an off hook action (it could take special actions in the case of a direct line). In most cases, it will send a notification request, asking for digits. It will also provide the gateway with a digit map, and requests the gateway to play a dialtone:

```
RQNT 1202 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hu, D/[0-9#*T](D)
D: (0T | 1xxxxxxxxxx | [2-9]xxxxxx | [4789]11 | 011x.T)
S: dl
```

The gateway immediately acknowledges that request.

```
200 1202 OK
```

The gateway will start accumulating digits according to that digit map. When it has noticed a sufficient set of values, it will notify the observed string to the call agent:

```
NTFY 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: 912018294266
```

The call agent immediately acknowledges that notification.

```
200 2002 OK
```

At this stage, the call agent seizes the incoming circuit, creating a connection. It will also send together with that creation request a notification request, to stop collecting digits yet continue watch for an on-hook transition:

```
CRCX 1204 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: recvonly
X: 0123456789AD
R: hu
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data:

```
200 1204 OK
I:FDE234C8

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The SDP announcement, in our example, specifies the address at which the gateway is ready to receive audio data (128.96.41.1), the transport protocol (RTP), the RTP port (3456) and the audio profile (AVP). The audio profile refers to RFC 1890, which defines that the payload type 0 has been assigned for G.711 transmission. The gateway is also ready to use ADPCM encoding at 32 kbps (G.726 -4). There is no standard payload type associated to ADPCM, so the gateway mentions its readiness to use a non standard payload associated to the dynamic type 96. The "rtpmap" attribute entry associates the payload type 96 to G726-32/4.

The call agent, having seized the incoming trunk, proceeds with call routing. Using local databases, it determines that the dialed digits (912018294266) correspond to a SIP agent. It will send an invitation to that agent:

```
INVITE sip:huitema@sip-station.bellcore.com SIP/2.0
Via: SIP/2.0/UDP 128.96.41.12
From: sip:123456789@ca.whatever.net
To: Christian Huitema <huitema@sip-station.bellcore.com>
Call-ID: A3C47F21456789F0@ca.whatever.net
Cseq: 1 INVITE
Content-type: application/sdp
Content-Length: ...

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The SDP attachment, in the INVITE message, is directly copied from the acknowledgement of the Create Connection request. The SIP agent alerts the user, and sends an immediate acknowledgement:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 128.96.41.12
From: Christian Huitema <huitema@sip-station.bellcore.com>
To: 123456789@ca.whatever.net
Call-ID: A3C47F21456789F0@ca.whatever.net
Cseq: 1 INVITE
```

The call agent will send a notification request that instruct the RGW to generate alerting tones:

```
RQNT 1206 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AE
R: hu
S: v
```

When the called user accepts the call, the SIP agent sends an acceptation message to the call agent:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 128.96.41.12
From: "Christian Huitema" <sip:huitema@sip-station.bellcore.com>
To: sip:123456789@ca.whatever.net
Call-ID: A3C47F21456789F0@ca.whatever.net
CSeq: 1 INVITE
Content-type: application/sdp
Content-Length:...

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The gateway immediately acknowledges the call set up:

```
ACK huitema@sip-station.bellcore.com SIP/2.0
Via: SIP/2.0/UDP 128.96.41.12
From: 123456789@ca.whatever.net
To: huitema@sip-station.bellcore.com (Christian Huitema)
Call-ID: 187602141351@ca.whatever.net
```

Then, the call agent will send a modification request to the residential gateway, in order to establish a full duplex connection. The SDP payload, in that request, is copied from the SIP agent’s response:

```
MDCX 1209 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv
X: 0123456789AF
R: hu

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
+
```

The gateway will acknowledge this request:

```
200 1209 OK
```

At this point, the connection is established. In our example, the call is terminated when the calling party hangs up. This triggers a Notify command to the call agent:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789AF
O: hu
```

After this notification, the call agent should send an acknowledgement:

```
200 2005 OK
```

The call agent will then clear the call, by sending a delete connection request to the calling gateway. This request is combined with a notification request, to be ready to detect the next call issued by the residential gateway:

```
DLCX 1210 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0

I:FDE234C8
X: 0123456789B0
R: hd
```

The gateway will respond with a message that should include a "call parameters" header field:

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

The call agent will in parallel sends a BYE request to the SIP agent:

```
BYE sip:huitema@sip-station.bellcore.com SIP/2.0
Via: SIP/2.0/UDP 128.96.41.12
From: sip:123456789@ca.whatever.net
To: "Christian Huitema" <sip:huitema@sip-station.bellcore.com>
Call-ID: A3C47F21456789F0@ca.whatever.net
CSeq: 2 BYE
```

The SIP agent will acknowledge that request:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 128.96.41.12
From: "Christian Huitema" <sip:huitema@sip-station.bellcore.com>
To: sip:123456789@ca.whatever.net
Call-ID: A3C47F21456789F0@ca.whatever.net
CSeq: 2 BYE  SIP/2.0 200 OK
```

The gateway, at this point, is ready for the next call. The SIP user will be ready as soon as the SIP agent notices that the phone is back on hook.

4.2. Basic call, SIP to RGW

User	SIP agent	CA	RGW	Usr
call	->			
	INVITE	->		
		(call processing)		
		CRCX+RQNT	->	
ring				
		<-		
Ack				
	<-	resp. 180		
	(ringing)	(ringing)		
		<-	Notify	off hook
		Ack	->	
		RQNT	->	
		<-	Ack	
	<-	resp. 200 (OK)		
	ACK	->		
		(call		
		established)		
		<-	Notify	on hook
		Ack	->	
		(no susp.		
		message)		
		RQNT	->	
		<-	Ack	
hangup	detected			
	BYE	->		
		DLCX+RQNT	->	
		<-	perf data	

This diagram shows the various exchange of messages during a call from a SIP user to a residential user. During these exchanges the MGCP is used by the call agent to control the residential gateway.

When the user initiates the call, the SIP agent will send an invitation to the call agent. (Our diagram assumes that the SIP agent sends that invitation directly; in fact, there could be several relays.) An example of invitation could be:

```
INVITE sip:watson@boston.bell-telephone.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: "T. A. Watson" <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 1 INVITE
Subject: Mr. Watson, come here.
Content-type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```

Upon reception of the set-up message, the call agent will perform call routing functions and determine the end point that corresponds to the invited user. It will then reserve the outgoing circuit. It does so at the same time that it requests ringing, by sending to the residential gateway a connection request combined with a notification request:

```
CRCX 1238 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711
M: sendrecv
X: 0123456789B1
R: hd
S: rg

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```

In this command, the SDP announcement is directly copied from the invitation. The gateway will acknowledge the connection request, sending in the session description its own parameters such as address, ports and RTP profile:

```
200 1238 OK
I:32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 3
```

The phone is ringing, and the gateway, is instructed to look for an off-hook event, and to report it. The call agent sends an alerting message to the calling SIP agent, which will generate ringing tones for the calling user:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 169.130.12.5
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: sip:watson@bell-telephone.com
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 1 INVITE
```

When the off hook event is noticed, the gateway initiates a notification request to the call agent:

```
NTFY 2001 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B0
O: hd
```

The call agent immediately acknowledges that notification.

```
200 2001 OK
```

The call agent must now ask the residential gateway to notify the occurrence of an on-hook event. It does so by sending a notification request to the residential gateway:

```
RQNT 1241 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B1
R: hu
```

The gateway acknowledges that request:

```
200 1241 OK
```

In parallel, the call agent will send a final answer to the SIP agent. The message, in its payload, copies the SDP announcement that was sent by the gateway:

```
SIP/2.0 200 OK
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: sip:watson@bell-telephone.com
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 1 INVITE
Contact: sip://watson@boston.bell-telephone.com
Content-Length: ...

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 3
```

The SIP agent acknowledges the set-up:

```
ACK sip:watson@boston.bell-telephone.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: "T. A. Watson" <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 1 ACK
```

At this point, the call is established. After some time, in our example, the residential user hangs up. The notify request is sent to the call agent:

```
NTFY 2005 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B1
O: hu
```

The call agent acknowledges the notification.

```
200 2005 OK
```

Upon reception of that notification, the call agent should send a "suspend" message to the calling SIP agent, but there is no such message in SIP. In order to preserve the user experience, the call agent will simply initiate a timer, after which it would actually release the call. (In North-America, the call is not actually terminated if the called party hangs up. If it hangs down in a short interval, the call will be resumed.) The call agent, in any case, sends a notification request to the gateway, to look for an off-hook event.

```
RQNT 1243 endpoint/1@rgw-2567.whatever.net MGCP 0.1
X: 0123456789B2
R: hd
```

The gateway will acknowledge this request:

```
200 1243 OK
```

In our example, the calling user releases the call immediately. The SIP agent sends a BYE message to the call agent:

```
BYE sip:watson@boston.bell-telephone.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: "T. A. Watson" <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 2 BYE
```

The call agent acknowledges that message:

```
SIP/2.0 200 OK
From: "A. Bell" <sip:a.g.bell@bell-telephone.com>
To: sip:watson@bell-telephone.com
Call-ID: 187602141351@worcester.bell-telephone.com
CSeq: 2 BYE
```

The call agent then sends a delete connection request to the residential gateway. The request sent to the gateway is combined with a request to detect a off-hook event, which will be used to detect rare conditions where the user would have gone off hook simultaneously with the release on the other side:

```
DLCX 1244 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
X: 0123456789B3
R: hd
I:FDE234C8
```

The gateway will respond with a message that should include a "call parameters" header fields:

```
200 1244 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

The gateway, at this point, is ready for the next call.

5. Data calls

We present here a set of call flows corresponding to data calls:

* Basic data call,

* Outgoing data call through a NAS,

* Call back, using a NAS,

* Data call to a NAS, using L2TP,

* Basic data call with continuity test.

5.1. Basic data call to a NAS

PC	CO	SS7/ ISUP	NAS	CA	ACC	Radius
dials in						
	IAM	->				
		IAM	- -	->		
				Check called		
				number.		
				Notices		
				data call.		
				Call start	->	
			<-	Create		
				Connection		
				(data)		
			Ack	->		
				Connection		
				is completed.		
				Call		
				established	->	
		<-	- -	ANM		
	<-	ANM				
modem	- -	- -	->			
<-	- -	- -	handshake			
PPP	- -	- -	->			
			obtain			
			user-id,			
			password			
			Check	- -	- -	->
			<-	- -	- -	Ack
<-	- -	- -	Validates			
			call,			
<-	- -	- -	procures			
			IP			
			address			
Connected						
to						
the						
Internet						

```
 _____________________________________________________________
|     PC      |  CO |  SS7/|   NAS  |      CA     |  ACC|  Radius|
|             |     |  ISUP|        |             |     |        |
|_____________|_____|______|________|_____________|_____|________|
|   Closes    |     |      |        |             |     |        |
| connection. |     |      |        |             |     |        |
|             |  REL|  ->  |        |             |     |        |
|             |     |  REL |   - -  |     ->      |     |        |
|             |     |      |   <-   |   Delete    |     |        |
|             |     |      |        |  Connection |     |        |
|             |     |      |  Perf  |             |     |        |
|             |     |      |  data  |     ->      |     |        |
|             |     |  <-  |  - -   |     RLC     |     |        |
|             | <-  |  RLC |        |             |     |        |
|             |     |      |        |   Call end  |  -> |        |
|_____________|_____|______|________|_____________|_____|________|
```

This diagram shows the exchange of messages during a call from a modem user to an Internet Service Provider, using a trunking gateway that doubles as a Network Access Server. During these exchanges the MGCP is used by the Call Agent to control both the trunking gateway. Since there is no "other end" of the call, only the trunk gateway is involved in the call.

Upon reception of the IAM message, the Call Agent determines that the call is a data call (e.g., by bearer capability, the called number, etc.). Using configuration databases, the Call Agent selects the type of modem parameters and authentication parameters that correspond to the called number and to the calling number. It uses this knowledge to send a CreateConnection command to the NAS, programming the incoming trunk:

```
CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
M: data
X: 0123456789B1
R: cl, ax

v=0
m=nas/radius
c=IN IP4 radius.example.net
a=bearer:v.32
a=framing:ppp-asynch
a=dialed:18001234567
a=dialing:2345678901
```

The trunking gateway checks that it has adequate resources for the call. If the trunking gateway did not have adequate resources, for example if it could not support the requested modem type, it should refuse the creation and send an error response to the Call Agent. If the gateway has sufficient resources, it immediately acknowledges the creation, sending back the identification of the newly created connection. (There is no need to transmit a session description in the case of a data call.)

```
200 1237 OK
I: FDE234C8
```

The Call Agent, knowing that this is a data call, can immediately acknowledge the establishment of the connection, sending an ANM message back to the calling switch.

The trunk gateway connects the incoming trunk to a DSP loaded with the specified modem code. Once the call is established, the modem of the calling PC will start a training sequence with the modem associated to the trunk in the trunk gateway. The caller will then proceed to a normal PPP synchronization, which probably implies a PPP login. The authentication parameters, in our example, are checked using Radius. The Radius server that will be used is typically chosen as a function of the called number, which identifies the data service that the calling modem requested. In fact, the number can also be used to identify the specific form of authentication that is requested (but not usually).

In our example, the call is completed when the calling modem hangs up. This triggers an ISUP release message, which is forwarded to the Call Agent. The Call Agent will request the NAS to delete the connection:

```
DLCX 1244 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
```

The gateways will respond with a message that should include a "call parameters" header fields:

```
250 1244 OK
P: PS=1245, OS=62345, PR=780, OR=45123
```

We should note that, because this is a data call, the call parameters only include a count of the packets and octets that were sent and received.

5.2. Outgoing data call through a NAS

```
         _____________________________________________________________________
        |     PC    |  CO |  SS7/|     NAS    |      CA     |  ACC|   Router |
        |           |     |  ISUP|            |             |     |          |
        |___________|_____|______|____________|_____________|_____|__________|
        |           |     |      |            |             |     |  notices |
        |           |     |      |            |             |     |  packet  |
        |           |     |      |            |             |     |   to PC  |
        |           |     |      |            |      <-     |  - -|    NTFY  |
        |           |     |      |            |      Ack    |  - -|     ->   |
        |           |     |      |            |  Decides to |     |          |
        |           |     |      |            |   place an  |     |          |
        |           |     |      |            |   outgoing  |     |          |
        |           |     |      |            |     call.   |     |          |
        |           |     |      |            |  Call start |  -> |          |
        |           |     |      |      <-    |    Create   |     |          |
        |           |     |      |            |  Connection |     |          |
        |           |     |      |            |    (data)   |     |          |
        |           |     |      |     Ack    |      ->     |     |          |
        |           |     |   <- |     - -    |      IAM    |     |          |
        |  (rings)  |  <- |  IAM |            |             |     |          |
        |           |  ACM|   -> |            |             |     |          |
        |           |     |  ACM |     - -    |      ->     |     |          |
        |  (answer) |  ANM|   -> |            |             |     |          |
        |           |     |  ANM |     - -    |      ->     |     |          |
        |           |     |      |            |  Connection |     |          |
        |           |     |      |            |  complete.  |     |          |
        |           |     |      |            |    Call     |     |          |
        |           |     |      |            |  established|  -> |          |
        |    PPP    |  - -|  - - |      ->    |             |     |          |
        |     <-    |  - -|  - - |  Validates |             |     |          |
        |           |     |      |    call,   |             |     |          |
        |           |     |      |  announces |             |     |          |
        |           |     |      |  IP address|      - -    |  - -|     ->   |
        | Connected |     |      |            |             |     |          |
        |  to the   |     |      |            |             |     |          |
        |  Internet |     |      |            |             |     |          |
        |           |     |      |            |             |     |          |
        |___________|_____|______|____________|_____________|_____|__________|
          ___________________________________________________________________
         |     PC      | CO  | SS7/|    NAS     |     CA      | ACC| Router|
         |             |     | ISUP|            |             |    |       |
         |_____________|_____|_____|____________|_____________|____|_______|
         |   Closes    |     |     |            |             |    |       |
         | connection. |     |     |            |             |    |       |
         |             | REL|  -> |            |             |    |       |
         |             |     | REL |    - -     |     ->      |    |       |
         |             |     |     |     <-     |   Delete    |    |       |
         |             |     |     |            | Connection|    |       |
         |             |     |     |   ceases   |             |    |       |
         |             |     |     | announcing|             |    |       |
         |             |     |     | IP address|     - -     | - -|   ->  |
         |             |     |     |    Perf    |             |    |       |
         |    data     | ->  |     |            |             |    |       |
         |             |     |  <- |    - -     |     RLC     |    |       |
         |             | <-  | RLC |            |             |    |       |
         |             |     |     |            |  Call end   | -> |       |
         |_____________|_____|_____|____________|_____________|____|_______|
```

This diagram shows the exchange of messages during a call from an an Internet Service Provider to a modem, using a trunking gateway that doubles as a Network Access Server. During these exchanges the MGCP is used by the Call Agent to control both the NAS, and will also be used between the Call Agent and a default router of the ISP.

In the example configuration, the calls are set on demand, when data have to actually be sent from the Internet to the dial-up user. When no connection is established, the local routing is configured to send the packets towards a default router which may or may not be the same machine as the NAS. In redundant configurations, there could be many default routers. Each of these default routers has been programmed (through a notification request) to send a notification to the Call Agent when it receives a packet on the default route:

```
NTFY 2005 default-route@router25.whatever.net MGCP 0.1
X:
0123456789AF
O: pa(192.96.41.1)
```

After this notification, the Call Agent should send an acknowledgement:

```
        200 2005 OK
```

(We should note here that using MGCP for this function is a stretch. There are other protocols, notably RMON, that already provide an adequate service. These protocols could be used instead of MGCP without affecting the discussion that follows.)

The Call Agent deduces from the notification that a circuit should be established towards the dial-up user, or towards the dial-up router. Using configuration databases, the Call Agent selects the number that should be called, and also the type of modem parameters and authentication parameters that correspond to the called number. The Call Agent uses its routing table to select an adequate NAS, with an available outgoing trunk. It uses a create connection command to seize this outgoing trunk:

```
        CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        M: data
        X: 0123456789B1
        R: cl

        v=0
        m=nas/none
        c=IN IP4 128.96.41.1
        a=subnet:IN IP4 123.45.67.64/26
        a=bearer:isdn64
        a=framing:ppp-hdlc
        a=dialed:18001234567
        a=dialing:2345678901
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection. (There is no session description in the case of a data call.)

```
        200 1237 OK
        I: FDE234C8
```

Once the trunk has been seized, the Call Agent will send an IAM message to the switch that controls the trunk. The dialed PC will "ring" and eventually take the call, triggering the arrival of progress messages and then an answer message (ANM). At that point, the Call Agent knows that the call is established.

The DSP associated to the incoming trunk has been loaded with the specified modem code - a simple HDLC framing in our example. Once the call is established, the calling PC will train with the modem associated with the trunk. In our example, no authentication is requested: the Call Agent has identified the dialed user through its called number.

Once the association is established and the IP service is validated, the gateway announces that it serves the local user. In our example, there is no address configuration performed through PPP: the dialed user has a permanent address, which has been programmed when it subscribed to the service. However, one the circuit is validated, the gateway should start announcing its access to this permanent address in the routing tables. In our example, the dialed station is in fact an access point to a local network, and the NAS should start announcing accessibility of that local network (123.45.67.64/26) through the local routing procedures (an IGP such as RIP, OSPF or EIGRP).

Note that the current design makes the hypothesis that the Call Agent "tells" the address of the LAN to the NAS. This is a very debatable design. If a secure IGP is used (e.g. using embedded keyed MD5 authentication, or using IPSEC) then the routing prefix will be naturally exchanged through this IGP. On the other hand, some form of configuration can provide a "double check" against user errors.

In our example, the call is completed when the called modem hangs up. This triggers an ISUP release message, which is forwarded to the Call Agent. The Call Agent will request the NAS to delete the connection:

```
        DLCX 1244 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I: FDE234C8
```

The gateways will respond with a message that should include a "call parameters" header fields:

```
250 1244 OK
P: PS=1245, OS=62345, PR=780, OR=45123
```

We should note that, because this is a data call, the call parameters only include a count of the packets and octets that were sent and received.

5.3. Call back, using a NAS

There are three classic forms of call-back:

1- ANI-based Callback

2- PPP Callback (Microsoft Callback is a variant of this)

3- Login-based callback

The ANI based call-back can be implemented entirely in the Call Agent, as indicated in the following diagram:

```
   PC      CO    SS7/   NAS              CA               ACC
                 ISUP

  dials    IAM    ->
                 IAM    - -              ->
                                  Notices that the
                              called number corresponds
                               to a call back service,
                                and that the calling
                               number has subscribed
                                  to that service.
                                   Terminates the
                                   incoming call.
                  <-    - -              REL
           <-    REL
           RLC    ->
 hangup          RLC    - -              ->
                                  Decides to place
                                  an outgoing call.
                                     Call start           ->
                        <-             Create
                                  Connection (data)
                        Ack              ->
                  <-    - -              IAM
 (rings)   <-    IAM
```

```
|________|_____|______|_____|___________________________|_____|
```

The PPP callback suppose that the modem first establishes an incoming connection, and go through the authentication exchange. The following diagram provides an example of these exchanges:

```
 ____________________________________________________________________
|    PC   |  CO |  SS7/|     NAS    |        CA      |  ACC|  Radius|
|         |     |  ISUP|            |                |     |        |
|_________|_____|______|____________|________________|_____|________|
| dials in|     |      |            |                |     |        |
|         |  IAM|   -> |            |                |     |        |
|         |     |  IAM |     - -    |        ->      |     |        |
|         |     |      |            |  Checks called |     |        |
|         |     |      |            |     number.    |     |        |
|         |     |      |            |     Notices    |     |        |
|         |     |      |            |    data call.  |     |        |
|         |     |      |            |    Call start  |  -> |        |
|         |     |      |      <-    |      Create    |     |        |
|         |     |      |            |    Connection  |     |        |
|         |     |      |            |      (data)    |     |        |
|         |     |      |     Ack    |        ->      |     |        |
|         |     |      |            |    Connection  |     |        |
|         |     |      |            |    completed.  |     |        |
|         |     |      |            |      Call      |     |        |
|         |     |      |            |   established  |  -> |        |
|         |     |   <- |     - -    |       ANM      |     |        |
|         |  <- |  ANM |            |                |     |        |
|  modem  |  - -|  - - |      ->    |                |     |        |
|   <-    |  - -|  - - |  handshake |                |     |        |
|   PPP   |  - -|  - - |      ->    |                |     |        |
|         |     |      |   obtain   |                |     |        |
|         |     |      |  user-id,  |                |     |        |
|         |     |      |   password |                |     |        |
|         |     |      |    Check   |       - -      |  - -|    ->  |
|         |     |      |      <-    |       - -      |  - -|   Ack  |
|         |     |      |   reports  |                |     |        |
|         |     |      |  call back |                |     |        |
|         |     |      |  condition |                |     |        |
|         |     |      |     NTFY   |        ->      |     |        |
|         |     |      |      <-    |       ACK      |     |        |
|         |     |      |            |     Decides    |     |        |
|         |     |      |            |   to place an  |     |        |
|         |     |      |            |  outgoing call.|     |        |
|         |     |      |      <-    |      Delete    |     |        |
|         |     |      |            |    Connection  |     |        |
|         |     |      |    Perf    |                |     |        |
|         |     |      |     data   |        ->      |     |        |
|         |     |   <- |     - -    |       REL      |     |        |
|         |  <- |  REL |            |                |     |        |
|         |  REL|   -> |            |                |     |        |
|  hangup |     |  REL |     - -    |        ->      |     |        |
|_________|_____|______|____________|________________|_____|________|
 __________________________________________________________________________
|     PC     |  CO |  SS7/|       NAS        |      CA     |  ACC|  Radius|
|            |     |  ISUP|                  |             |     |        |
|____________|_____|______|__________________|_____________|_____|________|
|            |     |      |                  |  Call start |  -> |        |
|            |     |      |         <-       |    Create   |     |        |
|            |     |      |                  |  Connection |     |        |
|            |     |      |                  |    (data)   |     |        |
|            |     |      |        Ack       |      ->     |     |        |
|            |     |   <- |        - -       |      IAM    |     |        |
|   (rings)  |  <- |  IAM |                  |             |     |        |
|            |  ACM|   -> |                  |             |     |        |
|            |     |  ACM |        - -       |      ->     |     |        |
|  (answer)  |  ANM|   -> |                  |             |     |        |
|            |     |  ANM |        - -       |      ->     |     |        |
|            |     |      |                  |  Connection |     |        |
|            |     |      |                  |  complete.  |     |        |
|            |     |      |                  |    Call     |     |        |
|            |     |      |                  |  established|  -> |        |
|     PPP    |  - -|  - - |         ->       |             |     |        |
|     <-     |  - -|  - - |  Validates call, |             |     |        |
| Connected  |     |      |                  |             |     |        |
|   to the   |     |      |                  |             |     |        |
|  Internet  |     |      |                  |             |     |        |
|            |     |      |                  |             |     |        |
|   Closes   |     |      |                  |             |     |        |
| connection.|     |      |                  |             |     |        |
|            |  REL|   -> |                  |             |     |        |
|            |     |  REL |        - -       |      ->     |     |        |
|            |     |      |         <-       |    Delete   |     |        |
|            |     |      |                  |  Connection |     |        |
|            |     |      |     Perf data    |      ->     |     |        |
|            |     |   <- |        - -       |      RLC    |     |        |
|            |  <- |  RLC |                  |             |     |        |
|            |     |      |                  |   Call end  |  -> |        |
|____________|_____|______|__________________|_____________|_____|________|
```

This diagram shows the exchange of messages during a call from a modem user to an Internet Service Provider, using a trunking gateway that doubles as a Network Access Server. During these exchanges the MGCP is used by the Call Agent to control the NAS.

Upon reception of the IAM message, the Call Agent notices that the called number corresponds to a data service. Using configuration databases, the Call Agent selects the type of modem parameters and authentication parameters that correspond to the called number and to the calling number. It uses this knowledge to send a connection command to the NAS, programming the incoming trunk:

```
CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
M: data
X: 0123456789B1
R: cl, cbk

v=0
m=nas/radius
c=radius.example.net
a=bearer:v.32
a=framing:ppp-asynch
a=dialed:18001234567
a=dialing:2345678901
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection. (There is no session description in the case of a data call.)

```
200 1237 OK
I: FDE234C8
```

The Call Agent, knowing that this is a data call, can immediately acknowledge the establishment of the connection, sending an ANM message back to the calling switch.

The DSP associated to the incoming trunk has been loaded with the specified modem code. Once the call is established, the modem of the calling PC will be synchronized with the modem associated to the trunk. The caller will then proceed to a normal PPP synchronization, which probably implies a PPP login. The login parameters, in our example, are checked using Radius. The Radius server that will be used is typically chosen as a function of the called number, which identifies the data service that the calling modem requested. In fact, the number can also be used to identify the specific form of authentication that is requested.

In the call back example, the Radius server will indicate that the call cannot be completed as such, and that the user should be called back (for example, using a "Callback Framed" service type in its access-accept response.) The NAS will thus send a Notify message to the Call Agent, indicating that a call-back is requested:

```
        NTFY 2005 card23/21@trgw-7.whatever.net MGCP 0.1
        X: 0123456789B1
        O: cbk(user-id)
```

After this notification, the Call Agent should send an acknowledgement:

```
        200 2005 OK
```

The Call Agent will check that the call back request can be followed through. In its databases, it will find the regular address associated to the "user-id," and prepare to set up a call to that address. It will first clear the incoming call, sending a DeleteConnection command to the NAS:

In our example, the call is completed when the calling modem hangs up. This triggers an ISUP release message, which is forwarded to the Call Agent. The Call Agent will request the NAS to delete the connection, and reset the list of observed events:

```
        DLCX 1244 card23/21@trgw-7.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I: FDE234C8
        X: 0123456789B2
        R:
```

The gateways will respond with a message that should include a "call parameters" header fields:

```
        250 1244 OK
        P: PS=2, OS=345, PR=1, OR=123
```

We should note that, because this is a data call, the call parameters only include a count of the packets and octets that were sent and received.

The Call Agent will then proceed to set up an outgoing data call. This call may be routed through the same NAS that received the incoming call, but can also be routed through an entirely different endpoint , for example if the calling user has moved out of its normal region.

5.4. Data call to a NAS, using L2TP

```
 _______________________________________________________________________
|    PC     | CO  |  SS7/|      NAS      |      CA      |  ACC|   LNS   |
|           |     |  ISUP|               |              |     |         |
|___________|_____|______|_______________|______________|_____|_________|
|  dials in |     |      |               |              |     |         |
|           |  IAM|   -> |               |              |     |         |
|           |     |  IAM |      - -      |      ->      |     |         |
|           |     |      |               |  Check called|     |         |
|           |     |      |               |   number.    |     |         |
|           |     |      |               |   Notices    |     |         |
|           |     |      |               |  data call.  |     |         |
|           |     |      |               |  Call start  |  -> |         |
|           |     |      |      <-       |    Create    |     |         |
|           |     |      |               |  Connection  |     |         |
|           |     |      |               |    (data)    |     |         |
|           |     |      |      Ack      |      ->      |     |         |
|           |     |      |               |  Connection  |     |         |
|           |     |      |               |  complete.   |     |         |
|           |     |      |               |     Call     |     |         |
|           |     |      |               |  established |  -> |         |
|           |     |   <- |      - -      |     ANM      |     |         |
|           |  <- |  ANM |               |              |     |         |
|   modem   |  - -|  - - |      ->       |              |     |         |
|     <-    |  - -|  - - |   handshake   |              |     |         |
|    PPP    |  - -|  - - |      ->       |              |     |         |
|           |     |      |    obtain     |              |     |         |
|           |     |      |   user-id,    |              |     |         |
|           |     |      |   password    |              |     |         |
|           |     |      |   Establish   |              |     |         |
|           |     |      |    Tunnel     |              |     |         |
|           |     |      |    SCC-REQ    |     - -      |  - -|    ->   |
|           |     |      |      <-       |     - -      |  - -|  SCC-REP|
|           |     |      |      <-       |     - -      |  - -|  SCC-CON|
|           |     |      |    IC-REQ     |     - -      |  - -|    ->   |
|           |     |      |      <-       |     - -      |  - -|  IC-REP |
|           |     |      |      <-       |     - -      |  - -|  IC-CON |
|           |     |      |  Spoof PPP/LCP|     - -      |  - -|    ->   |
|     <-    |  - -|  - - |  Relays PPP   |     - -      |  - -|    ->   |
| Connected |     |      |               |              |     |         |
|  to the   |     |      |               |              |     |         |
|  Internet |     |      |               |              |     |         |
|___________|_____|______|_______________|______________|_____|_________|
```

PC	CO	SS7/ ISUP	NAS	CA	ACC	LNS
Closes						
connection.						
	REL	->				
		REL	- -	->		
			<-	Delete		
				Connection		
			Perf			
			data	->		
		<-	- -	RLC		
	<-	RLC				
			CDN	- -	- -	->
			Stop-CC-N	- -	- -	->
				Call end	->	

This diagram shows the exchange of messages during a call from a modem user to an Internet Service Provider, using a trunking gateway that doubles as a Network Access Server. During these exchanges the MGCP is used by the Call Agent to control the NAS. The PPP information is relayed to a network server (LNS) using L2TP.

Upon reception of the IAM message, the Call Agent notices that the called number corresponds to a data service. Using configuration databases, the Call Agent selects the type of modem parameters and authentication parameters that correspond to the called number and to the calling number. It uses this knowledge to send a connection command to the NAS, programming the incoming trunk:

```
CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
M: data
X: 0123456789B1
R: cl

v=0
c=IN IP4 access.example.net
m=nas/l2tp
k=clear:some-shared-secret
a=bearer:v.32
```

```
a=framing:ppp-asynch
a=dialed:18001234567
a=dialing:2345678901
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection. (There is no need to transmit a session description in the case of a data call.)

```
200 1237 OK
I: FDE234C8
```

The Call Agent, knowing that this is a data call, can immediately acknowledge the establishment of the connection, sending an ANM message back to the calling switch.

The DSP associated to the incoming trunk has been loaded with the specified modem code. Once the call is established, the modem of the calling PC will be synchronized with the modem associated to the trunk. The caller will then proceed to a normal PPP synchronization, which probably implies a PPP login.

Once PPP has been properly synchronized, the NAS establishes a tunnel towards the LNS. Because L2TP is a two-layer protocol, the NAS must first establish an L2TP control connection between itself and the LNS. This connection may or may not have been established prior to the call set-up.

Tunnel establishment requires a shared secret between the LNS and the NAS; in our example, that secret is passed by the Call Agent, along with the name of the LNS. Once the supporting tunnel is installed, the NAS has to establish an L2TP tunnel, to relay the "incoming call." Once the call is established, the PPP packets received on the trunk will be relayed over the L2TP tunnel, and vice-versa.

In our example, the call is completed when the calling modem hangs up. This triggers an ISUP release message, which is forwarded to the Call Agent. The Call Agent will request the NAS to delete the connection:

```
DLCX 1244 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
```

The gateways will respond with a message that should include a "call parameters" header fields:

```
250 1244 OK
P: PS=1245, OS=62345, PR=780, OR=45123
```

We should note that, because this is a data call, the call parameters only include a count of the packets and octets that were sent and received.

5.5. Basic data call with continuity test

PC	CO	SS7/	NAS	CA	ACC	Radius
		ISUP				
dials in						
	IAM	->				
		IAM	- -	->		
				Check called		
				number.		
				Notices		
				data call,		
				continuity		
				test.		
				Call start	->	
			<-	Create		
				Connection		
				(loopback)		
			Ack	->		
	COT	->				
		COT	- -	->		
			<-	Modify		
				Connection		
				(data)		
			Ack	->		
				Connection		
				is completed.		
				Call		
				established	->	
		<-	- -	ANM		
	<-	ANM				
modem	- -	- -	->			
<-	- -	- -	handshake			
PPP	- -	- -	->			

This diagram shows the various exchange of messages during the beginning of a call from a data user on the circuit-switched PSTN to a NAS. During these exchanges the Call Agent uses MGCP to control the NAS and the residential gateway. The circuit switch decides to execute a continuity test during the call establishment. The exchanges occur on two sides.

Upon reception of the IAM message, the Call Agent recognizes that a continuity test has been requested. It immediately sends a CreateConnection request to the NAS to connect to the incoming trunk, creating a connection:

```
CRCX 1237 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: loopback
X: 0123456789B1
R: cl

v=0
m=nas/radius
c=IN IP4 radius.example.net
a=bearer:v.32
a=framing:ppp-asynch
a=dialed:18001234567
a=dialing:2345678901
```

The trunking gateway recognizes that the mode is set to "loopback". It places the circuit in "loopback" mode (we assume that this is the adequate way to perform continuity test in this specific environment). The gateway is then ready to send back a 2010 Hz return tone if it receives a 2010 Hz test tone. The gateway acknowledges the creation of the connection, sending back the identification of the newly created connection:

```
200 1237 OK
I: FDE234C8
```

At this point, the call agent is waiting for the result of the continuity test. The calling switch is sending the test tone (2010 Hz); if it receives the return tone (2010 Hz), it will send a "continuity passed" message (COT). At this point, the call agent will send a modify connection message to the NAS, in order to place the connection in "data" mode:

```
MDCX 1238 card23/21@trgw-7.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
M: data
```

The NAS will immediately acknowledge that command:

```
200 1238 OK
```

The NAS will then proceed with the establishment of the modem call.

6. Audit and Restart

The following call flows provide examples of the audit and restart commands.

6.1. Using the Audit commands

```
 _________________________________________________________________
| CO |  SS7|  TGW  |        CA        |  CDB  |  ACC  |  RGW|  Usr|
|____|_____|_______|__________________|_______|_______|_____|_____|
|    |     |       |                  |       |       |     |     |
|    |     |   <-  |   Audit Endpoint |       |       |     |     |
|    |     |       |    (endpoints)   |       |       |     |     |
|    |     |   Ack |         ->       |       |       |     |     |
|    |     |   <-  |        Audit     |       |       |     |     |
|    |     |       |     Endpoint     |       |       |     |     |
|    |     |       |   (capabilities) |       |       |     |     |
|    |     |   Ack |         ->       |       |       |     |     |
|    |     |       |                  |       |       |     |     |
|    |     |       |  Audit Endpoint  |  - - -|  - - -|  -> |     |
|    |     |       |    (endpoints)   |       |       |     |     |
|    |     |       |         <-       |  - - -|  - - -|  Ack|     |
|    |     |       |  Audit Endpoint  |  - - -|  - - -|  -> |     |
|    |     |       |   (capabilities) |       |       |     |     |
|    |     |       |         <-       |  - - -|  - - -|  Ack|     |
|    |     |       |                  |       |       |     |     |
| IAM|  -> |       |                  |       |       |     |     |
|    |  IAM|  - - -|         ->       |       |       |     |     |
|    |     |       |   (proceed with  |       |       |     |     |
|    |     |       |    call setup)   |       |       |     |     |
|    |     |       |        ...       |       |       |     |     |
|    |  <- |  - - -|        ANM       |       |       |     |     |
| <- |  ANM|       |                  |       |       |     |     |
|    |     |       |                  |       |       |     |     |
|    |     |   <-  |  Audit Endpoint  |       |       |     |     |
|    |     |       |        (misc)    |       |       |     |     |
|    |     |   Ack |         ->       |       |       |     |     |
|    |     |   <-  |  Audit Connection|       |       |     |     |
|    |     |   Ack |         ->       |       |       |     |     |
|    |     |       |                  |       |       |     |     |
|    |     |       |   Audit Endpoint |       |       |     |     |
|    |     |       |        (misc)    |  - - -|  - - -|  -> |     |
|    |     |       |         <-       |  - - -|  - - -|  Ack|     |
|    |     |       |  Audit Connection|  - - -|  - - -|  -> |     |
|    |     |       |         <-       |  - - -|  - - -|  Ack|     |
|    |     |       |                  |       |       |     |     |
|____|_____|_______|__________________|_______|_______|_____|_____|
```

This diagram shows the various exchanges of messages during auditing of a trunk gateway and a residential gateway. First, both gateways are auditing to learn about the endpoints supported by each. Secondly, capabilities information is audited for one of these endpoints. The procedure is carried out for the residential gateway as well. A call then arrives from the PSTN and is established exactly as described in the "basic call from TGW to RGW". After the call has been established, both endpoints involved in the call are audited -- this time in order to retrieve endpoint info and subsequently connection info associated with the call.

The Call Agent initially sends an AuditEndpoint command to the trunking gateway in order to discover what endpoints the trunking gateway has:

```
AUEP 1200 *@trgw-7.whatever.net MGCP 0.1
F:
```

Since we use the wildcard naming convention, we cannot retrieve any endpoint specific information but the RequestedInfo field must still be included. Had we specified any RequestedInfo, the trunking gateway would simply have ignored it.

The trunking gateway immediately acknowledges the auditing command and sends back the list of endpoints it contains. Our trunking gateway has two cards in it -- card23 and card24 each supporting two circuits:

```
200 1200 OK
Z: card23/20@trgw-7.whatever.net
Z: card23/21@trgw-7.whatever.net
Z: card24/20@trgw-7.whatever.net
Z: card24/21@trgw-7.whatever.net
```

Now that the Call Agent has learned about the endpoints present in the trunking gateway, it requests capability information for one of the endpoints. We here assume that all similar endpoints have the same capabilities and thus only audit one of them for capabilities, although it is possible that different endpoints have different capabilities:

```
AUEP 1201 card23/20@trgw-7.whatever.net
```

```
F: A
```

The trunking gateway acknowledges the command by returning to the Call Agent a response describing the capabilities it supports for the endpoint in question:

```
200 1201 OK
L: a:G.711;G.726-32, p:10-100, e:on, s:off, v:T;D,
   m:sendonly;recvonly;sendrecv;inactive;conttest
```

The Call Agent thereby learns, that the endpoint supports two codecs; G.711 and G.726-32 which can each be used with a packetization period of between 10 and 100 msec. Echo cancellation is supported while silence suppression is not, and the event packages supported are Trunk and DTMF. Since Trunk is the first package specified it is furthermore the default package. Also, connections can be established in either of the modes "Send Only", "Receive Only", "Send/Receive", "Inactive", and "Continuity Test". Finally, several parameters were not specified and default or deduced values will therefore apply. These parameters are Bandwidth (deduce from codec), Gain Control (supported), Type of Service (supported), Resource Reservation (best effort), Encryption Key (no encryption), or Type of Network (guess) was provided and default or deduced values for each of these will therefore apply. Next the Call Agent queries the residential gateway to discover what endpoints are present in it:

```
AUEP 2000 *@rgw.whatever.net MGCP 0.1
F:
```

As before, we use the wildcard naming convention and can therefore not retrieve any endpoint specific information, however the RequestedInfo field must still be included. If any values had been specified for it, they would simply be ignored.

The residential gateway acknowledges the command and includes a list of endpoints it contains:

```
200 2000 OK
Z: endpoint/1@rgw-2567.whatever.net
Z: endpoint/2@rgw-2567.whatever.net
```

The Call Agent thereby learns, that the residential gateway contains the two endpoints endpoint/1 and endpoint/2. Having learned about the endpoints present, the Call Agent next requests capability information for one of the endpoints. Again, we assume that all similar endpoints have the same capabilities and thus only audit one of them for capabilities, although it is possible that different endpoints have different capabilities:

```
AUEP 2001 endpoint/1@rgw-2567.whatever.net
F: A
```

The residential gateway acknowledges the command by returning to the Call Agent a response describing the capabilities it supports for the endpoint in question:

```
200 1201 OK
L: a:G.711;G.726-32, p:10-100, e:on, s:off, v:L;D,
   m:sendonly;recvonly;sendrecv;inactive
L: a:G.723.1; p:30-90, e:on, s:on, v:L;D,
   m:sendonly;recvonly;sendrecv;inactive;confrnce
```

The Call Agent thereby learns, that the endpoint supports three codecs; G.711, G.726-32, and G.723.1.

G.711 and G.726-32 can each be used with a packetization period of between 10 and 100 msec. Echo cancellation is supported while silence suppression is not, and the event packages supported are Line and DTMF. Since Line is the first package specified it is furthermore the default package. Also, connections can be established in either of the modes "Send Only", "Receive Only", "Send/Receive", and "Inactive". Finally, several parameters were not specified and default or deduced values will therefore apply. These parameters are Bandwidth (deduce from codec), Gain Control (supported), Type of Service (supported), Resource Reservation (best effort), Encryption Key (no encryption), or Type of Network (guess) was provided and default or deduced values for each of these will therefore apply.

G.723.1 can be used with a packetization period of between 30 and 90 msec. Both echo cancellation and silence suppression are supported, and the event packages supported are Line and DTMF with Line being the default package. Connections can be established in either of the modes "Send Only", "Receive Only", "Send/Receive", "Inactive", and "Conference". Finally, default or deduced values will be applied for the parameters that were not supplied.

The Call Agent now knows all endpoints as well as their capabilities in the trunking and the residential gateway.

An IAM signaling a call for the residential gateway now arrives from the PSTN. The call is then setup as described in Section 2.2. After the call has been successfully setup, the Call Agent now decides to audit the endpoint in the trunking gateway for information about RequestedEvents, DigitMap, SignalRequests, RequestIdentifier, NotifiedEntity, ConnectionIdentifiers, and DetectEvents:

```
AUEP 1202 card23/21@trgw-7.whatever.net MGCP 0.1
F: R,D,S,X,N,I,T
```

The trunking gateway acknowledges the command by returning to the Call Agent a response with the requested info for the endpoint in question:

```
200 1202 OK
R:
D:
S:
X:
N: [128.96.41.12]
I: FDE234C8, ABCD123
T:
```

The Call Agent thus sees, that there are currently no RequestedEvents, no DigitMap, no SignalRequests, no RequestIdentifier, and no DetectEvents for the endpoint in question. Instead of supplying empty list for these parameters, the gateway could simply have omitted them altogether in the response.

In the last command the call agent sent to the endpoint, it had not specified a NotifiedEntity, thus the notified entity returned here is the source IP address of the call agent encoded in RFC 821 format. Finally, the endpoint currently has two connections associated with it.

The first one was created during the call setup. Depending on previous message exchanges, the second one may or may not be valid. If the call agent believes it is stale, it could simply instruct the gateway to delete it.

In this case, the Call Agent decides to audit the connection FDE234C8 for further information and it therefore sends an AuditConnection command to the endpoint for information about CallId, NotifiedEntity, LocalConnectionOptions, Mode, LocalConnectionDescriptor, RemoteConnectionDescriptor, and ConnectionParameters:

```
AUCX 1203 card23/21@trgw-7.whatever.net MGCP 0.1
I: FDE234C8
F: C,N,L,M,LD,RD,P
```

When the trunking gateway receives the command it immediately sends a response to the Call Agent with the requested info:

```
200 1203 OK
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:G.711;G.726-32
M: sendrecv
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48

v=0
c=IN IP4 128.96.41.1
m=audio 1296 RTP/AVP 0
v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

Thus the CallId, LocalConnectionOptions, and Mode are as expected for this connection. The same holds for the RemoteConnectionDescriptor which is supplied as the last parameter separated by an empty line. The last connection handling command issued to the endpoint for this connection did not include the NotifiedEntity parameter, so the notified entity defaults to the source IP address for that command which is encoded in RFC 821 format. Finally, the Call Agent also obtained the current packet statistics for the call.

Next, the Call Agent audits the endpoint in the residential gateway for information about RequestedEvents, DigitMap, SignalRequests, RequestIdentifier, NotifiedEntity, ConnectionIdentifiers, and DetectEvents:

```
AUEP 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
F: R,D,S,X,N,I,T
```

The trunking gateway acknowledges the command by returning to the Call Agent a response with the requested info for the endpoint in question:

```
200 2002 OK
R: L/hu
D: (0T|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
S:
X: 0123456789B1
N: [128.96.41.12]
I: 32F345E2
T: hu
```

The Call Agent thus sees, that currently, the only event being looked for is the "On hook" event from the Line package. Since the Line package is the default package, the gateway could have simply specified this as "hu" instead. Although the residential gateway is currently not accumulating any digits, a digit map is still supplied. This digit map is the last one used by the endpoint used -- we here assume the endpoint was previously originated a call, e.g. as in Section 2.1. There are currently no signals being applied so the SignalRequests parameter is simply empty. There is however an active NotificationRequest thus the RequestIdentifier for that one is supplied. As before, no NotifiedEntity has been specified for the last NotificationRequest for this endpoint, so the source IP address of that request is supplied as the notified entity. There is only one ConnectionId for this endpoint, namely 32F345E2 as expected. Finally, since the last NotificationRequest did not specify any special value for DetectEvents, this parameter simply defaults to the same as RequestedEvents. In this case we omitted the specification of the Line package in the event name since the Line package is the default package for this endpoint.

Having audited the endpoint, the Call Agent decides to audit the connection for that endpoint and the Call Agent therefore sends an AuditConnection command to the requesting information about CallId, NotifiedEntity, LocalConnectionOptions, Mode, LocalConnectionDescriptor, RemoteConnectionDescriptor and ConnectionParameters.

```
AUCX 2003 endpoint/1@rgw-2567.whatever.net MGCP 0.1
I: 32F345E2
F: C,N,L,M,LD,RD,P
```

When the residential gateway receives the command it immediately sends a response to the Call Agent with the requested info:

```
200 2003 OK
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:G.711;G.726-32
M: sendrecv

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000

v=0
c=IN IP4 128.96.41.1
m=audio 1296 RTP/AVP 0
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48
```

Thus the CallId, LocalConnectionOptions, and Mode are as expected for this connection. The same holds for the LocalConnectionDescriptor which is supplied as the last parameter separated by an empty line. The last connection handling command issued to the endpoint for this connection did not include the NotifiedEntity parameter, so the notified entity defaults to the source IP address for that command which is encoded in RFC 821 format. Finally, the Call Agent also obtained the current packet statistics for the call.

6.2. Using the RestartInProgress command

```
 __________________________________________________________________
| Management System       |           TGW          |  CA-1 |  CA-2|
|_________________________|________________________|_______|______|
|                         |                        |       |      |
| Preparing for taking    |                        |       |      |
| endpoints out of        |                        |       |      |
| service when idle.      |           RSIP         |   ->  |      |
|                         |  (graceful: indefinite)|       |      |
|                         |            <-          |   Ack |      |
|                         |                        |       |      |
| Preparing to take       |                        |       |      |
| endpoints out of        |                        |       |      |
| service in 5 minutes    |           RSIP         |   ->  |      |
| (graceful: 5 min)       |                        |       |      |
|                         |            <-          |   Ack |      |
|                         |                        |       |      |
| Circuit failure leads   |                        |       |      |
| to an endpoint becoming |                        |       |      |
| unavailable.            |           RSIP         |   ->  |      |
| (forced: 0 min)         |                        |       |      |
|                         |            <-          |   Ack |      |
|                         |                        |       |      |
| Courtesy message        |                        |       |      |
| that all endpoints are  |                        |       |      |
| now out of service      |           RSIP         |   ->  |      |
| (forced: 0 min)         |                        |       |      |
|                         |            <-          |   Ack |      |
|                         |                        |       |      |
| All endpoints back      |                        |       |      |
| in service              |           RSIP         |  - - -|   -> |
| (restart: 0 min)        |                        |       |      |
|                         |            <-          |  - - -|  Ack |
|                         |                        |       |      |
|_________________________|________________________|_______|______|
```

This diagram shows the various exchanges of messages during restart of some of the endpoints in a trunking gateway. Our example assumes the existence of an external management system controlling the gateway, and the resulting changes in endpoint availability are then communicated to the Call Agent via the RestartInProgress command. First all endpoints are to be taken out-of-service gracefully, and later a warning is sent to the Call Agent, that all endpoints will now be taken out of service within 5 minutes. Following this, we assume a problem occurs on the gateway resulting in the immediate unavailability of a circuit. A little later, the gateway then informs the Call Agent that all endpoints are now out of service. Finally, we assume that the trunking gateway rebooted and placed all endpoints back in service which the gateway informs its default Call Agent about.

The trunking gateway initially sends a RestartInProgress command to the Call Agent (CA-1) informing it of an intention to take all endpoints out of service:

```
        RSIP 1200 *@trgw-7.whatever.net MGCP 0.1
        RM: graceful
        RD: 0
```

The Call Agent thereby sees, that the trunking gateway is planning on making all its endpoints unavailable gracefully. Since the restart delay is specified to be zero, the Call Agent furthermore knows, that it can safely leave all existing connections on the gateway, however it should not attempt to establish any new connections for these endpoints -- or at least only establish connections related to existing calls.

The Call Agent immediately acknowledges the command:

```
        200 1200 OK
```

Later, the external management system decides that it will no longer wait indefinitely for the existing connections to cease to exist. The gateway therefore sends a RestartInProgress message to the Call Agent informing it, that all the endpoints will become unavailable within 5 minutes (300 seconds), and any connections existing at that point in time will be torn down:

```
        RSIP 1201 *@trgw-7.whatever.net MGCP 0.1
        RM: graceful
        RD: 300
```

The Call Agent immediately acknowledges the command:

```
        200 1201 OK
```

The Call Agent now has 5 minutes to tear down any existing connections.

Before the 5 minutes have elapsed, we imagine a hardware problem develops with one of the circuits on card23 in the trunking gateway and that the circuit immediately must be taken out service. The gateway informs the Call Agent about this by issuing the RestartInProgress command as follows:

```
        RSIP 1202 card23/20@trgw-7.whatever.net MGCP 0.1
        RM: forced
```

In this case, no restart delay was specified since a "forced" restart always takes effect immediately. If a restart delay had been specified, it would simply have been ignored. Any connections that existed for the endpoint will have been lost.

The Call Agent immediately acknowledges the command and notes that card23/20 is out of service:

```
        200 1202 OK
```

A little later, the 5 minutes grace period the Call Agent was notified about earlier has now passed, and - as a courtesy - the trunking gateway informs the Call Agent that all endpoints have now been taken out of service:

```
        RSIP 1203 *@trgw-7.whatever.net MGCP 0.1
        RM: forced
```

Although the Call Agent has already been informed that card23/20 is out of service, the trunking gateway includes it in the list of endpoints here anyway. This is perfectly valid since placing an out-of-service endpoint in out-of-service can be considered idempotent as long as the gateway deletes all connections associated with those endpoints (out-of-service endpoints should not have any connections created on them). However, this would not be the case with placing in-service endpoints in-service as such operations have side-effects on existing connections. In that case, the Call Agent would therefore assume, that endpoints already in-service had been restarted to be in-service again.

The Call Agent immediately acknowledges the command:

```
200 1203 OK
```

At this point, all endpoints in the trunking gateway are now out of service.

We then assume that the trunking gateway is rebooted and all endpoints are placed back in service. After the reboot, the trunking gateway informs its default Call Agent (CA-2), that all its endpoints are now back in service by sending the following RestartInProgress command:

```
RSIP 1204 *@trgw-7.whatever.net MGCP 0.1
RM: restart
RD: 0
```

Since the restart delay specified is zero, all endpoints are in-service at this point in time. However, it should be noted, that this does not necessarily imply that the same endpoints are available as before. For instance, card23 could have been removed from the trunking gateway to correct the aforementioned circuit problem.

The Call Agent (CA-2) recognizes that CA-1 is the preferred Call Agent for this trunking gateway and when CA-2 acknowedges the command, it therefore includes CA-1 as the NotifiedEntity. Furthermore, CA-2 notes that any connections that may have existed for these endpoints prior to receiving this command no longer exists. CA-2 is expected to communicate this information to CA-1 in order to achieve the internal Call Agent synchronization required:

```
200 1204 OK
N: CA-1@whatever.net
```

7. Using MGCP to control an IVR

This section describes the call flows between the CA,MG and IVR in order to understand the way MGCP organises the communication between the Media gateway controller/call agent and the Media gateway or an IVR.

The IVR is controlled by the call agent using only the existing MGCP primitives.

The number of calls that an IVR can support is defined as the number of endpoints on that IVR. These end points may be maintained by the IVR itself in which case the Call agent always uses wildcards for createconnection or it may be maintained by the Call agent. There are a variety of scripts that can be executed on a particular ivr endpoint.They may be as simple as just playing a message (in which case the IVR is used as a simple anouncement server) or playing a message collect digits and give it to the call agent or as complex as an administrative script which allows a remote administrator to configure the call agent .

The format of the script and the implementation of the script may be vendor specific.

The only assumtion is that the call agent is pre configured with the script names and it knows what script to use when.

The MGCP primitives are interpreted by the IVR in the following way

CreateConnection:
When this command is received by the IVR it allocates the resources and returns the RTP profile associated with the logical endpoint. The connection mode should always be inactive. Note that the IVR starts executing the script if the connection mode is not set to inactive.

ModifyConnection:
This is used by the Call agent to trigger the execution of the script. Here the connection mode is set to sendrecv.If it is set to sendonly the IVR can play message or tones but cannot collect dtmf.

NotificationRequest:

Notifies the Call agent when the script has finished executed and returns the result.For now the result is only in the form of digits.

DeleteConnection:
Stops executing the script if it is not already terminated and frees the resources.

NOTE : The call flows do not show the Delete connection on the Incoming side when the IVR terminates. This is because the incomming call may be routed to another destination by the call agent depending on the notification results got from the IVR.

7.1. Connecting a Residential Gateway to the IVR

7.1.1. Connection from RGW to IVR

USER	RGW	CA	IVR
	<--	RQNT	
	ACK	-->	
OFF HOOK	NTFY	-->	
	<--	ACK	
	<--	CRCX+RQNT	
	ACK	-->	
		CRCX+RQNT	-->
		<--	ACK
	<--	MDCX	
	ACK	-->	
		MDCX	-->
		<--	ACK

The first command is a NotificationRequest, sent by the Call Agent to the residential gateway. The request will consist of the following lines:

```
RQNT 1201 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hd
```

The gateway immediately acknowledges the command, repeating in the acknowledgement message the transaction id that the Call Agent attached to the query.

```
        200 1201 OK
```

When the off hook event is noticed the gateway will notify the event to the Call Agent:

```
        NTFY 2002 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        N: ca@ca1.whatever.net:5678
        X: 0123456789AC
        O: hd
```

The Call Agent immediately acknowledges that notification.

```
        200 2002 OK
```

The Call Agent will then seize the incoming circuit, creating a connec- tion. The create connection command piggybacks a notification request, to watch for an on-hook transition:

```
        CRCX 1204 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: recvonly
        X: D123456789AD
        R: hu
```

The gateway immediately acknowledges the creation, sending back the identification of the newly created connection and the session descrip- tion used to receive audio data:

```
        200 1204 OK
        I:FDE234C8

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The Call Agent, having seized the incoming trunk and deciding that the call has to be terminated on the IVR and that a script will be executed, sends a connection command to the IVR.

```
CRCX 1205 #@ivr45.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711;G.726-32
M: inactive

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The CreateConnection is sent to a generic endpoint, asking the IVR to pick one of its available ports.

The IVR will acknowledge the connection command, sending in the identification of the selected endpoint, the connection identifier and the session description its own parameters such as address, ports and RTP profile:

```
200 1205 OK
Z:17@ivr45.whatever.net
I:32F345E2

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The Call Agent will relay the information to the Media gateway, using a ModifyConnection command:

```
MDCX 1206 endpoint/1@rgw-2567.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
M: sendrecv

v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

The media gateway immediately acknowledges the modification:

```
200 1206 OK
```

At this point the caller is ready, a duplex path has been established between the caller and the IVR, all the resources have been allocated and the Call agent has to trigger the script execution. It does this by sending a ModificationRequest with an embedded NotificationRequest command to the IVR.

```
MDCX 1207 17@ivr45.whatever.net MGCP 0.1
C: A3C47F21456789F0
M: sendrecv
X: D23456789AD
R: Script/oc, Script/of
D: Script/perl(http://database.whatever.net/script-23.prl)
```

The IVR immediately acknowledges the modification:

```
200 1207 OK
```

At this point the caller is interacting with the IVR. This ends with either the caller going on hook or the IVR deciding it has to notify the Call agent.

7.1.2. Disconnecting the user from IVR:(termination by IVR)

```
 ____________________________________
| USER |   RGW |     CA    |    IVR   |
|______|_______|___________|__________|
|      |       |    <--    |    NTFY  |
|      |       |    ACK    |    -->   |
|      |       |   DLCX--  |    -->   |
|      |       |    <--    |   ---ACK|
|______|_______|___________|__________|
```

When the call agent receives

```
NTFY 2002 17@ivr45.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: D23456789AD
```

```
O: script/oc(54321)
```

The Call agent immediately acknowledges the notification:

```
200 2002 OK
```

and send a delete connection to the IVR

```
DLCX 1211 script23/21@ivr45.whatever.net MGCP 0.1
C: 567F21456789F0
I:32F345E2
```

the IVR frees up the resources and responds with

```
200 1211 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48
```

The call agent may now deal with the incoming call by sending a delete connection, execute another script on the IVR, or route the call to another gateway, and so on.

7.1.3. Disconnecting The User From Ivr:(Termination By Caller)

```
 ___________________________________
|  USER    |   RGW |    CA   |   IVR |
|__________|_______|_________|_______|
| On hook  |   NTFY|   -->   |       |
|          |   <-- |    ACK  |       |
|          |       |   DLCX  |   --> |
|          |       |   <--   |   ACK |
|          |   <-- |   DLCX  |       |
|          |   ACK |     --> |       |
|__________|_______|_________|_______|
```

When the call agent receives

```
NTFY 2056 endpoint/1@rgw-2567.whatever.net MGCP 0.1
N: ca@ca1.whatever.net:5678
X: D123456789AD
O: hu
```

it responds with

```
        200 2056 OK
```

and deletes the connections on both the IVR

```
        DLCX 1211 17@ivr45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:32F345E2
```

IVR frees up the resources and responds with

```
        200 1211 OK
        P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48
```

and the media gateway

```
        DLCX 1261 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:FDE234C8
```

Gateway frees up the resources and responds with

```
        200 1261 OK
        P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48
```

7.2. Connection between a TGW and an IVR

7.2.1. Setting up the connection from TGW to IVR

The figure below gives the flow for the IVR connected to a TGW

```
  ___________________________________________
 | SS7/ISUP |   TGW |   CA            |   IVR |
 |__________|_______|_________________|_______|
 | IAM      |  ---  |   -->           |       |
 |          |   <-- |   CRCX          |       |
 |          |   ACK |   -->           |       |
 | <--      |   --- |   ACM           |       |
 |          |       |   CRCX+RQNT-->|         |
 |          |   <-- |   ---ACK        |       |
 |          |   <-- |   -MDCX         |       |
 |          |   ACK |   -->           |       |
 | <--      |   --- |   ANM           |       |
 |          |       |   MDCX          |   --> |
 |          |   <-- |   ----ACK       |       |
 |__________|_______|_________________|_______|
```

When the CA detects an incoming call it sends a CreateConnection command to the media gateway.

```
        CRCX 1204 ss7end/1@tgw90.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: recvonly
```

The media gateway responds with

```
        200 1204 OK
        I:FDE234C8

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The CA then creates a connection on the IVR

```
        CRCX 1205 #@ivr45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        L: p:10, a:G.711;G.726-32
        M: inactive

        v=0
        c=IN IP4 128.96.41.1
        m=audio 3456 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The IVR will acknowledge the connection command, sending in the session description its own parameters such as address, ports and RTP profile:

```
        200 1205 OK
        Z: 17@ivr45.whatever.net
        I:32F345E2

        v=0
        c=IN IP4 128.96.63.25
        m=audio 1296 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The Call Agent will relay the information to the Media gateway, using a ModifyConnection command:

```
        MDCX 1206 endpoint/1@rgw-2567.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:FDE234C8
        M: sendrecv

        v=0
        c=IN IP4 128.96.63.25
        m=audio 1296 RTP/AVP 0 96
        a=rtpmap:96 G726-32/8000
```

The media gateway immediately acknowledges the modification:

```
        200 1206 OK
```

At this point the caller is ready,a duplex path has been established between the caller and the IVR ,all the resources have been allocated and the Call agent has to trigger the script execution. It does this by sending a ModifyConnection command to the IVR, with an embedded notification request.

```
        MDCX 1206 17@ivr45.whatever.net MGCP 0.1
        C: A3C47F21456789F0
        I:32F345E2
        X: D23456789AD
        M: sendrecv
        R: Script/oc, Script/of
        D: Script/perl(http://database.whatever.net/script-12.prl)
```

The IVR immediately acknowledges the modification:

```
        200 1206 OK
```

At this point the caller is interacting with the IVR.This ends with either the caller going on hook or the IVR deciding it has to notify the Call agent.

7.2.2. Disconnecting The User From IVR:TGW

Termination by the IVR:

```
  ________________________________________
 | SS7/ISUP |   TGW |     CA     |   IVR  |
 |__________|_______|____________|________|
 |          |       |    <--     |  --NTFY|
 |          |       |   ACK--    |   -->  |
 |          |       |   DLCX--   |   -->  |
 |          |       |    <--     |  ---ACK|
 |__________|_______|____________|________|
```

7.2.3. Termination by the Caller

```
 _______________________________________________________
| SS7/ISUP |   TGW |            CA              |   IVR |
|    REL   |   --- |            -->             |       |
|          |       |                            |       |
|          |       |           DLCX             |   --> |
|          |       |           <--              |   ACK |
|          |   <-- |           -DLCX            |       |
|          |   ACK |            -->             |       |
|    <--   |   --- |            RLC             |       |
|__________|_______|____________________________|_______|
```

7.3. Hairpin IVR connection, double end-point model

The figure below gives the flow that results in an hairpin connection to an IVR device located on the same gateway as the trunk on which the call is incoming. In this flow, we assume that we use the "double endpoint" extensions to the create-connection command, and, as an example, assume that the IVR exchange results in an automatic disconnection.

```
 _________________________________________
| sw1|  sw2|  SG |   CA   |  TGW-1|   IVR |
|____|_____|_____|________|_______|_______|
| IAM|  - -|  -> |        |       |       |
|    |     |  IAM|  ->    |       |       |
|    |     |     |  CRCX+|   ->   |  (->) |
|    |     |     |  RQNT |   ->   |  (->) |
|    |     |     |    <- |   ACK  |  (ACK)|
|    |     |   <-|  ANM   |       |       |
|  <-|  - -|  ANM|        |       |       |
|    |     |     |    <- |   - -  |  NTFY |
|    |     |     |   ACK |   - -  |   ->  |
|    |     |     |  DLCX |   ->   |       |
|    |     |     |    <- |  Perf  |       |
|    |     |     |       |  data  |       |
|    |     |   <-|  REL   |       |       |
|  <-|  - -|  REL|        |       |       |
|    |     |     |  DLCX |        |   ->  |
|    |     |     |    <- |   - -  |  Perf |
|    |     |     |       |        |  data |
|____|_____|_____|________|_______|_______|
```

During these exchanges the MGCP is used by the Call Agent to control two endpoints located on the same TGW.

The exchanges start with the arrival from the first switch (SW1) of an SS7/ISUP "IAM" message, relayed by the signalling gateway to the Call Agent. The call agent performs the routing, and determines that the call will have to be relayed towards the second switch (SW2), using a trunk located on the same gateway.

The call agent starts the exchange by seizing the endpoint referenced in the IAM message, and by requesting a local connection between that endpoint and the IVR device. The command also instruct the IVR to start executing a script on the selected IVR endpoint:

```
CRCX 1204 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: nt=LOCAL
M: sendrecv
2/E: IVR/$
2/X: D23456789AD
2/R: Script/oc, Script/of
2/D: Script/perl(http://database.whatever.net/script-12.prl)
```

Upon reception of that command, the trunking gateway prepares a "local" connection description between the specified endpoint (trunk-group-1/17) and an endpoint that it selects within the IVR (IVR/6). The gateway acknowledges the two creations in a single message:

```
200 1204 OK
I:FDE234C8

v=0
c=IN tgw.whatever.net trunk-group-1/17
m=audio 0 LOCAL 0
2/Z:IVR/13@tgw.whatever.net
2/I:abc0
2/
v=0
c=IN tgw.whatever.net IVR/13
m=audio 0 LOCAL 0
```

The command has created a path between the endpoint and the IVR. Because the IVR is in fact answering the call, the call agent can immediately relay an ANM message to the calling switch.

At this point the caller is interacting with the IVR. This ends with either the caller going on hook or the IVR deciding it has to notify the Call agent.

```
NTFY 2001 IVR/13@tgw.whatever.net MGCP 0.1
X: D23456789AD
O: Script/oc(123245)
```

The call agent immediately acknowledges the notification:

```
200 2001 OK
```

The call agent should then remove the connections. The call agent releases the connection on the first endpoint:

```
DLCX 1207 trunk-group-1/17@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:FDE234C8
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1217 OK
P: OS=62345, OR=62345
```

The call agent can now notify the release of the trunk to the switch which will in response send an RLC message.

In parallel, the call agent releases the connection to the IVR:

```
DLCX 1208 IVR/13@tgw.whatever.net MGCP 0.1
C: A3C47F21456789F0
I:abc0
```

The gateway acknowledges the deletion, sending the connection parameters:

```
250 1218 OK
P: OS=62345, OR=62345
```

8. Acknowledgements

We want to thank here the many reviewers who helped design the MGCP flows, notably Ike Elliott and Chip Sharp.

9. References

[0] Arango, M., A. Dugan, I. Elliott, C. Huitema, S. Pickett, "Media Gateway Control Protocol (MGCP)", work in progress.

[1] ITU-T, Recommendation Q.761, "FUNCTIONAL DESCRIPTION OF THE ISDN USER PART OF SIGNALLING SYSTEM No. 7", (Malaga-Torremolinos, 1984; modified at Helsinki, 1993)

[2] ITU-T, Recommendation Q.762, "GENERAL FUNCTION OF MESSAGES AND SIGNALS OF THE ISDN USER PART OF SIGNALLING SYSTEM No. 7", (Malaga-Torremolinos, 1984; modified at Helsinki, 1993)

* ITU-T, Recommendation H.323, "VISUAL TELEPHONE SYSTEMS AND EQUIPMENT FOR LOCAL AREA NETWORKS WHICH PROVIDE A NON-GUARANTEED QUALITY OF SERVICE."

* ITU-T, Recommendation H.225, "Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communications Systems."

* ITU-T, Recommendation H.245, "LINE TRANSMISSION OF NON-TELEPHONE SIGNALS."

* Handley, Shulzrinne, Schooler, Rosenberg, "SIP: Session Initiation Protocol", work in progress

10. Authors’ Addresses

Christian Huitema
Bellcore
MCC 1J236B
445 South Street
Morristown, NJ 07960
U.S.A.

Phone: +1 973-829-4266
EMail: huitema@bellcore.com

Flemming Andreasen
Bellcore
RRC-1M223
444 Hoes Lane
Piscataway, NJ 08854-4157
U.S.A.

Phone: +1 732 699-7351
EMail: fandreas@notes.cc.bellcore.com

Mauricio Arango
RSL COM Latin America
6300 N.W. 5th Way, Suite 100
Ft. Lauderdale, FL 33309

Phone: (954) 492-0913
Email: marango@rslcom.com

Prakash K
TELESOFT INC
3000, Custer Road
Suite 270
Plano, Texas - 75075
U.S.A

Tel: 972-596-4158
Fax: 817-323-1605
EMail: prakashk@indts.com