Megaco/H.248 Call flow examples

Status of this Memo

This draft is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working drafts of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working drafts as Internet-Drafts. Internet-Drafts are draft drafts valid for a maximum of six months and may be updated, replaced, or obsoleted by other drafts at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

1. Abstract

This draft illustrates the usage of Megaco (version 1) protocol defined between the Media Gateway Controller and Media Gateway. In light of the vast features presently incorporated and continuously evolving features of the protocol, it serves the purpose of representing typical use case scenarios. There are a lot of possible scenarios for usage of MEGACO protocol. It is not the intent of the draft to represent the inter-working functionality among various protocols, however, an attempt is made to depict its usage in such a case for the purpose of completeness in the larger perspective. An attempt has been made to show the supplementary call services using MEGACO. The call flows can be categorized into two sub sections,

- MEGACO only call scenario
- MEGACO and other protocols

The draft begins with showing MEGACO only call scenarios, where the called and calling party lie on MEGACO domain. Various
permutations are possible even in this setup, viz
- RGW to RGW
- RGW to TGW
- TGW to TGW
- RGW/TGW to IVR

In the other case, typical cases of MEGACO interaction with other protocols have been depicted. Here it is assumed that the MG, which participates in the interaction, is RGW. This can be extended to any type of Media GW. The scenarios include

- MEGACO user with SIP
- MEGACO user with H.323
- MEGACO user with SS7
- MEGACO user with ISDN
- MEGACO user with R1
- MEGACO user with R2

The packages used in each of the calls flows are mentioned before each of the call flows. The packages that are addressed in this draft along with the packages defined in the base protocol also include packages like R2, R1, etc to illustrate the protocol usage. In case of Trunking gateways even though its not shown explicitly it is assumed that the messages from the CCS (Common Channel Signaling) switches are received by MGC through the Signaling Gateway. The emphasis of the draft is on the Megaco commands hence the messages between Signaling Gateway and MGC are not shown explicitly. Wherever applicable it should be assumed that the CCS switches/exchanges are communicating the messages to MGC through the Signaling Gateway.

One of the sections illustrates the usage of SDP for ATM. For simplicity residential gateway with ATM connectivity is assumed. However the same holds true for trunking gateways also. Two methods of call establishment with SDP for ATM are discussed, namely the "Backward Bearer Connection Set-up Model" and "Forward Bearer Connection Set-up Model".

This draft should be treated as only a means to illustrate the usage of Megaco but not as a guide for implementing Media Gateway or Media Gateway Controller. These calls flows are only informative. All the messages are encoded in the ABNF syntax for simplicity. The same calls flows are valid with binary messages also. Care has been taken to see that the messages are according to the protocol grammar, in case of discrepancies the protocol draft has to be considered. The Call flow diagrams are only a means to abstract the protocol messages exchanged between the MG and the MGC. These call flow diagrams are not according to any time scale. The IP addresses and port numbers used in the examples are fictitious. The statistic parameter values are also fictitious.
2. Internet Telephony Call Flows....................................4
2.1 Call between two residential gateways........................4
    Case (a)....................................................7
    Case (b)................................................................18
    Case (c)................................................................18
2.2 Call between Residential Gateway and Trunking Gateway....24
2.3 Call between Trunking gateway and Residential Gateway....34
2.4 Call between two Trunking gateways............................42
2.5 Call between Residential gateway and SS7 trunk in TGW....49
2.6 Call between SS7 trunk in TGW and residential gateway....58
2.7 Call between SS7 trunk in TGW and R2 trunk in TGW......67
2.8 Call between R2 trunk in TGW and SS7 trunk in TGW......76
2.9 Call between R1 trunk in TGW and SS7 trunk in TGW......85
2.10 Call between SS7 trunk in TGW and R1 trunk in TGW......94
2.11 Call between ISDN trunk in TGW and SS7 trunk in TGW...103
2.12 Continuity test from TGW......................................109
    Case (a)....................................................109
    Case (b)................................................................111
2.13 Call from residential gateway to H.323 Terminal........112
2.14 Call from residential gateway to SIP user................120
3 Service Change Command Usage....................................127
3.1 ROOT Termination................................................127
    3.1.1 Registration..............................................127
    3.1.2 Cold Start.................................................129
    3.1.3 Handoff....................................................131
    3.1.4 Disconnection.............................................133
3.2 Service Change for Termination................................136
    3.2.1 MG generated Service Change.............................136
    3.2.2 MGC generated Service Change............................143
4.0 Audit Command Usage.............................................146
    4.1 Audit Value..................................................146
    4.1.1 Audit value command on ROOT Termination.............146
    4.1.2 Audit value on non-ROOT terminations................148
4.2 Audit Capability..................................................150
    4.2.1 Audit Capability on ROOT termination................150
    4.2.2 Audit Capability on non-Root Terminations..........150
5. IVR using MEGACO..................................................151
    5.1 Connecting Residential gateway to IVR...................151
    5.2 Disconnecting Residential User from IVR...............158
5.3 Connecting Trunking Gateway to IVR..........................160
5.4 Disconnecting Trunking gateway from IVR..................164
6. Wildcard ContextID usage........................................166
7. Wildcard TerminationId Usage....................................167
8. Supplementary services support..................................168
    8.1 Call Transfer...............................................168
    8.2 Call waiting...............................................189
9. Conferencing.....................................................211
10.0 SDP for ATM..................................................243
    10.1.1 Forward Bearer Connection Set-up Method...........244
    10.1.2 Backward Bearer Connection Set-up Method.........257
1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

1.2 References:

4) Megaco mail archive http://standards.nortelnetworks.com/archives/megaco.html
5) R2 Package for Megaco Protocol, Kushanava Laha et al,"work in progress" <draft-ietf-megaco-r2package-02.txt>
6) V.Bajaj et al, Megaco/H.248 Basic CAS Packages,"Work in Progress" <draft-manyfolks-megaco-caspackage-00.txt>
7) Wendy et. al, draft-bothwell-megaco-mftonepkgs-01.txt, "Work in Progress", "MF Tone Generation and Detection Packages"
8) Rajesh Kumar et. al, Conventions for the use of the Session Description Protocol (SDP)for ATM Bearer Connections. "draft-ietf-mmusic-sdp-atm-05.txt", "Work in Progress".

2. Internet Telephony Call Flows

This section illustrates sample Internet telephone calls. Calls between Trunking gateway, Residential gateway, H.323 Endpoint, etc are illustrated. In all these call scenarios emphasis is given on the Megaco protocol messages rather than the remaining entities.2.1 Call between two residential gateways

2.1 Call between two Residential Gateways

The call establishment between two residential users is considered in this example. User A and User B are connected to two residential gateways RGW1 and RGW2. For simplicity we consider the case where the two MG’s are controlled by the same MGC. The call scenario assumes the implementation of analog line supervision package, RTP package, generic package, DTMF detection package, Call progress generator package, and the Network Package. Along with the successful call between the two users (case a), we also consider the case where the called party is busy (case c), and the call termination by both the users (case a for UserA terminated call flow and case b for UserB terminated call flow) is also discussed. In this example the registration of the MG (RGW) with the MGC is assumed to have happened as explained in section 3.1.1 Registration.
<table>
<thead>
<tr>
<th>USERA</th>
<th>RGW1</th>
<th>MGC</th>
<th>RGW2</th>
<th>USERB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserA offhook</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify offhook</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify Resp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify SG: dialtone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ED:al/on, dd/ce{Dmap1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DM:Dmap1 = 2XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dial Tone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Resp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Dials Digits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify digits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify Response</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add TermA SD: ringbacktone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add $, Local SDP Info - underspecified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RingBack Tone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Resp TermA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp Local SDP (Specified)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add TermB SD: Ring ED: offhook</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add $ Local (Underspecified)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote SDP (Specified)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserB Phone Ringing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp TermB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In all the telephone scenarios explained in this draft, once the call is terminated by either the Calling party or the called party, the other user listens a busy tone. A dial tone can be applied for the user to initiate another call. But for simplicity busy tone is applied so that the user goes onhook before initiating another call. It is assumed in the call scenarios that the registration of the MG with the MGC is done already. In Step 1 the MGC generates the Modify message towards both the Residential gateways to check for off hook on the terminations. (A wildcard command may also be used in this scenario but simplicity we consider only command to specific terminations). Modify message generated only for Residential gateway 1 is shown, similar message is sent to the other Residential gateway also. We are not considering the embedded signal and event descriptors here. Another call scenario will illustrate the use of embedded event and signal descriptors. The MGC in NULL context generates the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified here in the example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control descriptor.

Step 1
MGC to RGW1:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = Receiveonly}
      },
      Events = 1111 {al/of}
    }
  }
}
MG after receiving the command from MGC accepts it and responds with the transaction reply.

Step 2
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
  Context = - (Modify = TermA)
}
In this example User A goes off hook. This event is detected by the RGW1 and constructs and sends the Notify message towards the MGC. The MG uses the same request id (1111) sent by the MGC in its initial
command. The timestamp of the detected event is also passed as parameter to the observed event.

Step 3
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Transaction = 2000 {
  Context = - {
    Notify = TermA {ObservedEvents =1111 {
      20010202T10000000:al/of}}
  }
}

MGC generates the Notify response and responds with more messages towards the MG that generated the Notify command.

Step 4
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = TermA}
}

The MGC in the present example issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial pattern is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 5
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = - {
    Modify = TermA {
      Signals {cg/dt},
      DigitMap= Dmap1 {(2XXX)}
      Events = 1112 {
        al/on, dd/ce {DigitMap=Dmap1}
      },
    }
  }
}

MG after receiving the Modify command after validation responds to the MGC and starts processing the descriptors listed.

Step 6
MG1 to MGC:
The descriptors are processed in the order that is specified by the MGC. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the termination. The Digit map said to be ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself contains a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. It’s assumed that the user dials digits that match the pattern specified in the digit map. Lets assume that the user has dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is generated from MG1 to MGC. The MG again used the same request identifier as specified by the MGC.

Step 7
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
    Context = - {Notify = TermA {ObservedEvents =1112 {
        20010202T10010000:dd/ce {ds="2992", Meth=FM)}}
    }}
}

MGC after receiving the Notify command responds back with the Notify response.

Step 8
MGC to MG1:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2001 {
        Context = - {Notify = TermA}
    }

MGC after receiving the Notify command starts analyzing the dialed digits. In this example it is assumed that the called subscriber is connected to the RGW2, which is again controlled by the same MGC. The MGC generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. As the MGC is aware that the
destination user UserB is free it indicates MG1 to apply ringback tone to the termination of UserA. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created because of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used.

Step 9
MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
Context = $ {
    Add = TermA {
        Signals {cg/rt}
    } 
    Add = $ {
        Media {
            LocalControl {
                Mode = Receiveonly,
            },
            Local {
                v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
            }
        }
    }
}
}

In this example the connection fields IP address, the media field port number are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating the creation of a new context. In this example the MG creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. MG responds with the allocated IP address 209.110.59.33 and port number 30000.

Step 10
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
Context = 1 {
    Add = TermA,
    Add=EphA{
        Media {

    
}

Madhubabu, et al. [Page 10]
MGC generates a similar transaction towards the RGW2. The ContextID specified in the action is $. The first command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MGC has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the RGW2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MGC to MG2:
MEGACO/1 [216.33.33.61]:27000
Transaction = 1237 {
    Context = $ {
        Add = TermB { Media {
            LocalControl {Mode = Receiveonly} },
            Signals {al/ri}
            Events =1234{al/of {events = 1235 {al/on}}},
        },
        Add = $ {Media {
            LocalControl {
                Mode = Receiveonly,
            },
            Local {
                v=0
                c=IN IP4 $
                m=audio $ RTP/AVP 4
            },
            Remote {
                v=0
                c=IN IP4 209.110.59.33
                m=audio 30000 RTP/AVP 4
            } ; RTP profile for G.723 is 4
        }
    }
}

Madhubabu, et al.
MG2 after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists "ring" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates an ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example MG2 reserves IP address 207.176.47.90 and port number 40000. The MG2 responds to MGC with the following transaction reply.

Step 12
MG2 to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1237 {
  Context = 2 {
    Add = TermB,
    Add = EphB {
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      } ; RTP profile for G723 is 4
    }
  }
}

The MGC waits for the UserB to go offhook. Once the UserB goes offhook, MG2 reports the notification of the offhook event to the MGC.

Step 13
MG2 to MGC:
MEGACO/1 [207.176.47.89]: 26000
Transaction = 3000 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20000202T10020000:al/of}}
  }
}

The MGC responds to the MG2 with the Notify response.
Step 14
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 3000 {
  Context = 2 {Notify = TermB}
}
```

The MGC generates a transaction towards MG2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook event.

Step 15:
MG2 to MGC:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
  Context = 2 {
    Modify = TermB {
      Signals { }; to turn off ringing
      Events = 1235 {al/on},
      Media {
        LocalControl {
          Mode = SendRecv,
        }
      }
    }
    Modify = EphB {
      Media {
        LocalControl {
          Mode = SendRecv,
        }
      }
    }
  }
}
```

The MG2 responds to the request from MGC.

Step 16
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1238 {
  Context = 2 {Modify = TermB , Modify = EphB}
}
```

The MGC generates message to the MG1 to stop the ringback tone and to report the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MG2 to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
```
Transaction = 1239 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
        Signals { }
      }
      Modify = EphA {
        Media {
          LocalControl {
            Mode = sendrecv
          }
          Remote {
            v=0
c=IN IP4 207.176.47.90
m=audio 40000 RTP/AVP 4
          }
        }
      }
    }
  }
}

The empty signal descriptor in the Modify command for termination TermA, specifies to stop the ringback tone at the calling end. The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.

Step 18
MG1 to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1239 {
    Context = 1 {Modify = TermA, Modify = EphA}
  }

The two users can exchange the voice. The call can be termination either by the calling user or the called user. In this example it is assumed that the calling party has gone on-hook. The UserA after the conversation goes onhook indicating the tearing down of the call. The same is reported in the Notify command from MG1 to MGC.

Step 19
RGW1 to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Transaction = 2002 {
    Context = 1 {
      Notify = TermA {ObservedEvents =1112 {
        20010202T10030000:al/on
      }
    }
  }

The MGC responds to the MG1s Notify message.

Step 20
MGC to RGW1:

MEGACO/1 [216.33.33.61]:27000
Reply = 2002 {
  Context = 1 {
    Notify = TermA
  }
}

The MGC generates a Modify command towards the RGW2 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 21
MGC to RGW2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
  Context = 2 {
    Modify = TermB {
      Signals {cg/bt}
      Media {
        LocalControl {
          Mode = recvonly
        }
      },
      Modify = EphB {
        Media {
          LocalControl {
            Mode = recvonly
          }
        }
      }
    }
  }
}

The MG2 responds to this modify request.

Step 22
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1240 {
  Context = 2 {
    Modify = TermB, Modify = EphB
  }
}

The MGC generates transactions with two subtracts commands one for physical and other for ephemeral terminations. The MGC does the same for both the Contexts one at RGW1 and the other at RGW2.
Step 23:
MGC to MG1

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1241 {
   Context = 1 {
      Subtract = TermA {Audit{ }},
      Subtract = EphA {Audit{Statistics}}
   }
}
The MG subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 24
MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1241 {
   Context = 1 {
      Subtract = TermA
      Subtract = EphA {
         Statistics {
            rtp/ps=1234, ; packets sent
            nt/os=56789, ; octets sent
            rtp/pr=987, ; packets received
            nt/or=65432, ; octets received
            rtp/pl=10, ; % packets lost
            rtp/jit=30,
            rtp/delay=30 ; average latency
         }
      }
   }
}
The User B after going onhook, the RGW2 generates Notify command towards the MGC.

Step 25
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Transaction = 3001 {
   Context = 2 {
      Notify = TermB {ObservedEvents =1235 {
         20000202T10070000:al/on}}
   }
}
The MGC responds to the MG2 with the Notify response.

Step 26
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Reply = 3001 {
  Context = 2 {Notify = TermB}
}

The MGC generates subtract command towards RGW2 for removing TermB from valid context.

Step 26
MGC to MG2:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1242 {
    Context = 2 {
      Subtract = TermB {Audit{ }},
      Subtract = EphB {Audit{Statistics}}
    }
  }

The MG2 responds to the subtract commands generated by MGC.

Step 27
MG2 to MGC:
  MEGACO/1 [207.176.47.89]:26000
  Reply = 1242 {
    Context = 2 {
      Subtract = TermB
      Subtract = EphB {
        Statistics {
          rtp/ps=987, ; packets sent
          nt/os=65432, ; octets sent
          rtp/pr=1234, ; packets received
          nt/or=56789, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }

The MGC generates the message as shown in step 1 to both the RGW1 and RGW2, to enable the users to participate/initiate in further calls.
Case (b)

<table>
<thead>
<tr>
<th>USERA</th>
<th>RGW1</th>
<th>MGC</th>
<th>RGW2</th>
<th>USERB</th>
</tr>
</thead>
</table>

---

RTP MEDIA

<------------------------ UserB goes Onhook
<--------
Notify Onhook
---------->
Notify Resp
---------->

<-------- Modify TermA SD:BusyTone

<-------- Busy tone
-------->
Modify Resp

Subtract TermB
Subtract EphB
---------->
Subtract Resp TermB
Subtract Resp EphB Statistics
-------->

UserA goes Onhook
-------->
Notify OnHook
---------->
Notify Resp
---------->
Subtract TermA
Subtract EphA
---------->
Subtract Resp TermA
Subtract Resp EphA Statistics

The case (a) above illustrated the call flow between two Residential
Gateways controlled by a single MGC. In this section we will discuss the call scenario where the called party terminates the call. The assumptions for case (a) holds good for this call flow also. The call flow is same till step 18. The voice path is through and both the users are in conversation. We will consider the call flow when UserB goes on hook. As soon as the UserB goes on hook the RGW2 generates notify message to the MGC.

Step 19
MG2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Transaction = 3001 {
Context = 2 {
Notify = TermB {ObservedEvents = 1235 {
20010202T10030000:al/on}
}
}
}
The MGC responds to the MG’s notify message.

Step 20
MGC to MG2:

MEGACO/1 [216.33.33.61]:27000
Reply = 3001 {
Context = 2 {
Notify = TermB
}
}
The MGC generates a Modify command towards the RGW1 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 21
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
Context = 1 {
Modify = TermA {
Signals {cg/bt}
Media {
LocalControl {
Mode = recvonly}
}
},
Modify = EphA {
Media {
LocalControl {
Mode = recvonly}
}
}
}

The MG1 responds to this modify request.

Step 22
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1240 {
  Context = 1 {
    Modify= TermA, Modify = EphA
  }
}

The MGC generates transactions with two subtracts commands one for physical and other for ephemeral terminations. This command is generated towards the RGW2, since UserB has initiated the call tear down.

Step 23
MGC to MG2:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1242 {
  Context = 2 {
    Subtract = TermB {Audit {}},
    Subtract = EphB {Audit {Statistics}}
  }
}

The MG2 responds to the subtract commands generated by MGC.

Step 24
MG2 to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1242 {
  Context = 2 {
    Subtract = TermB
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}
The User A after hearing the busy tone goes onhook. The same activity is recognized as onhook event by the RGW1. The Notify command is generated towards the MGC.

Step 25
RGW1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Transaction = 2002 {
  Context = 1 {
    Notify = TermA {ObservedEvents =1112 {20010202T10030000:al/on}}
  }
}
The MGC responds to the MG1s Notify message.

Step 26
MGC to RGW1:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2002 {
  Context = 1 {
    Notify = TermA
  }
}
The MGC after receiving the Notify command with onhook event, generates subtract commands to remove the termination from context Identifier 1.

Step 27:
MGC to MG1

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1241 {
  Context = 1 {
    Subtract = TermA {Audit{ }},
    Subtract = EphA {Audit{Statistics}}
  }
}
The MG subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 28
MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1241 {
  Context = 1 {
    Subtract = TermA
    Subtract = EphA {
      Statistics {
      }
    }
  }
}
The MGC generates the message as shown in step 1 to both the RGW1 and RGW2, to enable the users to participate/initiate in further calls.

**Case (c)**

<table>
<thead>
<tr>
<th>USERA</th>
<th>RGW1</th>
<th>MGC</th>
<th>RGW2</th>
<th>USERB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modify to Check Offhook</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----------</td>
<td>Modify to check offhook</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-----------</td>
<td>Modify Resp Modify Resp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserA offhook</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Notify offhook</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dial Tone</td>
<td>---</td>
<td>Modify Resp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User Dials Digits</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Notify digits</td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al. [Page 22]
The two call flows explained in case (a) and (b) illustrate a successful call scenario. In this subsection we will look into the flow where UserA calls UserB and as the UserB is already in a call, UserA receives busy tone, thus illustrating an unsuccessful call scenario. The steps 1 through 8 remain the same.

When the MGC receives the digits dialed by UserA, it analyses the digits and find that the digits 2992 pertain to UserB, which is again controlled by the same MGC. In this example UserB is already in some call. Hence the call initiation from UserA becomes unsuccessful. MGC in this case issues a message, which instruct the MG, for busy tone application on the Termination TermA.

**Step 9**
MGC to MG1:  
```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
Context = - {
    Modify = TermA {
        Signals {cg/bt}
    }
}
}
```

MG responds to the Modify message from MGC. It applies busy tone on the termination TermA. MG also responds to the message generated from MGC.

**Step 10:**
MG1 to MGC:  
```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
Context = - {
    Modify = TermA
}
}
```
As soon as the UserA goes onhook MG detects the same and reports that event as parameter in the Notify command it generates towards the MGC.

Step 12:
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2002 {
  Context = - {
    Notify = TermA {ObservedEvents =1112 {
      20010202T10010000:al/on}
    }
  }
}
The MGC responds to the Notify generated by MG.

Step 11:
MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2002 {
  Context = - {
    Notify = TermA
  }
}
The MGC sends a Modify command towards that RGW1 for the Termination TermA as shown in step 1 that takes the termination TermA to idle state. In this state the RGW1 is again ready to detect events on the termination TermA.

2.2 Call between Residential Gateway and Trunking Gateway

In the earlier section we illustrated the call between two users UserA and UserB connected to residential gateways RGW1 and RGW2. In this section we illustrate the call flow between a Residential gateway user and a Trunking gateway. In this flow, the packages that are supported by the RGW are analog line supervision package, RTP package, generic package, DTMF detection package, and Call progress generator package, Network Package. The Trunking gateway is supports RTP package, generic package, and network package. We are not assuming any specific signaling to the other side of the Trunking gateway. This makes call flow simple and independent of the type of signaling towards the other side of the Trunking gateway.
USERA | RGW1 | MGC | TGW

<--------
Modify to
call offhook
-------->
Modify Resp

-------->
UserA offhook

-------->
Notify offhook
<--------
Notify Resp
<--------
Modify SG:dialtone
ED:al/on,dd/ce{Dmap1}
DM:Dmap1 = 2XXX

<--------
Dial Tone
<-------->
Modify Resp

<-------->
User Dials Digits

<-------->
Notify digits
<-------->
Notify Response
<-------->
Add TermA SD:ringbacktone
Add $, Local SDP Info -underspecified

<--------
RingBack Tone
<-------->
Modify Resp TermA
Add Resp Local SDP (Specified)

<-------->
Add Phy
Add $ Local(Underspecified)
Remote SDP (Specified)
<-------->
Add Resp Phy
The MGC generates modify command for to the residential gateway to check for offhook for Termination TermA. In this message the event offhook has an embedded signal descriptor and embedded event descriptor. The embedded signal descriptor is sent for application of dial tone immediately after the detection of the offhook event and the event descriptor then will be updated with the onhook and the digit map completion event. The Digit map is also defined in the digit map descriptor.

Step 1
MGC to RGW:
MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
Context = - {
Modify = TermA {
Media {
LocalControl {
Mode = Receiveonly}
},
DigitMap= Dmap1 {(2XXX)}
Events = 1111 {al/of Embed {signals {cg/dt}}, Events=1112 {al/on},

Madhubabu, et al.
The MG after receiving the MGC’s message responds to it.

Step 2
RGW to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1234 {
    Context = - {Modify = TermA}
  }

When the user A goes offhook RGW detects the offhook event and as it is listed in the event descriptor report the event detection using Notify command.

Step 3
RGW to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Transaction = 2000 {
    Context = - {
      Notify = TermA {ObservedEvents =1111 {
        20010202T10000000:al/of}}
    }
  }

the MGC responds with the Notify response.

Step 4
MGC to RGW:
  MEGACO/1 [216.33.33.61]: 27000
  Reply = 2000 {
    Context = - {Notify = TermA}
  }

The user gets Dial tone as part of Embedded descriptor processing. The digit map is active on the termination TermA. When the user dials the digits they are reported to MGC through Notify command.

Step 5
RGW to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Transaction = 2001 {
    Context = - {
      Notify = TermA {ObservedEvents =1112 {
        20010202T10010000:dd/ce {ds="2992", Meth=FM}}}
    }
  }
MGC after receiving the Notify command responds back with the Notify response.

Step 6
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2001 {
  Context = - {Notify = TermA}
}

The MGC generates the Add command for adding the physical termination TermA and to create an ephemeral termination EphA. The local SDP information for the ephemeral termination EphA is under specified to enable the RGW to allocate the necessary values by itself. The mode of the two terminations is set to receive only.

Step 7
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = $ {
    Add = TermA {
      Signals {cg/rt}
      Media {
        {LocalControl {
          Mode = ReceiveOnly,
        },
      }
      Add = $ {
        Media {
          {LocalControl {
            Mode = ReceiveOnly,
          },
          Local {
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 4
          }
        }
      }
    }
  }
}

MG after creating the context adds the physical termination TermA. In this case MG creates a context with ContextId 1. The ephemeral termination EphA is created and added to the context 1. The MG reserves resources for the local SDP information. In this case the IP address allocated is 209.110.59.33 and the port number used is 30000. The MG responds to the Add command with this information in
the response to MGC.

Step 8
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
  Context = 1 {
    Add = TermA,
    Add=EphA {
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        }; RTP profile for G.723 is 4
      }
    }
  }
  }
}

In this example as mentioned earlier we doesn’t assume any specific signaling in the other side of the Trunking gateway. This eliminates the complexity of reporting the address information and alerting the end user. The MGC then "Adds" a physical termination. The wildcard $ is used here to enable the Trunking gateway to choose one of its trunks on the other side of the Trunking gateway. The ephemeral termination is requested to create with the under specified SDP local information and the remote SDP information.

Step 9
MGC to TGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = $ {
    Add = Trunk1/$ {Media {
      LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
        c=IN IP4 $
        m=audio $ RTP/AVP 4
      },
      Remote {
        v=0
      }
    }
  }

The Trunking gateway then processes the command and does the necessary signaling towards the other side of the Trunking gateway. The seized trunk identifier is returned in the Add response for physical termination. In the response for the ephemeral termination, the TGW returns the local information of the ephemeral termination created. In this example the TGW creates a context with ContextID 2.

Step 10
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1236 {
  Context = 2 {
    Add = Trunk1/line1,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
        ; RTP profile for G723 is 4
      }
    }
  }
}

As mention earlier, How the MGC receives the indication about the status of the end user is out of scope of this call flow. But once the MGC receives the users status it forwards the remote SDP information for the RGW and change the mode for the termination both at RGW and TGW to SendRecv.

Step 11
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    }
  }
}
Signals { }
),
Modify = EphA {
    Media {
        LocalControl {
            Mode = sendrecv
        }
        Remote {
            v=0
c=IN IP4 207.176.47.90
m=audio 40000 RTP/AVP 4
        }
    } ; RTP profile for G723 is 4
}
}
}
}

The RGW after receiving the Modify command from MGC responds back with the response.

Step 12
RGW to MGC:
    MEGACO/1 [209.110.59.34]: 25000
    Reply = 1237 {
        Context = 1 {Modify = TermA, Modify = EphA}
    }

The MGC also generates commands to the TGW to change the mode of the Ephemeral termination to send receive. Thus enabling the virtual voice path between the two ephemeral terminations.

Step 13
MGC to TGW:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1238 {
        Context = 2 {
            Modify = Trunk1/line1 {
                Media {
                    LocalControl {
                        Mode = sendrecv
                    }
                }
            },
            Modify = EphB {
                Media {
                    LocalControl {
                        Mode = sendrecv
                    }
                }
            }
        }
    }

Madhubabu, et al.
The TGW after receiving the mode change information in Modify command responds to it.

Step 14
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
  Reply = 1238 {
    Context = 2 {Modify = Trunk1/line1, Modify = EphB}
  }

Now RTP media flow takes place. In the example the UserA goes onhook to terminate the call. The RGW after detecting the event reports the same to MGC in the Notify command.

Step 15:
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
  Transaction = 2002 {
    Context = 1 {
      Notify = TermA {ObservedEvents =1112 {
        20010202T10030000:al/on}
      }
    }
  }

The MGC responds to the MG1s Notify message.

Step 16
MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
  Reply = 2002 {
    Context = 1 {
      Notify = TermA
    }
  }

The MGC now generates subtract commands both to the RGW and the TGW. The mechanism of generating call progress tone towards the called user is not shown for simplicity. The two Subtract commands are clubbed in a single action.

Step 17
MGC to RGW
MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1239 {
    Context = 1 {
      Subtract = TermA {Audit{ }},
      Subtract = EphA {Audit(Statistics)}
    }
  }

The MG subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.
Step 18
RGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1239 {
  Context = 1 {
    Subtract = TermA
    Subtract = EphA {
      Statistics {
        rtp/ps=1234, ; packets sent
        nt/os=56789, ; octets sent
        rtp/pr=987, ; packets received
        nt/or=65432, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

The MGC generates similar command towards the TGW

Step 19
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
  Context = 2 {
    Subtract = Trunk1/line1{Audit{ }},
    Subtract = EphB {Audit{Statistics}}
  }
}

The TGW responds to the subtract commands generated by MGC.

Step 26
TGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1240 {
  Context = 2 {
    Subtract = Trunk1/line1
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}
2.3 Call between Trunking gateway and Residential Gateway

The earlier section illustrated the call between a residential user and a Trunking gateway. The call was initiated by the Residential gateway. In this call scenario we will illustrate the case where the call terminates on the Residential gateway when originated by the Trunking gateway. Even in this call flow we will assume that signaling beyond the Trunking gateway is taken care by an external entity, to avoid CAS/CCS specific details.

```
<table>
<thead>
<tr>
<th>TGW</th>
<th>MGC</th>
<th>RGW1</th>
<th>USERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Phy SD:ringbacktone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add $, Local SDP Info -underspecified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp Phy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp Local SDP (Specified)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add TermB SD:Ring ED:offhook</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add $ Local(Underspecified)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote SDP (Specified)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserB Phone Ringing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp TermB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Resp EphB Local Specified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserB goes Offhook</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify Offhook</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify Resp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Phy SendRecv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify EphA Remote(Specified) SendRecv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Resp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Phy SendRecv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify EphB SendRecv</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
The Media Gateway controller is intimated about the call initiation from the other side of the Trunking gateway. The MGC then indicates the TGW to seize a trunk using the Add command with $ (choose) wildcard. The MGC also requests for creation of an ephemeral termination using another Add command in the same action request. The Context Id specified by MGC is $ indicating that the TGW needs to create a context and "Add" these terminations in the new context.

Step 1
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = $ {
    Add = Trunk1/$ {Media {
      LocalControl {Mode = recvonly}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
    },
The TGW after receiving the Add request from MGC creates a context. In this example the ContextID allocated for the new context is 2. The physical termination chosen by the TGW is Trunk1/line1. The mode of the termination is set to Receiveonly. The TGW creates an ephemeral termination with the SDP information specified by MGC. The response contains all the specified values for the SDP information. To avoid complexity in this call flow no reserve group and reserve value properties are used. It should be assumed that they are all "OFF". In this example the TGW reserved 207.176.47.90 as the IP address for the RTP media stream and port number as 40000.

Step 2:

TGW to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1234 {
  Context = 2 {
    Add = Trunk1/line1,
    Add = EphA{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
    Media {
      Remote {
        v=0
        c=IN IP4 $ RTP/AVP 4
        m=audio $ RTP/AVP 4
      }
    }
  }
}
```

The MGC after receiving the response from TGW initiates the UserB alerting towards the RGW. This is done by MGC in the ADD command for the physical termination TermB. The MGC uses $ contextID. Indicating the creation of Context at MG. The Ephemeral termination is requested to create. The Remote SDP information is also passed as the parameters in the media descriptor. The signal descriptor in the Add command for physical termination enables RGW to apply the ring signal on the termination TermB.
Step 3
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000

Transaction = 1235 {
  Context = $ {
    Add = TermB {
      Signals { al/ri }  
      Events = 1111 { al/of Events = 1112 { al/on } }
      Media {
        LocalControl {
          Mode = ReceiveOnly,
        },
        Add = $ {
          Media {
            LocalControl {
              Mode = ReceiveOnly,
            },
            Local {
              v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
          }
          remote {
            v=0
c=IN IP4 207.176.47.90
            m=audio 40000 RTP/AVP 4
          }
        }
      }
    }
  }
}

The RGW after receiving the transaction request from MGC first creates a context. In this example the contextID allocated is 1. The RGW "adds" the termination TermB to context 1 and as part of the signal descriptor processing applies the "ring" signal on the termination. The Events descriptor lists the offhook event of analog line supervision package. The requestId specified in this example is 1111. The RGW also creates the ephemeral termination EphB. Whose remote SDP information is specified by MGC and the local SDP information is allocated and reserved by RGW. In this example the RGW allocates an IP address of 209.110.59.33 and port number 30000. The mode of the ephemeral termination is set to receive only.

Step 4
RGW to MGC:
The UserB goes Offhook after hearing the "ring" signal. The offhook event is reported to the MGC as observed event descriptor in Notify command.

Step 5
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2000 {
  Context = 1 {
    Notify = TermB {ObservedEvents =1111 {
      20010202T10001000:al/of}}
  }
}
The MGC responds with the Notify response.

Step 6
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = TermB}
}
The MGC now generates commands to both RGW and TGW. These commands are generated to modify the mode of the physical and ephemeral terminations towards both RGW and TGW.

Step 7:
MGC to RGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = 1 {
    Modify = TermB {
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        } ; RTP profile for G.723 is 4
      }
    }
  }
}

The UserB goes Offhook after hearing the "ring" signal. The offhook event is reported to the MGC as observed event descriptor in Notify command.

Step 5
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2000 {
  Context = 1 {
    Notify = TermB {ObservedEvents =1111 {
      20010202T10001000:al/of}}
  }
}
The MGC responds with the Notify response.

Step 6
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = TermB}
}
The MGC now generates commands to both RGW and TGW. These commands are generated to modify the mode of the physical and ephemeral terminations towards both RGW and TGW.

Step 7:
MGC to RGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = 1 {
    Modify = TermB {
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        } ; RTP profile for G.723 is 4
      }
    }
  }
}
The RGW responds with the Transaction response for the commands sent for both physical and ephemeral termination.

Step 8
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply= 1236 {
  Context = 1 {
    Modify = TermB {}
    Modify = EphB {
  }
}
The MGC also generates similar command towards the Trunking gateway.
The MGC in the Modify for the ephemeral termination also specifies the remote SDP information, and this enables the TGW to change the mode of the terminations to send and receive.

Step 9
MGC to TGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
  Context = 2 {
    Modify = Trunk1/line1 {
      Media {
        LocalControl {
          Mode = SendRecv
        }
      }
      Modify = EphA {
        Media {
          LocalControl {
            Mode = SendRecv
          }
          Remote {
            v=0
            c=IN IP4 207.176.47.90
            m=audio 40000 RTP/AVP 4
          }
        }
      }
    }
  }
}
The Trunking gateway responds to the MGC message and changes the mode for both the terminations to Send and receive.

Step 10
TGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1237 {
  Context = 2 {
    Modify = Trunk1/line1,
    Modify = EphA {
      
    }
  }
}

Once both the terminations mode is change to send receive the two users can start the conversation. The media stream is established. The user to the other side of the Trunking gateway terminates the Call. The MGC is updated with this information by the external entity. The MGC generates a busy signal towards UserB connected to the RGW.

Step 11
MGC to RGW:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1238 {
  Context = 1 {
    Modify = TermB {
      Signals { cg/bt }
    }
  }
}

The RGW as part of signal descriptor processing plays the busy tone towards the UserB termination TermB. The response is generated after processing the command from MGC.

Step 12
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1238 {
  Context = 1 {
    Modify = TermB {
    }
  }
}

Madhubabu, et al.
The MGC terminates the call by generating Subtract command for both the terminations in TGW and RGW.

Step 13  
MGC to TGW:  
MEGACO/1 [216.33.33.61]: 27000  
Transaction = 1239  
  Context = 2  
    Subtract = Trunk1/line1{Audit{ }},  
    Subtract = EphA {Audit{Statistics}}  
}  
The TGW responds with the Statistics in the Subtract response for the ephemeral termination. The context is created as an side effect of deleting both the terminations in the context.

Step 14:  
TGW to MGC:  
MEGACO/1 [207.176.47.89]:26000  
Reply = 1239  
  Context = 2  
    Subtract = Trunk1/line1  
    Subtract = EphA {  
      Statistics {  
        rtp/ps=987, ; packets sent  
        nt/os=65432, ; octets sent  
        rtp/pr=1234, ; packets received  
        nt/or=56789, ; octets received  
        rtp/pl=10, ; % packets lost  
        rtp/jit=30,  
        rtp/delay=30 ; average latency  
      }  
    }  
}  
The MGC generates similar command towards the RGW to delete the physical and ephemeral terminations.

Step 15  
MGC to RGW:  
MEGACO/1 [216.33.33.61]: 27000  
Transaction = 1240  
  Context = 1  
    Subtract = TermB {Audit{ }},  
    Subtract = EphB {Audit{Statistics}}  
}  
The RGW responds with the statistics for the ephemeral termination in the Subtract response.
Step 16
RGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1240 {
  Context = 1 {
    Subtract = TermB
    Subtract = EphB {
      Statistics {
        rtp/ps=1234, ; packets sent
        nt/os=56789, ; octets sent
        rtp/pr=987, ; packets received
        nt/or=65432, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}
The Call is terminated with the response received from RGW. The MGC generates the initial Modify command to check for offhook on the termination TermB. This takes the termination TermB to idle state and the UserB is ready to ready to generate and receive further calls.

2.4 Call between two Trunking gateways
The earlier three sections illustrated the calls between two residential gateways, and between residential gateway and Trunking gateway. This two Trunking gateway call flow scenario illustrates call between two users connected to the other side of the Trunking gateway. Even in this scenario to avoid unnecessary complexity we still assume that some external entity conveys the necessary signaling information to the MGC to enable it to control the two Trunking gateways. The CAS/CCS signaling details towards the other side of the each Trunking gateway is avoided for simplicity. This boils down to the message generation from MGC for enabling media flow. No signaling is done on either the physical or ephemeral termination.

<table>
<thead>
<tr>
<th>TGW1</th>
<th>MGC</th>
<th>RGW1</th>
<th>TGW2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--------</td>
<td>Add Phy SD:ringbacktone</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add $, Local SDP Info -underspecified</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al. [Page 42]
When the MGC receives the call establishment from one side of the Trunking gateway it generates two ADD commands in a single action. The action request includes creation of Context, choosing of physical termination and adding the same to the context created and creating of ephemeral termination and adding this to the newly created context. The local SDP parameters are under specified. The TGW1 in its response need to specify these under specified parameters.

Step 1
MGC to TGW1:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = $ {
    Add = Trunk1/$ {Media {
...
The TGW after receiving the Add request from MGC creates a context. In this example the ContextID allocated for the new context is 1. The physical termination chosen by the TGW is Trunk1/line1. The mode of the termination is set to Receiveonly. The TGW creates an ephemeral termination with the SDP information specified by MGC. The response contains all the specified values for the SDP information. To avoid complexity in this call flow no reserve group and reserve value properties are used. It should be assumed that they are all "OFF". In this example the TGW reserved 209.110.59.33 as the IP address for the RTP media stream and port number as 40000.

Step 2:
TGW1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
  Context = 1 {
    Add = Trunk1/line1,
    Add = EphA{
      Media {
        Local {
          v=0
          c=IN IP4 $ 209.110.59.33
          m=audio $ RTP/AVP 4
        }
      }
    }
  }
}
```

In this example as mentioned earlier it doesn’t assume any specific signaling in the other side of the Trunking gateway. Thus this eliminates the complexity of reporting the address information and alerting the end user. The MGC hence "Adds" a physical termination. The wildcard $ is used here to enable the Trunking gateway to choose...
one of its trunks on the other side of the Trunking gateway. The ephemeral termination is requested to create with the under specified SDP local information and the remote SDP information.

Step 3
MGC to TGW2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = $ {
    Add = Trunk2/$ {Media {
      LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
        c=IN IP4 $
        m=audio $ RTP/AVP 4
      },
      Remote {
        v=0
        c=IN IP4 209.110.59.33
        m=audio 40000 RTP/AVP 4
      } ; RTP profile for G.723 is 4
    }
  }
}

The Trunking gateway2 then process the command and does the necessary signaling towards the other side of the Trunking gateway. The seized trunk identifier is returned in the Add response for physical termination. In the response for the ephemeral termination, the TGW returns the local information of the ephemeral termination created. In this example the TGW creates a context with ContextID 2.

Step 4
TGW2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1235 {
  Context = 2 {
    Add = Trunk2/line1,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
  }
}
Once these MGC receives the answer of the call by the end user (through an entity outside the scope of the flow), changes the mode of the termination towards the both Trunking gateways. The MGC also indicates the remote SDP information for the TGW1. Thus enabling the exchange of media parameter information to both the Trunking gateways.

Step 5
MGC to TGW1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = 1 {
    Modify = Trunk1/line1 {
      Media {
        LocalControl {
          Mode = SendRecv
        }
      }
    }
    Modify = EphA {
      Media {
        LocalControl {
          Mode = SendRecv
        }
      }
    }
  }
}

The TGW1 after receiving the command from MGC does the necessary processing to update and processing the remote SDP information. The mode of the terminations is modified to send receive. The response of the transaction is sent back to MGC.

Step 6
TGW1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Modify = Trunk1/line1,
    Modify = EphA {
    }
  }
}
Similar command is generated from the MGC towards the second Trunking gateway.

Step 7
MGC to TGW1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
  Context = 2 {
    Modify = Trunk2/line1 {
      Media {
        LocalControl {
          Mode = sendrecv}
        }
    },
    Modify = EphB {
      Media {
        LocalControl {
          Mode = sendrecv}
        }
    }
  }
  }
}

The TGW2 after receiving the mode change information in Modify command responds to it.

Step 8
TGW1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1237 {
  Context = 2 {Modify = Trunk1/line1, Modify = EphB}
}

The RTP media flow takes place once the modes of the terminations are changed to send receive. The MGC is intimated about the call clearing from an external entity. The MGC then generates subtract commands to both the TGWs.

Step 9
MGC to TGW2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
  Context = 1 {
    Subtract = Trunk1/line1{Audit{ }},
    Subtract = EphA {Audit(Statistics)}
  }
}

The TGW responds with the Statistics in the Subtract response for the ephemeral termination. The context is created as an side effect of deleting both the terminations in the context.
Step 10:
TGW2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Reply = 1238 {
  Context = 1 {
    Subtract = Trunk1/line1
    Subtract = EphA {
      Statistics {
        rtp/pl=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

A similar command is generated from the MGC towards the second TGW.

Step 11
MGC to TGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239 {
  Context = 2 {
    Subtract = Trunk2/line1{Audit{ }},
    Subtract = EphB {Audit{Statistics}}
  }
}

The TGW responds to the subtract commands generated by MGC.

Step 12
TGW to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1239 {
  Context = 2 {
    Subtract = Trunk2/line1
    Subtract = EphB {
      Statistics {
        rtp/pl=1234, ; packets sent
        nt/os=56789, ; octets sent
        rtp/pr=987, ; packets received
        nt/or=65432, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

The two Trunking gateways are now ready to receive further commands
The calls flow scenarios from 2.2 to 2.4 illustrated the Trunking gateway without considering the signaling performed at the other side of the Trunking gateway. In this section we will illustrate the call flow scenario similar to the one in section 2.2. The emphasis here will be both on the Megaco messages exchanged between MG and MGC and also the signaling that towards the other side of the Trunking gateway.

Here in this example the CCS SS7 is assumed. The packages that supported are Call progress tone generation package, analog line supervision package, generic package, RTP package and Network package for the RGW and Network and RTP package for the TGW.
The MGC generates modify command for to the residential gateway to check for offhook for Termination TermA. In this message the event offhook has an embedded signal descriptor and embedded event descriptor. The embedded signal descriptor is sent for application of dial tone immediately after the detection of the offhook event and the event descriptor then will be updated with the onhook and the digit map completion event. The Digit map is also defined in the digit map descriptor.

Step 1
MGC to RGW:
```
MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = ReceiveOnly}
        DigitMap = Dmap1{(2XXX)}
      }
      Events = 1111 {al/of Embed {signals {cg/dt}, Events=1112 { al/on},
        {dd/ce {Dmap1}}}
    }
  }
}
```

The MG after receiving the MGC’s message responds to it.

Step 2
RGW to MGC:
```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
  Context = - (Modify = TermA)
}
```

When the user A goes offhook RGW detects the offhook event and as it is listed in the event descriptor reports the event detection using Notify command.

Step 3
RGW to MGC:
```
MEGACO/1 [209.110.59.34]:25000
Transaction = 2000 {
  Context = - {
    Notify = TermA {Observed Events =1111 {
      20010202T10000000:al/of}}
  }
}
```

the MGC responds with the Notify response.

Step 4

Madhubabu, et al.
The digit map is active on the termination TermA. When the user dials the digits they are reported to MGC through Notify command.

Step 5
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
  Context = - {
    Notify = TermA {ObservedEvents = 1112 {
      20010202T10010000:dd/ce{ds="2992",Meth=FM}}
    }
  }
}

MGC after receiving the Notify command responds back with the Notify response.

Step 6
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2001 {
  Context = - {Notify = TermA}
}

The MGC generates the Add command for adding the physical termination TermA and to create an ephemeral termination EphA. The local SDP information for the ephemeral termination EphA is under specified to enable the RGW to allocate the necessary values by itself. The mode of the two terminations is set to receive only.

Step 7
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = $ {
    Add = TermA {
      Signals {cg/rt}
      Media {
        LocalControl {
          Mode = ReceiveOnly,
        },
      }
      Add = $ {
        Media {
        }
      }
    }
  }
}
MG after creating the context adds the physical termination TermA. In this case MG creates a context with ContextId 1. The ephemeral termination EphA is created and added to the context 1. The MG reserves resources for the local SDP information. In this case the IP address allocated is 209.110.59.33 and the port number used is 30000. The MG responds to the Add command with this information in the response to MGC.

Step 8

RGW to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        }
      }
    }
  }
}
```

In this example as mentioned earlier the CCS SS7 signaling is assumed towards the other side of the Trunking gateway. The IAM message is generated towards the SS7 switch, with the necessary information about the called party and calling party. The SS7 switch responds with ACM indicating that address complete message towards the Trunking gateway. When the ANM "answer" message is received the MGC adds a physical termination. The wildcard $ is used here to enable the Trunking gateway to choose one of its trunks on the other side of the Trunking gateway. The ephemeral termination is requested to create with the under
specified SDP local information and the remote SDP information.

Step 9
MGC to TGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = $ {
    Add = Trunk1/$ {Media {
      LocalControl {Mode = SendRecv}},
    },
    Add  = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
      },
      Remote {
        v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
      } ; RTP profile for G.723 is 4
    }
  }
}

The Trunking gateway then processes the command and does the necessary signaling towards the other side of the Trunking gateway. The seized trunk identifier is returned in the Add response for physical termination. In the response for the ephemeral termination, the TGW returns the local information of the ephemeral termination created. In this example the TGW creates a context with ContextID 2.

Step 10
TGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1236 {
  Context = 2 {
    Add = Trunk1/line1,
    Add = EphB{
      Media {
        Local {
          v=0
c=IN IP4 207.176.47.90
m=audio 40000 RTP/AVP 4
      }
    }
  }
}

Madhubabu, et al.                                           [Page 54]
As mentioned earlier, MGC, after receiving the ANM message to indicate the status of the end user, it forwards the remote SDP information for the RGW and changes the mode for the termination both at RGW and TGW to SendRecv.

Step 11
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
   Context = 1 {
      Modify = TermA {
         Media {
            LocalControl {
               Mode = sendrecv
            }
         }
         Signals { }
      },
      Modify = EphA {
         Media {
            LocalControl {
               Mode = sendrecv
            }
         }
      }
   }
}

The RGW, after receiving the Modify command from MGC, responds back with the response.

Step 12
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1237 {
   Context = 1 {Modify = TermA, Modify = EphA}
}

The MGC also generates commands to the TGW to change the mode of the Ephemeral termination to send receive. Thus enabling the virtual voice path between the two ephemeral terminations.
Step 13
MGC to TGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
  Context = 2 {
    Modify = Trunk1/line1 {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    },
    Modify = EphB {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    }
  }
}
```

The TGW after receiving the mode change information in Modify command responds to it.

Step 14
TGW to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1238 {
  Context = 2 (Modify = Trunk1/line1, Modify = EphB)
}
```

Now RTP media flow takes place. In the example the UserA goes onhook to terminate the call. The RGW after detecting the event reports the same to MGC in the Notify command.

Step 15:
RGW to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2002 {
  Context = 1 {
    Notify = TermA {ObservedEvents = 1112 {
      20010202T10030000:al/on}
    }
  }
}
```

The MGC responds to the RGWs Notify message.

Step 16
MGC to RGW:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 2002 {
  Context = 1 {
    Notify = TermA
  }
}
```
The MGC now generates subtract commands both to the RGW and the TGW. The mechanism of generating call progress tone towards the called user is not shown for simplicity. The two Subtract commands are clubbed in a single action.

Step 17
MGC to RGW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1239 {
    Context = 1 {
      Subtract = TermA {Audit {}},
      Subtract = EphA {Audit {Statistics}}
    }
  }

The MG subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 18
RGW to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Reply = 1239 {
    Context = 1 {
      Subtract = TermA
      Subtract = EphA {
        Statistics {
          rtp/ps=1234, ; packets sent
          nt/os=56789, ; octets sent
          rtp/pr=987, ; packets received
          nt/or=65432, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }

The MGC generates a REL "release" message towards the SS7 switch. After receiving the RLC "release complete" message command is generated towards the TGW to subtract the two terminations from context.

Step 19
MGC to TGW:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1240 {
    Context = 2 {
      Subtract = Trunk1/line1{Audit{ }}
    }
  }

Madhubabu, et al.                                           [Page 57]
The TGW responds to the subtract commands generated by MGC.

Step 26
TGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1240 {
  Context = 2 {
    Subtract = Trunk1/line1
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

2.6 Call between SS7 trunk in TGW and residential gateway

In the earlier call scenario 2.5, we illustrated the call flow between MGC and MG considering the CCS SS7 signaling towards one side of the Trunking gateway. The residential Gateway user in that scenario initiated the call. In this scenario we illustrate similar situation with call initiated by a user connected to the PSTN network that originates SS7 signaling. The packages that are supported include Call progress tone generation package, analog line supervision package, generic package, RTP package and Network package for the RGW and Network and RTP package for the TGW.

<table>
<thead>
<tr>
<th>SS7 Switch</th>
<th>TGW</th>
<th>MGC</th>
<th>RGW1</th>
<th>USERB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>USERB</td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al.
Add Phy
Add $, Local SDP Info -underspecified
-------------------->
Add Resp Phy
Add Resp Local SDP (Specified)
-------------------->
Add Resp TermB
Add Resp EphB Local Specified
<------------------->
UserB goes Offhook

<----------------------->
ANM

-------------------
Notify Offhook
-------------------->
Notify Resp
<--------
Modify Phy SendRecv
Modify EphA Remote(Specified) SendRecv
-------------------->
Modify Resp
-------------------->
Modify Phy SendRecv
Modify EphB SendRecv
<--------
Modify Resp
/R--------------------------\
\----------------------------/

RTP MEDIA

------------------------>
REL

---------------
Modify TermB SD:BusyTone
--------------->
Busy tone to UserB
<--------
Modify Resp
<--------
Subtract Phy
Subtract EphA
--------------->
Subtract Resp Phy
Subtract Resp EphA Statistics
<---------------------->
We assume that the TGW and SG are in a single box, such that both signaling and Media adaptation is done by the same entity. The Media Gateway controller is intimated about the call initiation from the other side of the Trunking gateway, when the SS7 message IAM is received. The MGC after receiving the IAM messages responds with the ACM message indicating that the called party address is sufficient to identify the end user. The ACM message is sent to the SS7 switch through the signaling gateway. The circuit to be used for media information is indicated by the SS7 message received by the MGC.

The MGC generates the Modify command to the Residential gateway for application of "ring" signal to the user. The event descriptor in the Modify command lists the offhook event.

Step 1
MGC to RGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
Context = - {
Modify = TermA {
   Media {
      LocalControl {
         Mode = ReceiveOnly
      },

   Events = 1111 {al/of}
   Signals = {al/ri}; For application of ring signal
   }
  }
},
```

The Residential gateway after receiving the Modify command responds with the Modify response.

Step 2
RGW to MGC:
The users after receiving the "ring" signal assume to go offhook. The same event is reported to the MGC in the notify command.

Step 3

RGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Transaction = 2000 {
    Context = - {
        Notify = TermA {Observed Events =1111 {
            20010202T10000000:al/of}}
    }
}

the MGC responds the Notify command with the Notify response.

Step 4

MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
    Context = - {Notify = TermA}
}

The MGC after receiving the offhook event responds with the SS7 message to be generated towards the SS7 switch. The ANM message generated from MGC indicates the answering state of the user. The MGC meanwhile generate ADD commands to both the Trunking gateway and Residential gateway. The message towards the Trunking gateway includes the addition of physical circuit that was seized for media transfer. The request for creating ephemeral termination is also sent in the same request. The Local SDP information is under specified allowing the Trunking gateway to choose the necessary SDP parameters.

Step 5

MGC to TGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
    Context = $ {
        Add = Trunk1/line1 {Media {
            LocalControl {Mode = SendRecv}},
        },
        Add = $ {Media {
            LocalControl {
                Mode = Receiveonly,
            },
            Local {
                v=0
            }
        },
    }
The Trunking gateway after responds with a contextID in this example 1. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 6
TGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1235 {
     Context = 1 {
        Add = Trunk1/line1,
        Add = EphB {
            Media {
                Local {
                    v=0
                    c=IN IP4 209.110.59.33
                    m=audio 30000 RTP/AVP 4
                }
                ; RTP profile for G723 is 4
            }
        }
    }
}

The MGC after receiving the response from the Trunking gateway uses the SDP information in the response sent from TG to the Residential gateway. The commands for adding the physical termination TermA and ephemeral termination to be created are sent in the same transaction. The MGC requests creation of context and to add the two terminations in the same.

Step 7
MGC to RGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
     Context = $ {
        Add = TermA  {Media {
            LocalControl {Mode = SendRecv}},
        },
        Add = $ {Media {

The Residential gateway creates a context with ContextId 2. It adds the physical termination TermA in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 8
RGW to MGC:
 MEGACO/1 [209.110.59.34]: 25000
 Reply = 1237 {
   Context = 2 {
     Add = TermA,
     Add = EphA{
       Media {
         Local {
           v=0
           c=IN IP4 $ 209.110.59.34
           m=audio 30000 RTP/AVP 4
         }
       }
     }
   }

   ; RTP profile for G723 is 4
 }

The MGC after receiving the response with the local SDP information conveys the same to the Trunking gateway as remote SDP information in the Modify command.

Step 9
MGC to TGW
 MEGACO/1 [216.33.33.61]: 27000
 Transaction = 1238 {
   Context = 1 {
     Modify = EphA
The Trunking gateway responds to the Modify command.

Step 10
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1238 {
   Context = 1 {
      Modify = EphA
   }
}
The media properties are exchanged between the two media gateways and the RTP media can be transferred between these two. The call be will be disconnected with one of the two users going onhook. In this example we consider the user connected to the SS7 domain initiates the termination of the Call. The SS7 switch generates the REL "release" message towards the MGC. The MGC after receiving the REL message initiates the call teardown. The MGC generates a busy tones towards the user connected to the Residential gateway to indicate the disconnection of the call. The modify command is sent with event descriptor that lists the onhook event.

Step 11
MGC to RGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239 {
   Context = 2 {
      Modify = TermA {
         Signals = cg/bt,
         Events = 1236 {
            al/on
         },
         Media {
            LocalControl { recvonly }
         }
      }
   }
}
The Residential gateway responds with the Modify command response.
Step 12
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1239 {
  Context = 2 {
    Modify = TermA,
  }
}
The media connection tears down will be complete by responding the REL message by the RLC. The MGC subtracts the terminations added in the two Media gateways for this call. The Subtract command is sent to the Trunking gateway. The audit descriptor lists the statistics to enable statistics reporting to MGC from the Trunking gateway.

Step 13
MGC to TGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
  Context = 1 {
    Subtract = Trunk1/line1{Audit{}},
    Subtract = EphB {Audit{Statistics}}
  }
}
The TGW responds to the subtract commands generated by MGC.

Step 14
TGW to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1240 {
  Context = 1 {
    Subtract = Trunk1/line1
    Subtract = EphB {
      Statistics {
        rtp/pr=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

Step 15
RGW to MGC:

MEGACO/1 [209.110.59.34]:25000
Transaction = 2001 {
  Context = 2 {
    Notify = TermA {Observed Events =1236 {
The MGC responds the Notify command with the Notify response.

Step 16
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2001 {
  Context = 2 {Notify = TermA}
}

The MGC after receiving the Notify for onhook from UserA generates
transaction with two Subtract command for subtracting both the physical
and ephemeral terminations from the Residential gateway.

Step 15
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1241 {
  Context = 2 {
    Subtract = TermA{Audit{ }},
    Subtract = EphA {Audit{Statistics}}
  }
}

The RGW responds to the subtract commands generated by MGC.

Step 16
TGW to MGC:

MEGACO/1 [209.110.59.34]: 26000
Reply = 1241 {
  Context = 2 {
    Subtract = TermA
    Subtract = EphA {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

Madhubabu, et al.                                           [Page 66]
2.7 Call between SS7 trunk in TGW and R2 trunk in TGW

This section of the call flow illustrates the call between two Trunking gateways, one that does R2 signaling towards the PSTN and the second TGW CCS SS7 towards the PSTN. In this example the user in the CCS SS7 signaling network originates the call. The destination user is connected to the PSTN network with R2 signaling. The R2 package draft is taken as the input for illustrating the events and signals used for communication between MG and MGC. The assumption of the R2 package, that the compelled signaling is part of the MG to generate signals or detect events holds true for this call scenario also.
When the MGC through the signaling gateway receives the IAM it generates a Modify message towards the Trunking gateway that supports R2 package. The signal descriptor has the seize signal and the events descriptor has the event seizure. The seizure event has an embedded signal "address". The Trunking gateway is intended to seize a circuit and apply the address signals after the seizure acknowledgement is received from the other R2 exchange. The calling party information can also be provided in the address signal. The embedded event of the seizure acknowledgement event is the answer. This event has to be reported when the digits reach the other R2 exchange and there is answer from the terminating R2 exchange. The message from MGC to R2TGW is as follows.

Step 1
MGC to R2TGW
MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1234 {
        Context = - {
            Modify = Trunk1/line1 {
                Signals = {R2/sz}
                Events = 1111 {R2/sa Embed {signals {R2/address {si = 2992, di = 2989} },
                    Events = 1112 {R2/clear, R2/answer}
                }
            }
        }
    }

The R2TGW after receiving the Modify command does the command processing and responds with Modify command response.

Step 2
R2TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
    Reply = 1234 {
        Context = - {
            Modify = Trunk1/line1
        }
    }

The R2 TGW applies seize signal on the specified circuit group and after receiving the acknowledgement from the other R2 exchange updates the embedded events descriptor as the events descriptor. The R2TGW also generates a Notify command towards the MGC to indicate that the seize acknowledgement has been received.

Step 3
R2GW to MGC:
the MGC responds the Notify command with the Notify response.

Step 4
MGC to R2GW:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = Trunk1/line1}
}

The MGC now generates the address complete message to the SS7 switch
that has generated the IAM message. The R2TGW after receiving the
answer event from the other R2 exchange generates message to notify
this event.

Step 5
R2GW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
  Context = - {
    Notify = Trunk1/line1 {Observed Events =1112 {
                      20010202T10000000:R2/ans}}
  }
}

The MGC responds the Notify command with the Notify response.

Step 6
MGC to R2GW:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2001 {
  Context = - {Notify = Trunk1/line1}
}

MGC generates the ANM message towards the SS7 switch through the
signaling gateway. It also generates commands to add terminations into
specific contexts. It generates a single transaction with two Add
commands towards the TGW that supports R2 terminations. One command
is to add the physical circuit group Trunk1/line and the other for the
ephemeral termination. The SDP information is unspecified that enables
the TGW to choose the under specified parameters.
MEGACO/1 [216.133.33.61]: 27000
Transaction = 1235{
  Context = $ {
    Add = Trunk1/line1 (Media {
      LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
        c=IN IP4 $
        m=audio $ RTP/AVP 4
      }
    })
  }
}

The R2 Trunking gateway after receiving the Add command responds with a contextID in this example 1. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 8
R2TGW to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1235{
    Context = 1 {
      Add = Trunk1/line1,
      Add = EPHA {
        Media {
          Local {
            v=0
            c=IN IP4 209.110.59.33
            m=audio 30000 RTP/AVP 4
          }
        }; RTP profile for G723 is 4
      }
    }
  }
}

The MGC after receiving the response generates similar transaction with two ADD commands to the other TGW. The local SDP information specified in by the R2TGW is used to as remote SDP information for the other TGW. The MGC conveys this information in the Add command.
Step 9
MGC to TGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236{
  Context = $ {
    Add = Trunk2/$ {Media {
      LocalControl {Mode = SendRecv},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
    }
  }
  }
Remote{
  v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
}
  } ; RTP profile for G723 is 4
}
}

The Trunking gateway creates a context with ContextId 2. It adds the physical termination Trunk2/line1 in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 10
RGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1236{
  Context = 2 {
    Add = Trunk2/line1,
    Add = EphB{
      Media {
        Local {
          v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
      }
    }
  } ; RTP profile for G723 is 4
}
The MGC after receiving the response generates a modify command to the R2TGW to inform the local SDP information of TGW as the remote SDP for the R2TGW.

Step 11
MGC to TGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237{
  Context = 1 {
    Modify = EphA
    {Media {
      LocalControl {
        Mode = SendRecv,
      },
      Remote{
        v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
      }
    }
  }
}
}

The R2 Trunking gateway responds to the Modify command.

Step 12
TGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1237{
  Context = 1 {
    Modify = EphA
  }
}

The RTP media flow is established and the two users connected to the SS7 trunk and R2 trunk starts the conversation. After conversation any of the user can disconnect the call, in this example the user connected to the SS7 domain releases the call. The SS7 switch generates a REL message towards the MGC. The Signaling gateway forwards the same request towards the MGC. The MGC initiates the tearing down of the call. This is done initially by generating a Modify command towards the R2 TGW to generate clear forward signal towards the R2 exchange. In the message the MGC sends a signal descriptor with clear signals and events descriptor with the clear in the event. The event in the events descriptor is for detecting the clear back signal generated from the R2 exchange.

Step 13
MGC to TGW:

MEGACO/1 [216.33.33.61]: 27000
The R2 TGW after receiving the commands does the signals and events descriptor processing and responds to the MGC with the Modify command response.

Step 14
R2TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
Reply = 1238 {
  Context = 1 {
    Modify = Trunk1/line1
  }
}

Mean while the MGC generates the subtract message towards the originating TGW to remove the terminations from the newly created context.

Step 15
MGC to TGW:
  MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239
  Context = 2 {
    Subtract = Trunk2/line1{Audit{ }},
    Subtract = EphB {Audit{Statistics}}
  }
}
The RGW responds to the subtract commands generated by MGC.

Step 16
TGW to MGC:
  MEGACO/1 [207.176.47.89]:26000
Reply = 1239
  Context = 2 {
    Subtract = Trunk2/line1
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
    }
  }
}
The MGC after receiving the response from the TGW generates the Release complete message RLC towards the SS7 switch through the Signaling Gateway.

The R2 TGW after receiving the clear back signal from the other R2 exchange detects it and generates a Notify command towards the MGC.

Step 17
R2GW to MGC:

```
MEGACO/1 [209.110.59.34]:25000
Transaction = 2002 {
    Context = - {
        Notify = TermA {Observed Events = 1113 {
            20030202T100000000:R2/clear}}
    }
}
```

the MGC responds the Notify command with the Notify response.

Step 18
MGC to R2GW:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 2002 {
    Context = - {Notify = TermA}
}
```

MGC after receiving the notify command generates subtract command to remove both the physical termination and ephemeral termination.

Step 19
MGC to R2TGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240{
    Context = 1 {
        Subtract = Trunk1/line1{Audit{ }},
        Subtract = EphA {Audit(Statistics)}
    }
}
```

The R2TGW responds to the subtract commands generated by MGC.

Step 20
R2TGW to MGC:

```
MEGACO/1 [209.110.59.34]:25000
Reply = 1240{
    Context = 1 {
        Subtract = Trunk1/line1
        Subtract = EphA {
            Statistics {
```

Madhubabu, et al.                                            [Page 75]
2.8 Call between R2 trunk in TGW and SS7 trunk in TGW

This section of the call flow illustrates the call between two Trunking gateways, one that does R2 signaling towards one end and the second TGW CCS SS7. In this example the user in the R2 signaling network originates the call. The destination user is connected to the PSTN network with CCS SS7 signaling. The R2 package draft is considered for illustrating the events and signals used for communication between MG and MGC. The assumption of the R2 package, that the compelled signaling is part of the MG to generate signals or detect events holds true for this call scenario also.

<table>
<thead>
<tr>
<th>R2Exchange</th>
<th>TGW1</th>
<th>MGC</th>
<th>TGW2</th>
<th>SS7 Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Modify ED:sz{SD:sa ED:addr, cl}</td>
<td>&lt;------------</td>
<td>Modify Resp</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>&lt;-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Seize</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>------------</td>
<td>&lt;-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>SeizAck</td>
<td>---------------</td>
<td>Notify OE:sa</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>&lt;-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Digit 1</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>------------</td>
<td>&lt;-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Send Next Digit</td>
<td>&lt;-----------</td>
<td>End of Singaling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Message</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify OE:addr</td>
<td>IAM</td>
<td>Notify address</td>
<td>IAM</td>
<td>Notify address</td>
</tr>
<tr>
<td>Notify Resp</td>
<td>ACM</td>
<td>Notify response</td>
<td>ACM</td>
<td>Notify response</td>
</tr>
<tr>
<td>Modify Phy</td>
<td>IAM</td>
<td>Modify physical</td>
<td>IAM</td>
<td>Modify physical</td>
</tr>
<tr>
<td>Answer</td>
<td>ACM</td>
<td>Answer call</td>
<td>ACM</td>
<td>Answer call</td>
</tr>
<tr>
<td>Modify Resp</td>
<td>Modify</td>
<td>Modify response</td>
<td>Modify</td>
<td>Modify response</td>
</tr>
<tr>
<td>Add Phy</td>
<td>Add Phy</td>
<td>Add physical</td>
<td>Add Phy</td>
<td>Add physical</td>
</tr>
<tr>
<td>Add Eph Local Unspecified</td>
<td>Add Eph Local Unspecified</td>
<td>Add ephemeral</td>
<td>Add Eph Local Unspecified</td>
<td>Add ephemeral</td>
</tr>
<tr>
<td>Add Phy Resp</td>
<td>Add Phy Resp</td>
<td>Add physical response</td>
<td>Add Phy Resp</td>
<td>Add physical response</td>
</tr>
<tr>
<td>Add Eph Resp Local Specified</td>
<td>Add Eph Resp Local Specified</td>
<td>Add ephemeral response</td>
<td>Add Eph Resp Local Specified</td>
<td>Add ephemeral response</td>
</tr>
<tr>
<td>Modify Eph</td>
<td>Modify Eph</td>
<td>Modify ephemeral</td>
<td>Modify Eph</td>
<td>Modify ephemeral</td>
</tr>
<tr>
<td>Modify Resp</td>
<td>Modify Resp</td>
<td>Modify response</td>
<td>Modify Resp</td>
<td>Modify response</td>
</tr>
<tr>
<td>Clear Forward</td>
<td>Clear Forward</td>
<td>Clear forward</td>
<td>Clear Forward</td>
<td>Clear forward</td>
</tr>
<tr>
<td>Notify OE:R2/clear</td>
<td>Notify OE:R2/clear</td>
<td>Notify address</td>
<td>Notify OE:R2/clear</td>
<td>Notify address</td>
</tr>
<tr>
<td>Clear Back</td>
<td>Clear Back</td>
<td>Clear back</td>
<td>Clear Back</td>
<td>Clear back</td>
</tr>
<tr>
<td>Notify Resp</td>
<td>Notify Resp</td>
<td>Notify response</td>
<td>Notify Resp</td>
<td>Notify response</td>
</tr>
<tr>
<td>REL</td>
<td>REL</td>
<td>Release</td>
<td>REL</td>
<td>Release</td>
</tr>
<tr>
<td>RLC</td>
<td>RLC</td>
<td>Releasing connection</td>
<td>RLC</td>
<td>Releasing connection</td>
</tr>
<tr>
<td>Sub Phy</td>
<td>Sub Phy</td>
<td>Sub physical</td>
<td>Sub Phy</td>
<td>Sub physical</td>
</tr>
<tr>
<td>Sub Eph</td>
<td>Sub Eph</td>
<td>Sub ephemeral</td>
<td>Sub Eph</td>
<td>Sub ephemeral</td>
</tr>
<tr>
<td>Sub Phy Resp</td>
<td>Sub Phy Resp</td>
<td>Sub physical response</td>
<td>Sub Phy Resp</td>
<td>Sub physical response</td>
</tr>
</tbody>
</table>
The MGC generates a Modify command towards the R2 Trunking gateway with the seize event and an embedded seize acknowledgement signal. The digit map in this scenario is assumed to be provisioned in the MG (shown to be 2XXX which may not be true for practical R2 exchanges). The seize event has a embedded signal seizeack to be applied after detection of seize event. The embedded signal is accompanied by another embedded events namely the clear event to detect the clear forwards signal if generated from the R2 domain.

Step 1
MGC to R2TGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
    Context = - {
        Modify = Trunk1/line1 {
            Events = 1111 {R2/sz Embed { signals {R2/sa}, Events=1112
                { R2/clear  signals { R2/clear }, R2/address}}
            }
        }
    }
}
The R2TGW after receiving the Modify command does the command processing and responds with Modify command response.

Step 2
R2TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
    Context = - {
        Modify = Trunk1/line1
    }
}

In this example as mentioned earlier, the call is originated from a user connected to the R2 domain of PSTN. The R2 exchange that is connected to the R2 Trunking gateway generates the seize signal. The seize signal is identified by the R2TGW and seize event is detected. The seize event is notified to the MGC. The R2TGW meanwhile does the embedded descriptor processing for the seize event. The application of the seizeack and detection of clear event is activated on the specific circuit group.
The Notify command is generated towards the MGC.

Step 3
R2GW to MGC:

   MEGACO/1 [209.110.59.34]:25000
   Transaction = 2000 {
      Context = - {
         Notify = TermA {Observed Events =1111 {
            20010202T10000000:R2/sz}}
      }
   }

   the MGC responds the Notify command with the Notify response.

Step 4
MGC to R2GW:

   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2000 {
      Context = - {Notify = TermA}
   }

   The R2TGW collects the digits and reports the same to the MGC as parameters of the address event. The other observed event parameters like the source number, subscribe category, etc are also indicated to the MGC.

Step 5
R2GW to MGC:

   MEGACO/1 [209.110.59.34]:25000
   Transaction = 2001 {
      Context = - {
         Notify = TermA {Observed Events =1112 {
            20010302T10000000:R2/address { di=2992, si=2804 , sc=1 } }}
      }
   }

   the MGC responds the Notify command with the Notify response.

Step 6
MGC to R2GW:

   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2001 {
      Context = - {Notify = TermA}
   }

   The MGC then initiates the necessary signaling towards the exchange to which the destination user is connected. In this example the MGC has to generates SS7 messages to the SS7 switch. The IAM message is sent with all the necessary address signaling information. The SS7 switch responds with the ACM message.

   The MGC after receiving the ANM message generates command to apply answer signal. The MGC sends a Modify command towards the R2 TGW to
indicate that the answer needs to be applied

Step 7
MGC to R2TGW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1235 {
    Context = - {
      Modify = Trunk1/line1
      (signals{ R2/ans })
    }
  }
}

The R2trunking gateway responds to the Modify command.

Step 8
R2TGW to MGC:
  MEGACO/1 [207.176.47.89]: 26000
  Reply = 1235 {
    Context = - {
      Modify = Trunk1/line1
    }
  }
}

The MGC after receiving the ANM message generates commands to both the Trunking gateways to add physical circuits and create ephemeral terminations. The MGC in its message to the R2TGW in its signal descriptor lists the answer signal to indicate that the call establishment is successful and the end user is free to receive the call.

Step 9
MGC to R2TGW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1236 {
    Context = $ {
      Add = Trunk1/line1 (Media {
        LocalControl {Mode = SendRecv},
      },
      Add = $ (Media {
        LocalControl {
          Mode = Receiveonly,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        }
      }
    }
  }
}

Madhubabu, et al.
The R2 Trunking gateway after responds with a contextID in this example 1. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 10
R2TGW to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1236{
    Context = 1 {
      Add = Trunk1/line1,
      Add = EphA{
        Media {
          Local {
            v=0
            c=IN IP4 209.110.59.33
            m=audio 30000 RTP/AVP 4
          } ; RTP profile for G723 is 4
        }
      }
    }
  }

The MGC after receiving the response generates similar transaction with two ADD commands to the other TGW. The local SDP information specified in by the R2TGW is used to as remote SDP information for the other TGW. The MGC conveys this information in the Add command.

Step 11
MGC to TGW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1237{
    Context = $ {
      Add = Trunk2/$ {Media {
        LocalControl {Mode = SendRecv}},
      },
      Add = $ {Media {
        LocalControl {
          Mode = Receiveonly,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        }
      }}
    }
  }

Madhubabu, et al.
The Trunking gateway creates a context with ContextId 2. It adds the physical termination Trunk2/line1 in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 12
TGW to MGC:
  MEGACO/1 [207.176.47.89]: 26000
  Reply = 1237{
    Context = 2 {
      Add = Trunk2/line1,
      Add = EphB{
        Media {
          Local {
            v=0
            c=IN IP4 207.176.47.90
            m=audio 40000 RTP/AVP 4
          }
        } ; RTP profile for G723 is 4
      }
    }
  }

The MGC after receiving the response generates a modify command to the R2TGW to inform the local SDP information of TGW as the remote SDP for the R2TGW.

Step 13
MGC to R2TGW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1238{
    Context = 1 {
      Modify = EphA
      Media {
        LocalControl {
          Mode = SendRecv,
        },
        Remote{
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
  }
The R2 Trunking gateway responds to the Modify command.

Step 14
R2TGW to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1238{
    Context = 1 {
        Modify = EphA
    }
}
```

The RTP media transfer takes place. The user connected to the R2 exchange domain terminates the call. The clear forwards signal generated by the R2 exchange connected to the R2 TGW is detected and reported to the MGC as the clear event in the R2 package. The clear back signal, which is part of the clear event, enables the R2TGW to generate the clear back signal. The clear event detection is reported to MGC as part of the Notify command.

Step 15
R2TGW to MGC:

```
MEGACO/1 [209.110.59.34]:25000
Transaction = 2003 {
    Context = 1 {
        Notify = TermA {ObservedEvents =1112 {
            20010402T10030000:R2/clear}
    }
}
```

The MGC responds to the RGWs Notify message.

Step 16
MGC to R2TGW:

```
MEGACO/1 [216.33.33.61]:27000
Reply = 2003 {
    Context = 1 {
        Notify = TermA
    }
}
```

The MGC generates necessary command to the SS7 switch like the REL message. The SS7 switch responds with the RLC message. The MGC now generates commands to both the TGWs for subtracting the terminations from the contexts created.

Step 17
MGC to R2TGW:

```
MEGACO/1 [216.33.33.61]: 27000
```
The R2TGW responds to the subtract commands generated by MGC.

Step 18
TGW to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1239{
  Context = 1 {
    Subtract = Trunk1/line1{Audit{ }},
    Subtract = EphA {Audit{Statistics}}
  }
}

The MGC generates similar transaction with two Subtract command for subtracting both the physical and ephemeral terminations from the second Trunking gateway.

Step 19
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240
Context = 2 {
  Subtract = Trunk2/line1{Audit{ }},
  Subtract = EphB {Audit{Statistics}}
}

The RGW responds to the subtract commands generated by MGC.

Step 20
TGW to MGC:
MEGACO/1 [207.176.47.89]:26000
Reply = 1240
Context = 2 {
  Subtract = Trunk2/line1
  Subtract = EphB {
    Statistics {
      rtp/ps=987, ; packets sent
      nt/os=65432, ; octets sent
      rtp/pr=1234, ; packets received
      nt/or=56789, ; octets received
      rtp/pl=10, ; % packets lost
      rtp/jit=30,
      rtp/delay=30 ; average latency
    }
  }
}

### 2.9 Call between R1 trunk in TGW and SS7 trunk in TGW

In the earlier section we illustrated the Megaco messages between MGC and two Trunking gateway one that perform CCS SS7 and the other CAS R2 signaling. This section illustrates another similar scenario, but now the call is initiated by the user that originates R1 signaling to the user connected to CCS SS7 signaling. Both the Trunking gateways are assumed to be controlled by the same Media Gateway Controller. The packages that are considered are R1 package, network package, MF tone detection package and RTP package.

<table>
<thead>
<tr>
<th>R1Exchange</th>
<th>TGW1</th>
<th>MGC</th>
<th>TGW2</th>
<th>SS7 Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;-----</td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>Modify ED:sz(SD:sa ED:addr, cl)</td>
<td></td>
<td></td>
<td>Modify Resp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>------</td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>Seize</td>
<td></td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>SeizeAck</td>
<td>&lt;------</td>
<td></td>
<td>Notify OE:sa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>------</td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>KP</td>
<td></td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>Digit1</td>
<td>&lt;------</td>
<td></td>
<td>Notify OE:addr</td>
<td>IAM</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>------</td>
<td></td>
<td>------</td>
<td>------------</td>
</tr>
</tbody>
</table>
Internet-Draft      Megaco/H.248 Call flow examples       July 2001

| <- --------- | IAM       |
| Notify Resp  |
| <- --------- | ACM       |
| Modify Phy   |

Answer

---------->
Modify Resp
<- ---------
Add Phy
Add Eph Local Unspecified
---------->
Add Phy Resp
Add Eph Resp Local Specified
---------->
Add Phy
Add Eph Local Unspecified
Remote Specified
<- ---------
Add Phy Resp

<- ---------
Modify Eph
---------->
Modify Resp

---------->
RTP MEDIA

---------->
Clear Forward

---------->
Notify OE:R1/clear

Clear Back

<- ---------
Notify Resp

---------->
REL

---------->
REL

---------->
RLC

<- ---------
Sub Phy
Sub Eph

---------->
Sub Phy Resp
Sub Eph Resp Statistics
---------->
Sub Phy
Sub Eph

Madhubabu, et al.                                             [Page 86]
The MGC generates a Modify command towards the R1 Trunking gateway with the seize event and an embedded seize acknowledgement signal. The digit map in this scenario is assumed to be provisioned in the MG (shown to be 2XXX which may not be true for practical R1 exchanges). The seize event has a embedded signal seizeack to be applied after detection of seize event. The embedded signal is accompanied by another embedded events namely the clear event to detect the clear forwards signal if generated from the R1 domain.

Step 1
MGC to R1TGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
    Context = - {
        Modify = Trunk1/line1 {
            Events=1111{R1/sz Embed{ signals {R1/sd}, Events=1112 { mfd/ce{dmap}, R1/clearforward signals { R1/clearback }, R1/address}}
        }
    }
}
The R1TGW after receiving the Modify command does the command processing and responds with Modify command response.

Step 2
R1TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
    Context = - {
        Modify = Trunk1/line1
    }
}

In this example as mentioned earlier, the call is originated from a user connected to the R1 domain of PSTN. The R1 exchange that is connected to the R1 Trunking gateway generates the seize signal. The seize signal is identified by the R1TGW and seize event is detected. The seize event is notified to the MGC. The R1TGW meanwhile does the embedded descriptor processing for the seize event. The application of the seizeack and detection of clear event is activated on the specific circuit group. The Notify command is generated towards the MGC.

Step 3
R1GW to MGC:

    MEGACO/1 [209.110.59.34]:25000
    Transaction = 2000 {
        Context = - {
            Notify = Trunk1/linel {Observed Events =1111 {
                20010202T10000000:R1/sz}}
        }
    }

the MGC responds the Notify command with the Notify response.

Step 4
MGC to R1GW:

    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2000 {
        Context = - {Notify = Trunk1/linel}
    }

The R1TGW collects the digits and reports the same to the MGC as parameters of the address event The digit completion event of the MF detection package is used here. This is extracted from the MF tone detection package.

Step 5
R1GW to MGC:

    MEGACO/1 [209.110.59.34]:25000
    Transaction = 2001 {
        Context = - {
            Notify = Trunk1/linel {Observer Events =1112 {
                20010302T10000000:mfd/ce{ds=2992}}
        }
    }

the MGC responds the Notify command with the Notify response.

Step 6
MGC to R1GW:

    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2001 {
        Context = - {Notify = Trunk1/linel}
    }

The MGC then initiates the necessary signaling towards the exchange to which the destination user is connected. In this example the MGC has to generates SS7 messages to the SS7 switch. The IAM message is sent with all the necessary address signaling information. The SS7 switch responds with the ACM message.

The MGC sends a Modify command towards the R1 TGW to indicate that has gone offhook. The answer signal is sent in the Signals descriptor.

Step 7
MGC to R1TGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = - {
    Modify = Trunk1/line1
    {signals{ R1/ans } }
  }
}

The R1 Trunking gateway responds to the Modify command.

Step 8
TGW to MGC:
  MEGACO/1 [207.176.47.89]: 26000
  Reply = 1235 {
    Context = - {
      Modify = Trunk1/line1
    }
  }

The MGC after receiving the ANM message generates commands to both the
Trunking gateways to add physical circuits and create ephemeral
terminations. The MGC in its message to the R1TGW in its signal
descriptor lists the answer signal to indicate that the call
establishment is successful and the end user is free to receive the
call.

Step 9
MGC to R1GW
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1236 {
    Context = $ {
      Add = Trunk1/line1 {Media {
        LocalControl {Mode = SendRecv}},
      },
      Add = $ {Media {
        LocalControl {
          Mode = Receiveonly,
        },
        Local {
          v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
        }
      }
    }
  }

The R1 Trunking gateway after responds with a contextID in this

Madhubabu, et al.                                             [Page 89]
example 1. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 10
R1TGW to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = Trunk1/line1, 
    Add = Epha{
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33 
          m=audio 30000 RTP/AVP 4
        }
      ; RTP profile for G723 is 4
      }
    }
  }
}
```

The MGC after receiving the response generates similar transaction with two ADD commands to the other TGW. The local SDP information specified in by the R1TGW is used to as remote SDP information for the other TGW. The MGC conveys this information in the Add command.

Step 11
MGC to TGW

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
  Context = $ {
    Add = Trunk2/$ {Media {
        LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
        LocalControl {
          Mode = Receiveonly,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        }
      ; RTP profile for G723 is 4
      }
    }
}}
```
The Trunking gateway creates a context with ContextId 2. It adds the physical termination Trunk2/line1 in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 12
R1GW to MGC:

```plaintext
MEGACO/1 [207.176.47.89]: 26000
Reply = 1237 {
  Context = 2 {
    Add = Trunk2/line1,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      } ; RTP profile for G723 is 4
    }
  }
}
```

The MGC after receiving the response generates a modify command to the R1TGW to inform the local SDP information of TGW as the remote SDP for the R1TGW.

Step 13
MGC to R1TGW

```plaintext
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
  Context = 1 {
    Modify = EphA{
      Media {
        LocalControl {
          Mode = SendRecv,
        },
        Remote{
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
  }
}
```
Internet-Draft      Megaco/H.248 Call flow examples       July 2001

The R1 Trunking gateway responds to the Modify command.

Step 14
R1TGW to MGC:
  MEGACO/1 [207.176.47.89]: 26000
  Reply = 1238 {
    Context = 1 {
      Modify = EphA
    }
  }

The RTP media transfer takes place. The user connected to the R1
exchange domain terminates the call. The clear forwards signal
generated by the R1 exchange connected to the R1 TGW is detected and
reported to the MGC as the clear event in the R1 package. The
clear back signal, which is part of the clear event, enables the R1TGW
to generate the clear back signal. The clear event detection is
reported to MGC as part of the Notify command.

Step 15:
R1TGW to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Transaction = 2003 {
    Context = 1 {
      Notify = TermA {ObservedEvents =1112 {
        20010402T10030000:R1/clear}
      }
    }
  }

The MGC responds to the R1GWs Notify message.

Step 16
MGC to R1TGW:
  MEGACO/1 [216.33.33.61]:27000
  Reply= 2003 {
    Context = 1 {
      Notify = TermA
    }
  }

The MGC generates necessary command to the SS7 switch like the
REL message. The SS7 switch responds with the RLC message. The MGC now
generates commands to both the TGWs for subtracting the terminations
from the contexts created.

Step 17
MGC to R1TGW:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1239 {
    Context = 1 {
      Subtract = Trunk1/line1{Audit{ }},
      Subtract = EphA {Audit{Statistics}}
    }
  }

Madhubabu, et al.                                             [Page 92]
The R2TGW responds to the subtract commands generated by MGC.

Step 18
R1TGW to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Reply = 1239 {
        Context = 1 {
            Subtract = Trunk1/line1
            Subtract = EphA {
                Statistics {
                    rtp/ps=987, ; packets sent
                    nt/os=65432, ; octets sent
                    rtp/pr=1234, ; packets received
                    nt/or=56789, ; octets received
                    rtp/pl=10, ; % packets lost
                    rtp/jit=30,
                    rtp/delay=30 ; average latency
                }
            }
        }
    }

The MGC generates similar transaction with two Subtract command for subtracting both the physical and ephemeral terminations from the second Trunking gateway.

Step 19
MGC to TGW:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1240 {
        Context = 2 {
            Subtract = Trunk2/line1{Audit{ }},
            Subtract = EphB {Audit{Statistics}}
        }
    }

The R1GW responds to the subtract commands generated by MGC.

Step 20
TGW to MGC:
    MEGACO/1 [207.176.47.89]:26000
    Reply = 1240
    Context = 2 {
        Subtract = Trunk2/line1
        Subtract = EphB {
            Statistics {
                rtp/ps=987, ; packets sent
                nt/os=65432, ; octets sent
                rtp/pr=1234, ; packets received
                nt/or=56789, ; octets received
                rtp/pl=10, ; % packets lost
            }
        }
    }
2.10 Call between SS7 trunk in TGW and R1 trunk in TGW
This section of the call flow illustrates the call between two Trunking gateways, one that does R1 signaling towards one end and the second TGW CCS SS7. In this example the user in the CCS SS7 signaling network originates the call. The destination user is connected to the PSTN network with R1 signaling.
When the MGC through the signaling gateway receives the IAM it generates a Modify message towards the Trunking gateway that supports R1 package. The signal descriptor has the seize signal and the events descriptor has the event seizeack. The seizeack event has an embedded signal "address". The Trunking gateway is intended to seize a circuit and apply the address signals after the seize acknowledgement is received from the other R1 exchange. The calling party information can also be provided in the address signal. For simplicity it is omitted here. The embedded event of the seize acknowledgement event is the answer. This event has to be reported when the digits reach the other R1 exchange and there is answer from the terminating R1 exchange. The message from MGC to R1TGW is as follows.

Step 1
MGC to R1TGW
MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1234 {
    Context = - {
      Modify = Trunk1/line1 {
        Signals = { R1/sz}
        Events = 1111 {R1/sd Embed { signals {R1/address { ds = 2989} },
                        Events=1112 { R1/clear, R1/ans}
                      }
      }
    }
  }

The R1TGW after receiving the Modify command does the command processing and responds with Modify command response.

Step 2
R1TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
  Reply = 1234 {
    Context = - {
      Modify = Trunk1/line1
    }
  }

The R1 TGW applies seize signal on the specified circuit group and after receiving the acknowledgement from the other R1 exchange updates the embedded events descriptor as the events descriptor. The R1TGW also generates a Notify command towards the MGC to indicate that the seize acknowledgement has been received.

Step 3
R1GW to MGC:
    MEGACO/1 [209.110.59.34]:25000
Transaction = 2000 {
    Context = - {
        Notify = Trunk1/line1{Observed Events =1111 {
            20010202T10000000:R1/sd}}
    }
}

the MGC responds the Notify command with the Notify response.

Step 4
MGC to R1GW:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2000 {
        Context = - {Notify = Trunk1/line1}
    }

The MGC now generates the address complete message to the SS7 switch that has generated the IAM message. The R1TGW after receiving the answer event from the other R1 exchange generates message to notify this event.

Step 5
R1GW to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Transaction = 2001 {
        Context = - {
            Notify = Trunk1/line1{Observed Events =1112 {
                20010202T10000000:R1/ans}}
        }
    }

the MGC responds the Notify command with the Notify response.

Step 6
MGC to R1GW:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2001 {
        Context = - {Notify = Trunk1/line1}
    }

MGC generates the ANM message towards the SS7 switch through the signaling gateway. It also generates commands to add terminations into specific contexts. It generates a single transaction with two Add commands towards the TGW that supports R1 terminations. One command is to add the physical circuit group Trunk1/line and the other for the ephemeral termination. The SDP information is unspecified that enables the TGW to choose the underspecified parameters.

Step 7
MGC to R1TGW
    MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235{
  Context = $ {
    Add = Trunk1/line1 {Media {
        LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
        LocalControl {
            Mode = Receiveonly,
        },
        Local {
            v=0
c=IN IP4 $ 
m=audio $ RTP/AVP 4
        }
    }
}
}
}

The R1 Trunking gateway after receiving the Add command responds with a contextID in this example 1. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 8
R1TGW to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1235{
    Context = 1 {
      Add = Trunk1/line1,
      Add = EphA{
        Media {
          Local {
            v=0
c=IN IP4 209.110.59.33 
m=audio 30000 RTP/AVP 4
        }
      } ; RTP profile for G723 is 4 
    }
  }
}

The MGC after receiving the response generates similar transaction with two ADD commands to the other TGW. The local SDP information specified in by the R1TGW is used to as remote SDP information for the other TGW. The MGC conveys this information in the Add command.
Step 9

MGC to TGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236{
  Context = {$
    Add = Trunk2/$ {Media {
      LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
    } ; RTP profile for G723 is 4
  } Remote{
    v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
  }
} ; RTP profile for G723 is 4
}

The Trunking gateway creates a context with ContextId 2. It adds the physical termination Trunk2/line1 in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 10

RGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1236{
  Context = 2 {
    Add = Trunk2/line1,
    Add = EphB{
      Media {
        Local {
          v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
        } ; RTP profile for G723 is 4
      }
    }
  }
}

The MGC after receiving the response generates a modify command to
the R1 Trunking gateway to inform the local SDP information of TGW as the remote SDP for the R1 TGW.

Step 11
MGC to TGW

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237{
  Context = 1 {
    Modify = EphA
    {Media {
      LocalControl {
        Mode = SendRecv,
      },
      Remote{
        v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
      }
    }
  }
}
```

The R1 Trunking gateway responds to the Modify command.

Step 12
TGW to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1237{
  Context = 1 {
    Modify = EphA
  }
}
```

The RTP media flow is established and the two users connected to the SS7 trunk and R1 trunk starts the conversation. After conversation any of the user can disconnect the call in this example the user connected to the SS7 domain releases the call. The SS7 switch generates a REL message towards the MGC. The Signaling gateway forwards the same request towards the MGC. The MGC initiates the tearing down of the call. This is done initially by generating a Modify command towards the R1 TGW to generate clear forward signal towards the R1 exchange. In the message the MGC sends a signal descriptor with clear signals and events descriptor with the clear in the event. The event in the events descriptor is for detecting the clear back signal generated from the R1 exchange.

Step 13
MGC to TGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238{
```

Madhubabu, et al. [Page 100]
The R1 TGW after receiving the commands does the signals and events descriptor processing and responds to the MGC with the Modify command response.

Step 14
R1TGW to MGC
MEGACO/1 [209.110.59.34]: 25000
   Reply = 1238 {
      Context = 1 {
         Modify = Trunk1/line1
      }
   }

Meanwhile the MGC generates the subtract message towards the originating TGW to remove the terminations from the newly created context.

Step 15
MGC to TGW:
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1239
      Context = 2 {
         Subtract = Trunk2/line1{Audit{ }},
         Subtract = EphB {Audit{Statistics}}
      }
   }

The RGW responds to the subtract commands generated by MGC.

Step 16
TGW to MGC:
   MEGACO/1 [207.176.47.89]: 26000
   Reply = 1239
      Context = 2 {
         Subtract = Trunk2/line1
         Subtract = EphB {
            Statistics {
               rtp/ps=987, ; packets sent
               nt/os=65432, ; octets sent
               rtp/pr=1234, ; packets received
               nt/or=56789, ; octets received
               rtp/pl=10, ; % packets lost
               rtp/jit=30,
               rtp/delay=30 ; average latency
            }
         }
      }

The MGC after receiving the response from the TGW generates the Release complete message RLC towards the SS7 switch through the Signaling Gateway.

The R1 TGW after receiving the clear back signal from the other R1 exchange detects it and generates a Notify command towards the MGC.

Step 17
R1GW to MGC:
   MEGACO/1 [209.110.59.34]:25000
   Transaction = 2002 {
      Context = - {
         Notify = TermA {Observed Events =1113 {
            20030202T10000000:R1/clear}}
      }
   }

the MGC responds the Notify command with the Notify response.

Step 18
MGC to R1GW:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2002 {
      Context = - {Notify = TermA}
   }
MGC after receiving the notify command generates subtract command to remove both the physical termination and ephemeral termination.

Step 19
MGC to R1TGW:
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1240{
      Context = 1 {
         Subtract = Trunk1/line1{Audit{ }},
         Subtract = EphA {Audit{Statistics}}
      }
   }
The R1TGW responds to the subtract commands generated by MGC.

Step 20
R1TGW to MGC:
   MEGACO/1 [209.110.59.34]:25000
   Reply = 1240{
      Context = 1 {
         Subtract = Trunk1/line1
         Subtract = EphA {
            Statistics {
               rtp/ps=987, ; packets sent
```
2.11 Call between ISDN trunk in TGW and SS7 trunk in TGW

This section illustrates the Megaco message flow between MGC and two Trunking media gateways one that perform ISDN signaling towards the user and the other TGW performs SS7 signaling. In this example we assume that an ISDN user initiates that call.

<table>
<thead>
<tr>
<th>ISDNSwitch</th>
<th>TGW1/SG</th>
<th>MGC</th>
<th>TGW2/SG</th>
<th>SS7Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setup</td>
<td></td>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Setup</td>
<td>IAM</td>
<td>IAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IAM</td>
<td>ACM</td>
<td>ACM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alerting</td>
<td></td>
<td>Alerting</td>
<td>Disconnect</td>
</tr>
<tr>
<td></td>
<td>Connect</td>
<td></td>
<td>Connect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add Phy</td>
<td></td>
<td>Add Phy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add Eph</td>
<td></td>
<td>Add Eph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local unspecfied</td>
<td></td>
<td>Local Specified</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add Phy Resp</td>
<td></td>
<td>Add Phy Resp Local Specified</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add Eph Resp Local Specified</td>
<td></td>
<td>Add Eph Local Unspecified Remote Specified</td>
<td></td>
</tr>
</tbody>
</table>

Now the termination is free to take part in other calls.
The initial message sent by the ISDN switch is the Setup message. The setup message has all the address information. The MGC after receiving the Setup message from the Signaling gateway generates the IAM message towards the SS7 switch through the Signaling gateway. After receiving the ACM address complete message from the SS7 switch the MGC generates Alerting message towards the ISDN switch. The MGC after receiving the ANM message generates Connect message towards the ISDN switch. The MGC now adds terminations in both the Trunking gateways.

Step 1
MGC to ISDNTGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234{
  Context = $ {
    Add = Trunk1/line1 {Media {
      LocalControl {Mode = SendRecv},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
      }
    }
  }
  }
}

The ISDN Trunking gateway after receiving the Add command responds
with a contextID in this example 1. The ephemeral termination is
created and added to the same context as the physical termination.
The ephemeral termination added in this example is EPHA. The local
parameters are specified in the response. The IP address chosen for
the media transport in this example is 209.110.59.33, and the
port number specified 30000.

Step 2
ISDNTGW to MGC:
  MEGACO/1 [207.176.47.89]: 25000
Reply = 1234{
  Context = 1 {
    Add = Trunk1/line1,
    Add = EphA{
      Media {
        Local {
          v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
      }
    }
  }
}; RTP profile for G723 is 4
}

The MGC after receiving the response generates similar transaction
with two ADD commands to the other SS7TGW. The local SDP information
specified in by the ISDNTGW is used to as remote SDP information for
the other TGW. The MGC conveys this information in the Add command.
Step 3
MGC to SS7GW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235{
    Context = $ {
        Add = Trunk2/$ {Media {
            LocalControl {Mode = SendRecv},
        }},
        Add = $ {Media {
            LocalControl {
                Mode = Receiveonly,
            },
            Local {
                v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
            }
        }},
        Remote{
            v=0
c=IN IP4 209.110.59.33
        m=audio 30000 RTP/AVP 4
        } ; RTP profile for G723 is 4
    }
}

The Trunking gateway creates a context with ContextId 2. It adds the physical termination Trunk2/line1 in that context. The ephemeral termination EPHB is created with the specified SDP information. The response from the MG specifies the local SDP information.

Step 4
RGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1235{
    Context = 2 {
        Add = Trunk2/line1,
        Add = EPHB{
            Media {
                Local {
                    v=0
c=IN IP4 209.110.59.33
        m=audio 30000 RTP/AVP 4
            }
            Remote{
                v=0
c=IN IP4 209.110.59.33
                m=audio 30000 RTP/AVP 4
            }
        } ; RTP profile for G723 is 4
    }
}

Madhubabu, et al. [Page 106]
The MGC after receiving the response generates a modify command to the ISDNTGW to inform the local SDP information of TGW as the remote SDP for the ISDNTGW.

Step 5
MGC to TGW
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1236{
        Context = 1 {
            Modify = EphA
            Media {
                LocalControl {
                    Mode = SendRecv,
                },
                Remote{
                    v=0
                    c=IN IP4 209.110.59.33
                    m=audio 30000 RTP/AVP 4
                }
            }
        }
    }

The ISDN Trunking gateway responds to the Modify command.

Step 6
ISDNTGW to MGC:
    MEGACO/1 [207.176.47.89]: 26000
    Reply = 1236{
        Context = 1 {
            Modify = EphA
        }
    }

The RTP flow is established. The call can be terminated either from the ISDN user or from the user connected to SS7 domain. In this example we assume that the ISDN user terminates the call. The ISDN switch generates Disconnect message towards the MGC through Signaling gateway. The MGC generates Release message towards the SS7 switch. The MGC also generates messages to subtract termination from contexts in both the Trunking gateways.

Step 7
MGC to ISDNTGW:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1237{
        Context = 1 {
            Subtract = Trunk1/line1{Audit{ }},
            Subtract = EphA {Audit{Statistics}}
        }
    }

Madhubabu, et al.
The ISDNTGW responds to the subtract commands generated by MGC.

Step 8
ISDNTGW to MGC:
   MEGACO/1 [209.110.59.34]:25000
   Reply = 1237
   Context = 1 {
       Subtract = Trunk1/line1
       Subtract = EphA {
           Statistics {
               rtp/ps=987, ; packets sent
               nt/os=65432, ; octets sent
               rtp/pr=1234, ; packets received
               nt/or=56789, ; octets received
               rtp/pl=10, ; % packets lost
               rtp/jit=30,
               rtp/delay=30 ; average latency
           }
       }
   }

The MGC generates similar transaction with two commands towards the TGW that terminates SS7 media trunks.

Step 9
MGC to TGW:
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1238
   Context = 2 {
       Subtract = Trunk2/line1{Audit{ }},
       Subtract = EphB {Audit{Statistics}}
   }

The SS7TGW responds to the subtract commands generated by MGC.

Step 10
TGW to MGC:
   MEGACO/1 [209.110.59.34]:26000
   Reply = 1238
   Context = 2 {
       Subtract = Trunk2/line1
       Subtract = EphB {
           Statistics {
               rtp/ps=987, ; packets sent
               nt/os=65432, ; octets sent
               rtp/pr=1234, ; packets received
               nt/or=56789, ; octets received
               rtp/pl=10, ; % packets lost
               rtp/jit=30,
               rtp/delay=30 ; average latency
           }
       }
   }
2.12 Continuity test from TGW
In this section we will illustrate the usage of Megaco command for performing continuity tests. The basic continuity package as defined in the protocol is considered. The procedures specified in the package are illustrated. There are two cases in the continuity test. One in which the MGC generates Megaco commands towards MG to initiate an continuity test and the second one in which the command from MGC enables the gateway to return any continuity test originated from switched circuit network.

Case (a)

<table>
<thead>
<tr>
<th>MG</th>
<th>MGC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;---------------------------</td>
</tr>
<tr>
<td></td>
<td>Modify TermA SD:ct/ct ED:ct/cmp state=test</td>
</tr>
</tbody>
</table>
|                     | --------------------------->
|                     | Modify TermA Resp |
|                     | ----------------------------------|
|                     | Continuity Signal |
|                     | ----------------->|
|                     | Continuity Signal Resp |
|                     | --------------------------->
|                     | Notify TermA OE:ct/cmp {resp=success} |
|                     | --------------------------->
|                     | Notify TermA Resp |

The case of originating Continuity test by the Trunking gateway is considered in this section. The MGC intends to check the continuity of a specified circuit group linel of the trunk Trunk1 in the Trunking gateway. The MGC generates a modify command with the termination state set to "test". The event descriptor specifies the "cmp" completion event of the continuity package. The Signal descriptor lists the "ct" continuity test signal.

Step 1
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
    Context = '-' {
        Modify = Trunk1/line1 {Media {
            TerminationState {ServiceState = test}}
        Signals = { ct/ct }
        Events = 1111 { ct/cmp }
    }
}

The TGW after receiving the Modify command for termination Trunk1/line1 in Null context, updates the termination state. The event descriptor is updated in the Trunking gateway. The signal "ct" is applied on the specified termination. The Trunking gateway now states for detecting any tones/signals on the line on which the continuity test signal was applied. The response for the message generated by MGC is sent back.

Step 2
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1234 {
    Context = '-' {
        Modify = Trunk1/line1
    }
}

The Trunking gateway detects for any return tone/signal. The frequency of the tone is assumed to be provisioned at the TGW. As soon as the response is received from the other side of the trunk the Trunking gateway reports the same to the MGC in the form of event detection. The event "cmp" in the event descriptor is used to notify the observed activity on the termination. The parameter value for the response parameter indicates the result of the continuity test performed.

Step 3
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Transaction = 2000 {
    Context = '-' {
        Notify = Trunk1/line1 {ObservedEvents =1111 {
            20010202T10000000: ct/cmp { res=success}}
        }
    }
}

The MGC after receiving the Notify message responds to the MGC with the response.

Step 4
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
    Context = - (Notify = Trunk1/line1)
Case (b)
In the earlier continuity test scenario we saw the Megaco messages exchanged between MG and MGC when the MGC initiated the continuity test. In some scenarios the MG should respond to the continuity test originated from the PSTN switches. In this section we will consider such a scenario.

The continuity package signal "rsp" is used for this purpose. This signal duration and frequency are provisioned at the TGW. When the MGC is intimated from the PSTN switch for responding the continuity test, the MGC generates a Modify command with the "rsp" signal in the signal descriptor. The termination state is set to "test".

Step 1
MGC to TGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = '-' {
    Modify = Trunk1/line1 {Media {
      TerminationState {ServiceState = test}
      LocalControl { mode = loopback} }
    Signals = { ct/rsp }
  }
}
}

There can be two possible cases for responding the continuity test originated from the PSTN network. One in which the Trunking gateway generates a signal of different frequency when it receives a continuity check specific signal. The second possibility is to respond with the same signal looped back to the same switch that originated it. In
this example we assume the case of Loopback. The mode of the termination is also set to loopback to facilitate this. The Trunking gateway after receiving the signal loops back the same signal. The Trunking gateway also responds with the transaction response message towards the MGC.

Step 2
TGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1234 {
  Context = '-' {
    Modify = Trunk1/line1
  }
}
Once the test is complete the MGC is intimated about the success of failure of the continuity test through the switch that originated the continuity test.

2.13 Call from residential gateway to H.323 Terminal.

In This section we illustrate a call between a Residential gateway user and to an endpoint in the H.323 domain. We assume that the call is initiated from the user of the Residential gateway and after dialing the digits the MGC does necessary mapping of the number to identify that the called number belongs to the H.323 network.
Internet-Draft Megaco/H.248 Call flow examples July 2001

|               |------------------>|                 |
|               |       ARQ         |                 |
|               |<------------------|                 |
|               |------------------------------------>|
|               |                Setup                |
|               |                   |<----------------|
|               |                   |       ARQ       |
|               |                   |---------------->|
|               |                   |       ACF       |
|               |<------------------------------------|
|               |               Alerting              |
|               |                   |                 |
|<------|               |    RingBack tone                  |
|               |----------------------|------------------|------------------|
|               |                Add Phy             |
|               |                   |                 |
|               |                   |       Add Eph Local Unspecified   |
|               |                   |------------------>|
|               |                   |                 |
|               |                   |       Add Eph Resp Local Specified|
|               |                   |<------------------------------------|
|               |                Add Phy Resp         |
|               |                   |                 |
|               |                   |                 |
|               |                   |       Connect              |
|               |                   |<----------------------|
|               |                   |                 |
|               |                   |       TCS             |
|               |                   |<----------------------|
|               |                   |                 |
|               |                   |       MSD             |
|               |                   |<----------------------|
|               |                   |                 |
|               |                   | OLC (forward Logcial ch)RGW rtp/rtcp|
|               |                   |<----------------------|
|               |                   | OLC (Rev Logical Ch) H.323 rtp/rtcp|
|               |                   |<----------------------|
|               |               Modify Eph Remote Specified|
|               |                   |<----------------------|
|               |               Modify Resp          |
|               |                   |<----------------------|
|               |               /----------------------|
|               |               RTP Media             |
|               |                   |<----------------------|
|               |                   | CLC SessionId = 1    |
|               |                   |<----------------------|
|               |               Modify SD:cg/bt       |
|               |                   |<----------------------|
|               |                   | CLC SessionId = 2    |
|               |                   |<----------------------|
|               |                   | Modify Resp           |
|               |                   |<----------------------|
|               |                   | Release Complete      |
|               |                   |<----------------------|
|               |                   | DRQ                   |
|               |                   |<----------------------|
|               |                   | Sub Phy               |

Madhubabu, et al. [Page 113]
The MGC generates modify command for to the residential gateway to check for offhook for Termination TermA. In this message the event offhook has an embedded signal descriptor and embedded event descriptor. The embedded signal descriptor is sent for application of dial tone immediately after the detection of the offhook event and the event descriptor then will be updated with the onhook and the digit map completion event. The Digit map is also defined in the digit map descriptor.

Step 1
MGC to RGW:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
    Context = - {
        Modify = TermA {
            Media {
                LocalControl {
                    Mode = ReceiveOnly
                },
                DigitMap = Dmap1 {(2XXX)}
            }
            Events = 1111 {al/of Embed {signals {cg/dt}}, Events = 1112 {al/on},
                            {dd/ce {Dmap1}}}
        }
    }
}

The MG after receiving the MGC’s message responds to it.

Step 2
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
    Context = - (Modify = TermA)
}

When the user A goes offhook RGW detects the offhook event and as it is listed in the event descriptor report the event detection using Notify command.

Step 3
RGW to MGC:

MEGACO/1 [209.110.59.34]: 25000
Transaction = 2000 {
    Context = - {
        Notify = TermA {ObservedEvents = 1111 {
            20010202T10000000:al/of}}
    }
}
The MGC responds with the Notify response.

Step 4
MGC to RGW:

    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2000 {
        Context = - {Notify = TermA}
    }

The digit map is active on the termination TermA. When the user dials the digits they are reported to MGC through Notify command.

Step 5
RGW to MGC:

    MEGACO/1 [209.110.59.34]: 25000
    Transaction = 2001 {
        Context = - {
            Notify = TermA {Observed Events =1112 {
                20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
        }
    }

MGC after receiving the Notify command responds back with the Notify response.

Step 6
MGC to RGW:

    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2001 {
        Context = - {Notify = TermA}
    }

The MGC analyses the digits sent by the Residential Gateway. The routing functionality in MGC enabled and it indicates that the dialed digits belong to the user in H.323 network. The MGC acts as a H.323 Terminal towards H.323 network and generates the signaling towards the destined Called party. The MGC initially generates an admission request towards the Gatekeeper. The Gatekeeper acknowledges the admission request message by generating the Admission confirmation message ACF. Assuming that the GK used directed-routed call model, the Gatekeeper provides the transport address information of the destination user. The MGC then initiates the H.225 signaling by generating the SETUP message towards the H.323 endpoint. The H.323 Endpoint after receiving the Setup message from the MGC initiates the Admission request towards the Gatekeeper. The H.323 Endpoint after receiving the Admission confirmation message from the Gatekeeper generates the Alerting message towards the MGC. The MGC after receiving the ALERT message from the H.323 Endpoint generates a Add command to the Residential gateway to apply ring back tone to the given termination. In the Add command the MGC requests the creation
of ephemeral termination and under specifies the Local SDP information. The mode of the termination is set to receive only.

Step 7
MGC to RGW:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1235 {
        Context = $ {
            Add = TermA {
                Signals { cg/ rt }
                Media {
                    LocalControl {
                        Mode = sendrecv,
                    },
                }
            Add = $ {
                Media {
                    LocalControl {
                        Mode = sendrecv,
                    },
                    Local {
                        v=0
                        c=IN IP4 $
                        m=audio $ RTP/AVP 4
                    }
                    }
                }
            }
        }
    }

MG after creating the context adds the physical termination TermA. In this case MG creates a context with ContextId 1. The ephemeral termination EphA is created and added to the context 1. The MG reserves resources for the local SDP information. In this case the IP address allocated is 209.110.59.33 and the port number used is 30000. The MG responds to the Add command with this information in the response to MGC.

Step 8
RGW to MGC:
    MEGACO/1 [209.110.59.34]: 25000
    Reply = 1235 {
        Context = 1 {
            Add = TermA,
            Add=EphA{
                Media {
                    Local {
                        v=0
                        c=IN IP4 209.110.59.33
                    }
                    }
                }
            }
        }
    }
The H.323 Endpoint generates the CONNECT message after the called party goes offhook. We assume that the H.245 information is passed in Connect message. The Terminal Capability set and the Master slave determination occurs between the MGC and the H.323 Endpoint. The Open Logical channel messages (OLCs) indicate the session and media related parameters. This enables the MGC to inform and receive the SDP information of both the users. By the end of this OLCs the MGC is aware of the SDP information of the H.323 Endpoint. The MGC now generates Modify message towards the MG, with the Remote SDP information for the ephemeral termination EphA.

Step 9:
MGC to RGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = 1 {
    Modify = TermA {
      Signals {} ; to turn off ringback tone
      Events = 1235 {al/on},
      Media {
        LocalControl {
          Mode = SendRecv,
        }
      }
    }
    Modify = EphA{
      Media {
        LocalControl {
          Mode = SendRecv,
        }
        Remote {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
  }
}
```

The RGW responds to the request from MGC.

Step 10
RGW to MGC:

```
Madhubabu, et al.                                             [Page 117]
```
Now RTP media flow takes place. In the example we assume that the 
H.323 goes onhook to terminate the call. The MGC initiates the tearing 
down of the Call by closing the logical channels that were earlier 
created for exchanging the H.245 information. The MGC after receiving 
the DISCONNECT message generates a Modify message towards the 
Residential gateway user for applying the busy tone and for making 
the mode of the two terminations to receive only.

Step 11
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
  Context = 1 {
    Modify = TermA {
      Signals {cg/bt}
      Media {
        LocalControl {
          Mode = recvonly}
        }
    },
    Modify = EphA {
      Media {
        LocalControl {
          Mode = recvonly}
      }
    }
  }
}
The RGW responds to this modify request.

Step 12
MG2 to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1240 {
  Context = 1 {
    Modify= TermA, Modify = EphA}
}

Step 13:
RGW1 to MGC:
MEGACO/1 [209.110.59.34]:25000
Transaction = 2002 {
  Context = 1 {
    Notify = TermA {ObservedEvents =1235{

Madhubabu, et al.                                             [Page 118]
The MGC after the Release Complete message waits for the Notify command with onhook event from UserA and after receiving subtracts the physical and ephemeral termination at the Residential gateway. Thus enabling the user to participate in further calls.

Step 15
MGC to RGW
MEGACO/1 [216.33.33.61]:27000
Transaction = 1238 {
    Context = 1 {
        Subtract = TermA {Audit{ }}
        Subtract = EphA {Audit(Statistics)}
    }
}

The MG subtracts the two terminations from the context. The context itself is deleted with subtract of the last termination from it. The RGW responds to this transaction from MGC with statistics on ephemeral termination.

Step 16
RGW to MGC
MEGACO/1 [209.110.59.34]:25000
Reply = 1238 {
    Context = 1 {
        Subtract = TermA
        Subtract = EphA {
            Statistics {
                rtp/ps=1234, ; packets sent
                nt/os=56789, ; octets sent
                rtp/pr=987, ; packets received
                nt/or=65432, ; octets received
                rtp/pl=10, ; % packets lost
                rtp/jit=30,
                rtp/delay=30 ; average latency
            }
        }
    }
}
2.14 Call from residential gateway to SIP user.
This section illustrates a call between a user connected to Residential gateway and another SIP user. It is assumed that the MGC acts as a SIP server agent. The MGC generates SIP messages towards the SIP user agent/called party. When the MGC receives the digits dialed by the user even though not explicitly shown in the figure, it communicates with a routing database and finds the SIP user agent’s URL.
OnHook |--------------------|                         |
|     Notify OnHook   |                         |
|<--------------------|------------------------>|
|    Notify Resp      |      BYE                |
|<--------------------|                         |
|  Sub TermA          |                         |
|  Sub EphA           |<------------------------|
|-------------------->|         200 OK          |
|     Sub Phy Resp    |                         |
|     Sub Eph Resp Statistics                   |

The MGC generates modify command for to the residential gateway to check for offhook for Termination TermA. In this message the event offhook has an embedded signal descriptor and embedded event descriptor. The embedded signal descriptor is sent for application of dial tone immediately after the detection of the offhook event and the event descriptor then will be updated with the onhook and the digit map completion event. The Digit map is also defined in the digit map descriptor.

Step 1
MGC to RGW:
MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = Receiveonly
        },
        DigitMap= Dmap1{(2XXX)}
      };
      Events = 1111 {al/of Embed {signals {cg/dt}, Events=1112 { al/on},
      {dd/ce {Dmap1}}}
    }
  }
}

The MG after receiving the MGC’s message responds to it.

Step 2
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1234 {
  Context = - (Modify = TermA)
}

When the user A goes offhook RGW detects the offhook event and as it is listed in the event descriptor report the event detection using
Notify command.

Step 3
RGW to MGC:
   MEGACO/1 [209.110.59.34]:25000
   Transaction = 2000 {
      Context = - {
         Notify = TermA {ObservedEvents =1111 {
            20010202T10000000:al/of}}
      }
   }
the MGC responds with the Notify response.

Step 4
MGC to RGW:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2000 {
      Context = - {Notify = TermA}
   }

The digit map is active on the termination TermA. When the user dials the digits they are reported to MGC through Notify command.

Step 5
RGW to MGC:
   MEGACO/1 [209.110.59.34]: 25000
   Transaction = 2001 {
      Context = - {
         Notify = TermA {ObservedEvents =1112 {
            20010202T10010000:dd/ce{ds="2992",Meth=FM}}}  
      }
   }
MGC after receiving the Notify command responds back with the Notify response.

Step 6
MGC to RGW:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2001 {
      Context = - {Notify = TermA}
   }

The MGC analyses the digits sent by the Residential Gateway. The routing functionality in MGC enabled and it indicates that the dialed digits belong to the SIP user. The MGC acts as a SIP User agent, and generates a INVITE message towards the SIP user.

INVITE sip:UserB@there.com SIP/2.0
   Via: SIP/2.0/UDP here.com:5060
The SDP information is not provided in this INVITE message. The MGC receives the 180-OK message from the SIP user to indicate that ringing is done towards that end.

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP here.com:5060
From: Julien<sip:UserB@here.com>
To: Bob<sip:UserA@there.com>;tag=8321234356
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Content-Length: 0

The MGC generates a Modify message to the Residential gateway to apply ringback tone to the given termination.

Step 7
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
    Context = $ {
        Modify = TermA {
            Signals { cg/rt }
        }
    }
}

The MG after receiving the Modify message applies ringback tone to the user and generates Modify response.

Step 8
RGW to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
    Context = - (Modify = TermA)
}

The MGC meanwhile receives the 200 OK message from the SIP user to indicate its SDP information.

SIP/2.0 200 OK
Via: SIP/2.0/UDP here.com:5060
From: Julien<sip:UserA@here.com>
To: Bob<sip:UserB@there.com>;tag=8321234356
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Contact: Julien<sip:UserB@there.com>
Content-Type: application/sdp
Content-Length: 147

v=0
o=UserB 2890844527 2890844527 IN IP4 there.com
s=Session SDP
c=IN IP4 207.176.47.90
t=0 0
m=audio 35000 RTP/AVP 4
a=rtpmap:0 PCMU/8000

The MGC now generates the Add command for adding the physical termination TermA and to create an ephemeral termination EphA. The local SDP information for the ephemeral termination EphA is under specified to enable the RGW to allocate the necessary values by itself. The Remote SDP information is also sent. The mode of the two terminations is set to send receive.

Step 9
MGC to RGW:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = $ {
    Add = TermA {
      Media {
        LocalControl {
          Mode = sendrecv,
        },
      }
      Add = $ {
        Media {
          LocalControl {
            Mode = sendrecv,
          },
          Local {
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 4
          }
        }
      }
      Remote{
        v=0
        c=IN IP4 207.176.47.90
        m=audio 35000 RTP/AVP 4
      }
    }
  }
}
MG after creating the context adds the physical termination TermA. In this case MG creates a context with ContextId 1. The ephemeral termination EphA is created and added to the context 1. The MG reserves resources for the local SDP information. In this case the IP address allocated is 209.110.59.33 and the port number used is 30000. The MG responds to the Add command with this information in the response to MGC.

Step 10
RGW to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
a=sendrecv
        } ; RTP profile for G.723 is 4
      }
    }
  }
}
```

In the ACK message of SIP, the local SDP information is indicated towards the remote SIP user.

```
ACK sip:UserB@there.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: Bob<sip:UserA@here.com>
To: Julien<sip:UserB@there.com>;tag=8321234356
Call-ID: 12345601@here.com
CSeq: 1 ACK
Content-Length: 147

  v=0
  o=UserA 2890844526 2890844526 IN IP4 here.com
  s=Session SDP
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
a=rtpmap:0 PCMU/8000
```

Now RTP media flow takes place. In the example we assume that the UserA goes onhook to terminate the call. The RGW after detecting the
event reports the same to MGC in the Notify command.

Step 11:
RGW to MGC:

MEGACO/1 [209.110.59.34]:25000
Transaction = 2002 {
  Context = 1 {
    Notify = TermA {ObservedEvents = 1112 {
      20010202T10030000:al/on}
    }
  }
}
The MGC responds to the MGIs Notify message.

Step 12
MGC to MG1:

MEGACO/1 [216.33.33.61]:27000
Reply = 2002 {
  Context = 1 {
    Notify = TermA
  }
}
The MGC now generates subtract commands to the RGW. The BYE message is generated by the MGC towards the other SIP user.

BYE sip:UserB@here.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: Bob<sip:UserA@there.com>;tag=8321234356
To: Julien<sip:UserB@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0

The MGC after sending the BYE message recives the 200 OK message from the remote SIP user.

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: Julien<sip:UserB@there.com>;tag=8321234356
To: Bob<sip:UserA@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0

The two Subtract commands are clubbed in a single action.

Step 13
MGC to RGW

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
  Context = 1 {

Madhubabu, et al. [Page 126]
The MG subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 14
RGW to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Reply = 1238 {
    Context = 1 {
      Subtract = TermA
      Subtract = EphA {
        Statistics {
          rtp/ps=1234, ; packets sent
          nt/os=56789, ; octets sent
          rtp/pr=987, ; packets received
          nt/or=65432, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }

3 Service Change Command Usage

This section lists the few methods that illustrate the usage of MEGACO protocol defined service change methods. The scenarios takes into consideration both ROOT termination and specific terminations. The intention of the section is to provide the usage of different methods and the situations when each of the methods needs to be used.

3.1 ROOT Termination.
3.1.1 Registration.

The MEGACO protocol defined a special termination called the ROOT termination to address the gateway as a whole. This enables easy addressing from both MG and MGC to address the gateway as a single entity. The virtual MG situations are not discussed in this version of the call flow draft.

The Media Gateway once ready to receive messages from MGC registers itself using the Service change command. The termination Identifier ROOT is used for this purpose. This can be treated as the first
messages sent from MG to MGC. The call flow assumes that the MGC is ready to receive the messages from the MG.

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>PortNum 20000</td>
<td>Port Num 2944</td>
</tr>
<tr>
<td>ServiceChange</td>
<td>ServiceChange Resp</td>
</tr>
<tr>
<td>ROOT Method=Restart Sca=15000</td>
<td>ROOT sca=25000</td>
</tr>
<tr>
<td>PortNum 15000</td>
<td>PortNum 25000</td>
</tr>
<tr>
<td>Modify *Event al/of</td>
<td>Modify Response</td>
</tr>
<tr>
<td>PortNum 15000</td>
<td>PortNum 25000</td>
</tr>
<tr>
<td>PortNum 20000</td>
<td>Port Num 2944</td>
</tr>
<tr>
<td>ServiceChange</td>
<td>ServiceChange Resp</td>
</tr>
<tr>
<td>ROOT Method=Restart Sca=15000</td>
<td>ROOT sca=25000</td>
</tr>
</tbody>
</table>

In the following example MG generates the registration message to the default port of MGC, it also specifies the service change address to which the MGC has to send further messages. MGC also specifies the new transport address so that when MG generates any messages it generates to this address instead of the default transport address for the MGC. In the example it is to be noted that the MGC is generating the MODIFY command to the new transport address specified by MG in its registration command.

The first command sent from MG (once it is ready to receive message) is the Service Change command, with ROOT termination identifier. In the example MG is generating the registration message to the default port of MGC (defined by protocol 2944 for Text message)

Step 1:
MG to MGC:

```
MEGACO/1 [209.110.59.34]: 20000
Transaction = 1234 {
    Context = - {
        ServiceChange = ROOT {Services {
            Method=Restart,
            ServiceChangeAddress=15000, Profile=ResGW/1,
            12345677T87654320
        }
    }
}
```

Madhubabu, et al. [Page 128]
The MGC responds the registration request to the same transport address used by MG to send the request. In the response MGC specifies its new transport address, that needs to be used by MG for further messages generated towards MGC (in example 25000).

**Step 2:**
MGC to MG:

```
MEGACO/1 [216.33.33.61]:2944
Reply = 1234 {
  Context = - {
    ServiceChange = ROOT {
      Services {ServiceChangeAddress=25000, Profile=ResGW/1 ,
        12345678T87654321} }
  }
}
```

In the example it is also shown that the MGC uses the transport address specified by MG for receiving messages. The initial MODIFY command sent to check for offhook event on all termination is sent to the new transport address specified by MG.

**Step 3:**
MGC to MG:

```
MEGACO/1 [216.33.33.61]:25000
Transaction = 9999 {
  Context = - {
    Modify = * {
      Events = 1234 {al/of}
    }
  }
}
```

The MG generates response to the new transport address specified by MGC in its registration response.

**Step 4:**
MG to MGC:

```
MEGACO/1 [209.110.59.34]:15000
Reply = 9999 {
  Context = - {
    Modify = A1
    Context = - {
      Modify = A2
      Context = - {
        Modify = A3
      }
    }
  }
}
```

### 3.1.2 Cold Start

A MG is pre-provisioned by a management mechanism with a Primary and (optionally) an ordered list of Secondary MGC’s. Upon a cold start of the MG, it will issue a ServiceChange command with a "Restart" method, on the Root Termination to its primary MGC. If the MGC accepts the MG, it will send a Transaction Accept, with the ServiceChangeMgcId set to it. If the MG receives an ServiceChangeMgcId not equal to the MGC
it contacted, it sends a ServiceChange to the MGC specified in the ServiceChangeMgcId. In this example the MG generate a registration command towards the secondary MGC. The secondary MGC responds with the service change address to enable MG to send further messages to that specified port.

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Primary MGC</th>
<th>Secondary MGC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServiceChange ROOT Method = Restart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServiceChange Resp ROOT ServiceChangeMGCId=209.110.59.33:25000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServiceChange ROOT Method=Restart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServiceChange ROOT Resp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first command sent from MG (once it is ready to receive message) is the Service Change command, with ROOT termination identifier. In the example MG is generating the registration message to the Primary MGC.

Step 1:
MG to MGC:
```
MEGACO/1 [209.110.59.34]: 20000
Transaction = 1234 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Restart,
    }
  }
}
```

The MGC responds the registration request by specifying a new transport address of the secondary MGC that needs to be used by MG for generating another registration command towards secondary MGC.

Step 2:
MGC to MG:
```
MEGACO/1 [216.33.33.61]:25000
Reply = 1234 {
  
Madhubabu, et al.        [Page 130]
MG generates the Service Change command to the MGC specified by the primary MG.

Step 3:
MG to MGC:

MEGACO/1 [209.110.59.34]: 25000
Transaction = 4321 {
    Context = - {
        ServiceChange = ROOT {Services {
            Method=Restart, Reason="Cold Boot",
            Profile=ResGW/1, 12345677787654320}
        }
    }
}

The MGC after receiving the registration request from the MG responds with the Service Change reply.

Step 4:
MGC to MG:

MEGACO/1 [216.33.33.62]: 27000
Transaction = 4321 {
    Context = - {
        ServiceChange = ROOT {Services {
            Profile=ResGW/1, 12345677787654320}
        }
    }
}

3.1.3 Handoff

The Handoff method can be used both by MG and MGC. In the scenario below MGC uses this method to indicate that it is going out of service and the MGC specified in the service change need to be contacted. The MG subsequently generates the handoff message to the MGC specified in the Service change mgId in the message generated by the controlling MGC.
The MGC generates the ServiceChange command with Handoff as the method. It also specifies the MGC that needs to be tried by the MG.

Step 1:
MGC to MG:

```
MEGACO/1 [209.110.59.34]: 20000
Transaction = 1234 {
    Context = - {
        ServiceChange = ROOT {Services {
            Method=Handoff,
            MgcIdToTry=199.164.0.197:45678, Profile=ResGW/1, 12345677T87654320}
        }
    }
}
```

After receiving the command from MGC, the MG tries for the MG that is specified in the message. It first responds to the service Change command generated by the controlling MGC.

Step 2:
MG to MGC:

```
MEGACO/1 [216.33.33.61]: 25000
Reply = 1234 {
    Context = - {
        ServiceChange = ROOT
    }
}
```
MG generates the Service Change command to the MGC specified by the primary MG.

Step 3:
MG to MGC:

MEGACO/1 [209.11.59.34]: 25000
Transaction = 4321 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Handoff, Reason="MGC Directed Change",
      Profile=ResGW/1, 12345677T87654320}
    }
  }
}

The MGC after receiving the Handoff request from the MG responds with the Service Change reply.

Step 4:
MGC to MG:

MEGACO/1 [216.33.33.62]: 27000
Reply = 4321 {
  Context = - {
    ServiceChange = ROOT {Services {
      Profile=ResGW/1, 12345677T87654320}
    }
  }
}

3.1.4 Disconnection
The MG issues the disconnection method of Service Change when it lost the communication with MGC and later regains it. The MGC normally generates an Audit message to assess the state of the terminations before and after the disconnection. The Audit command enables MGC to synchronize its state with the MG’s state.

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MG lost communication with MGC</td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al.  [Page 133]
The MG generates ServiceChange command towards the MGC with method = disconnection on the ROOT termination and with reason = "Loss of lower layer connectivity".

Step 1:
MG to MGC:
    MEGACO/1 [209.110.59.34]: 20000
    Transaction = 1234 {
        Context = - {
            ServiceChange = ROOT {Services {
                Method=Disconnection,
                Reason="Loss of lower layer connectivity"
            }
        }
    }
}

The MGC responds the disconnection message by responding with Service Change response.

Step 2:
MGC to MG:
    MEGACO/1 [216.33.33.61]: 25000
    Reply = 1234 {
        Context = - {ServiceChange = ROOT {
            }
        }
    }
}

The MGC after receiving the disconnection message audits the terminations that are present in the Media Gateway. The Audit response enables MGC to assess the state of the termination before and after disconnection.

Step 3
MGC to MG:
MEGACO/1 [216.33.33.61]:27000
Transaction = 1235 {
    Context = 2 {AuditValue = *{
        Audit{Media, DigitMap, Events, Signals, Packages, Statistics
    } }
}
}

The MG responds with the media descriptor, packages and statistics. The Digit map, signal and events are not defined on this termination hence only tokens are sent back in the response.

Step 4
MG to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1235 {
    Context = 2 {
        AuditValue = EphA {
            Media {
                TerminationState { ServiceState = InService,
                                Buffer = OFF },
                Stream = 1 {
                    LocalControl { Mode = SendReceive,
                                    nt/jit=40 },
                    Local {
                        v=0
                        c=IN IP4 209.110.59.33
                        m=audio 30000 RTP/AVP 0
                        a=ptime:30
                    },
                    Remote {
                        v=0
                        c=IN IP4 207.176.47.90
                        m=audio 40000 RTP/AVP 0
                        a=ptime:30
                    } },
                audit { Events,
                          Signals,
                          DigitMap},
                Packages {nt-1, rtp-1},
                Statistics { rtp/ps=1200, ; packets sent
                              nt/os=62300, ; octets sent
                              rtp/pr=700, ; packets received
                              nt/or=45100, ; octets received
                              rtp/pl=0.2, ; % packet loss
                              rtp/jit=20,
                              rtp/delay=40 } ; avg latency
            }
    }
}

Madhubabu, et al.                                             [Page 135]
3.2 Service Change for Termination

The service change command can be used on terminations to take terminations from in-service to out-of-service and vice versa. This section illustrates the use of service change on termination generated both from MG and MGC. The wildcard termination identifiers scenarios are discussed in a separate section.

3.2.1 MG generated Service Change

3.2.1.1 Graceful OOS from MG

The graceful method of service change when generated from MG is used to intimate MGC the removal of terminations from in-service. The Service Change delay is used to specify the period after which the termination will be removed from service. The service change delay can be NULL or any other 32bit-value. We will consider both the cases here. Case a deals the simple case where the delay is 350 seconds and in case b we will consider NULL value for the delay.

case (a)

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>ServiceChange CtxId=2 TermA</td>
<td></td>
</tr>
<tr>
<td>Method=Graceful, Delay = 350</td>
<td></td>
</tr>
<tr>
<td>&lt;------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ServiceChange Resp</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>After 350 Seconds TermA is taken out of service</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Subtract Ctxid=2 TermA</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Subtract Resp Ctxid=2 TermA</td>
<td></td>
</tr>
</tbody>
</table>
Step 1
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
   Transaction = 4321 {
      Context = 123 {
         ServiceChange = TermA {Services {
            Method=Graceful, Reason="Termination taken out of Service", Delay = 350 }
         }
      }
   }

In the first step MG generates the service change command with Graceful method. The termination "TermA" is in context 123. The delay 350 indicates MGC that the termination will be moved to out-of-service after 350 seconds.

Step 2
MGC to MG:
   MEGACO/1 [207.176.47.89]: 27000
      Reply = 4321 {
         Context = 123 {
            ServiceChange = TermA {
            }
         }
      }

After 350 seconds the MG removes the termination "TermA" out of service. As the responsibility of clearing the context lies with the MGC, it generates the Subtract command to remove the termination out of context. The audit descriptor in this example is shown to be present and shown empty. This indicates that the MGC is not interested in any statistics to be returned in the response from MG.

Step 3
MGC to MG:
   MEGACO/1 [207.176.47.89]: 27000
      Transaction = 1234 {
         Context = 123 {
            Subtract = TermA {
            Audit = {}
            }
         }
      }

The MG responds with the Subtract response. By the time the subtract is received by MG, the termination is out-of-service. The Subtract command can be received even for terminations that are out-of-service. After subtracting the termination from the context, the dynamic information for the context pertained to that termination is cleared.
Step 4
MG to MGC:
MEGACO/1 [987.654,321.1]: 25000
   Reply = 1234 {
      Context = 123 {
         Subtract = TermA {
      }
   }
}

case (b)

In this subsection we consider the case where the delay specified by MG is NULL. This indicates that the termination will be taken out-of-service after the termination is subtracted from valid context.

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceChange CtxId=2 TermA</td>
<td></td>
</tr>
<tr>
<td>Method=Graceful, Delay = NULL</td>
<td></td>
</tr>
<tr>
<td>ServiceChange Resp</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>Subtract Ctxid=2 TermA</td>
<td></td>
</tr>
<tr>
<td>Subtract Resp Ctxid=2 TermA</td>
<td></td>
</tr>
<tr>
<td>TermA Taken Out-of-service</td>
<td></td>
</tr>
</tbody>
</table>

Step 1
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
   Transaction = 4321 {
      Context = 123 {
   }

Madhubabu, et al.
ServiceChange = TermA {Services {
    Method=Graceful, Reason="Termination taken out of Service", Delay = 0
} }

MGC after receiving gets that information that the termination will not be available after the call. MGC should inhibit using this termination further in its calls. MGC responds to the MG for the command it received.

Step 2
MGC to MG:
    MEGACO/1 [207.176.47.89]: 27000
    Reply = 4321 {
        Context = 123 {
            ServiceChange = TermA {
            }
        }
    }

The MGC after the call is complete subtracts the termination from the context and avoids using this termination in further calls. Even in this example the audit descriptor is empty.

Step 3
MGC to MG:
    MEGACO/1 [207.176.47.89]: 27000
    Transaction = 1234 {
        Context = 123 {
            Subtract = TermA {
                Audit = {}  
            }
        }
    }

MG after receiving the message from MGC, responds to the message and immediately removes the termination to out-of-service.

Step 4
MG to MGC:
    MEGACO/1 [987.654,321.1]: 25000
    Reply = 1234 {
        Context = 123 {
            Subtract = TermA {
            }
        }
    }  

3.2.1.2 Forced OOS from MG
The Forced method of service change when generated from MG is used to intimate MGC the removal of terminations from in-service. The termination may be in a valid Context or NULL context when MG has generated this message. We will consider both the cases here. Case (b) deals the simple case where the termination is in NULL context and in case (a) the termination is in Valid Context.

case (a)

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>ServiceChange CtxId=123 TermA</td>
</tr>
<tr>
<td></td>
<td>Method = Forced</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>ServiceChange Resp</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>Immediately TermA is taken out of service</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>Subtract Ctxid=123 TermA</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>Subtract Resp Ctxid=123 TermA</td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
</tbody>
</table>

Step 1
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
  Transaction = 4321 {
    Context = 123 {
      ServiceChange = TermA {Services {
        Method=Forced }
      }
    }
  }
}

In the first step MG generates the service change command with "Forced" Service Change method. The termination "TermA" is in context 123.

Step 2
MGC to MG:
Madhubabu, et al.
After generating the message MG immediately removes the termination "TermA" out of service. As the responsibility of clearing the context lies with the MGC, it generates the Subtract command to remove the termination out of context. The audit descriptor in this example is shown to be present and shown empty. This indicates that the MGC is not interested in any statistics to be returned in the response from MG.

Step 3
MGC to MG:

MEGACO/1 [207.176.47.89]: 27000
Transaction = 1234 {
    Context = 123 {
        Subtract = TermA {
            Audit = {}
        }
    }
}

The MG responds with the Subtract response. When the subtract message is received by MG, the termination is out-of-service. The Subtract command can be received even for terminations that are out-of-service. After subtracting the termination from the context, the Call specific information for the context pertained to that termination is cleared.

Step 4
MG to MGC:

MEGACO/1 [987.654,321.1]: 25000
Reply = 1234 {
    Context = 123 {
        Subtract = TermA {
        }
    }
}

case (b)

In this subsection we consider the case where the termination is in NULL context. This indicates that the termination will be immediately taken out-of-service.
### Media Gateway to Media Gateway Controller

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>ServiceChange CtxId=NULL TermA</td>
</tr>
<tr>
<td></td>
<td>Method = Forced</td>
</tr>
<tr>
<td></td>
<td>&lt;- ServiceChange Resp</td>
</tr>
<tr>
<td></td>
<td>Immediately TermA is taken out of service</td>
</tr>
</tbody>
</table>

#### Step 1
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
Transaction = 4321 {
  Context = - {
    ServiceChange = TermA {Services {
      Method=Forced }
    }
  }
}

MGC after receiving gets that information should inhibit using this termination further in its calls. MGC responds to the MG for the command it received.

#### Step 2
MGC to MG:
MEGACO/1 [207.176.47.89]: 27000
Reply = 4321 {
  Context = - {
    ServiceChange = TermA {
    }
  }
}

#### 3.2.1.3 Restart INS from MG
In this we will consider the message generation from MG when terminations are brought back to in-service state after they have been removed from out-of-service. Initially when the MG comes up, it doesn’t generate any in-service message for specific terminations, as the service change generated on ROOT terminations serves the same purpose.
MG can generate restart for multiple terminations using the wildcard mechanism. But for simplicity we are considering the case where the termination "TermA" is brought back to service.

<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>ServiceChange CtxId=NULL TermA</td>
</tr>
<tr>
<td></td>
<td>Method = Restart</td>
</tr>
<tr>
<td></td>
<td>ServiceChange Resp</td>
</tr>
<tr>
<td></td>
<td>Immediately TermA is brought back to service</td>
</tr>
</tbody>
</table>

Step 1
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
Transaction = 4321 {
  Context = - {
    ServiceChange = TermA {Services {
      Method=Restart}
    }
  }
}

MGC after receiving the message updates its Database for using this Termination for future use . MGC responds to the MG for the command it received.

Step 2
MGC to MG:
MEGACO/1 [207.176.47.89]: 27000
Reply = 4321 {
  Context = - {
    ServiceChange = TermA {
  }
}
}

3.2.2 MGC generated Service Change
In the earlier section we saw few scenarios where the MG generated
messages to MG indicating the change of state in the termination. In this section we will look into the cases where MGC generates the service change message on terminations to remove the terminations from in-service to out-of-service and vice-versa.

3.2.2.1 Forced OOS from MGC

When the MGC intends to remove the termination from in-service to out-of-service, it generates a forced service change message towards MG. MG immediately removes the termination from in-service to out-of-service. If the termination is in valid context, MGC can generate subtract command and following that another message with service change command to remove it from service.

```
<table>
<thead>
<tr>
<th>Media Gateway</th>
<th>Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>ServiceChange CtxId=NULL TermA</td>
<td></td>
</tr>
<tr>
<td>Method = Forced</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>ServiceChange Resp</td>
<td></td>
</tr>
<tr>
<td>Immediately TermA is taken out of service</td>
<td></td>
</tr>
</tbody>
</table>
```

Step 1
MGC to MG:
```
MEGACO/1 [207.176.47.89]: 27000
Transaction = 4321 {
  Context = - {
    ServiceChange = TermA {
      Method=Forced
    }
  }
}
```

MG after receiving the message from MG removes the termination immediately from service. The response is generated towards for the message received from MGC.

Step 2
MG to MGC:
```
MEGACO/1 [209.11.33.62]: 25000
Reply = 4321 {
  Context = - {
    
  }
}
```
3.2.2.2 Restart from MGC
When the MGC intends to bring the termination from out-of-service to in-service, it generates a restart method of service change message towards MG. MG immediately brings the termination from out-of-service to in-service.

Step 1
MGC to MG:
MEGACO/1 [207.176.47.89]: 27000
Transaction = 4321 {
  Context = - {
    ServiceChange = TermA {
      Method=Restart
    }
  }
}

MG after receiving the message from MG brings the termination immediately to service. The response is generated towards for the message received from MGC.

Step 2
MG to MGC:
MEGACO/1 [209.11.33.62]: 25000
Reply = 4321 {
  Context = - {
    ServiceChange = TermA {
      Method=Restart
    }
  }
}
4.0 Audit Command Usage
The Audit command is defined in the protocol to enable MGC to get information from MG to MGC. The information received may contain the values present at the MG on a given termination or possible values that can be supported on the given termination depending upon whether the command is Audit Value or Audit Capability. The Audit command can be used on Root Termination also. The Audit command is used by MGC when it needs the state of the termination and the parameter values for that given termination. Wildcard termination identifier and wildcard contextID can be used by MGC in these Audit command.

4.1 Audit Value
The Audit value command can be sent from MGC to know the present values of the descriptors requested. The Audit Value command can be sent on either ROOT termination or any other termination in the MG. The section 2.1.1 illustrates the usage of Audit Value command on ROOT termination and section 2.1.2 illustrates the usage of Audit value command on a given termination.

4.1.1 Audit value command on ROOT Termination
The ROOT termination denotes the entire gateway as a single entity. There are two possible cases of Audit Value command on ROOT termination. The one in which the ContextId is NULL and the other in which the contextID is ALL. The case (a) below illustrate when the Audit Value command when the ContextId is * and in case (b) other in which the ContextId is NULL. The base ROOT package is assumed to be implemented on the ROOT termination.

Case (a)
In this section we will illustrate how MGC retrieves the lists of all contexts in MG. The Audit Value command with ContextId ‘*’ and termination identifier ROOT enables MGC to get all the valid contextID values.

Step 1
MGC to MG:
    MEGACO/1 [207.176.47.89]: 27000
    Transaction = 1234 {
        Context = * {
            AuditValue= ROOT {Audit { } }
        }
    }

The MG after receiving the command from MGC constructs the response with all the contextID that is active in that MG.

Step 2

Madhubabu, et al.                                           [Page 146]
Case (b)

This section illustrates the usage of ROOT termination identifier with NULL ContextID. This command is supposed to return the values of properties and statistics implemented on ROOT. The MGC generates the AuditValue message to MGC.

Step 1
MG to MGC:

```
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
Reply = 1234 {
    Context = 100 {
        AuditValue = TermA, TermB
    }
    Context = 200 {
        AuditValue = TermC, TermD
    }
    Context = 300 {
        AuditValue = TermE, TermF
    }
}
```

The MG responds with the values of properties that are defined on the ROOT termination. These properties include the Maximum number of contexts, Maximum number of terminations per context, normal MG execution time, normal MGC execution time and Provisional response timer value. There are no statistics that are defined on the ROOT termination. Hence the MG responds with the statistics token only.

Step 2
MG to MGC:

```
MG to MGC:
MEGACO/1 [216.33.33.61]: 25000
Reply = 1234 {
    Context = - {
        AuditValue = ROOT {Audit {Media, Statistics} }
    }
}
```

The MG responds with the values of properties that are defined on the ROOT termination. These properties include the Maximum number of contexts, Maximum number of terminations per context, normal MG execution time, normal MGC execution time and Provisional response timer value.
4.1.2 Audit value on non-ROOT terminations

This section illustrates the usage of AuditValue command on terminations other than ROOT. The Audit value command response constitutes all the active value of the descriptors that are requested by MGC. In this example we assume that the descriptors are already updated with commands from MGC. The MGC audit is done when the termination is in valid context. This is to include the statistics descriptor in the response from MG. The termination is assumed to be Ephemeral to use all the statistics defined in network package and RTP package. The second message illustrates the use of AuditValue to get the active descriptor values defined on physical terminations.

Step 1
MGC to MG:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
  Context = 2 {AuditValue = EphA{
    Audit{Media, DigitMap, Events, Signals, Packages, Statistics
  }}
}
}

The MG responds with the media descriptor, packages and statistics. The Digit map, signal and events are not defined on this ephemeral termination hence only tokens are sent back in the response.

Step 2
MG to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1234 {
  Context = 2 {
    AuditValue = EphA {
      Media {
        TerminationState { ServiceState = InService,
          Buffer = OFF },
        Stream = 1 {
          LocalControl { Mode = SendReceive,
            nt/jit=40 },
          Local {
            v=0
          } c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 0
          a=ptime:30
          },
        Remote {

v=0
c=IN IP4 207.176.47.90
m=audio 40000 RTP/AVP 0
a=rtpmap:0 0/8000
a=ptime:30
}
}
audit { Events,
Signals,
DigitMap},
Packages {nt-1, rtp-1},
Statistics { rtp/ps=1200, ; packets sent
nt/os=62300, ; octets sent
rtp/pr=700, ; packets received
nt/or=45100, ; octets received
rtp/pl=0.2, ; % packet loss
rtp/jit=20,
rtp/delay=40 } ; avg latency
}
}

The MGC in the following command generates audit value command for
physical termination that is in a context. Here only DigitMap, event
and signal descriptor are audited.

Step 3
MGC to MG:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1235 {
  Context = 2 {AuditValue = TermA{
    Audit{ DigitMap, Events, Signals
  }}
}
}

The MG after receiving the command responds with the event, signal
and digit map descriptors that are active on the termination.

Step 4
MG to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1235 {
  Context = 2 {
    Modify = TermA{
      Events = 2223 {
        al/on, dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
        (0| 00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxx|9011x.)}
    }
  }
}
4.2 Audit Capability

The Audit capability command from MGC retrieves the possible values of the descriptors requested. The first section 5.2.1 illustrates the Audit capability command on ROOT termination and 5.2.2 on terminations other than ROOT termination.

4.2.1 Audit Capability on ROOT termination

The Audit Capability command on ROOT termination is used to determine the properties that are implemented on the ROOT termination. The response from MG includes the possible values of the properties.

Step 1:
MGC to MG:

```
MEGACO/1 [216.33.33.61]:27000
Transaction = 1235 {
  Context = - {Audit Capability = ROOT{
    Audit{ Media }}
  }
}
```

The properties that are defined on the ROOT in this example are the Max number of Contexts, maximum terminations per context, normal MG execution time, normal MGC execution time and provisional response timer value.

Step 2
MG to MGC:

```
MEGACO/1 [209.110.59.34]:25000
Reply = 1235 {
  Context = - {
    Modify = ROOT{
      Media { TerminationState { MaxNumberOfContexts =10,
        MaxTerminationsPerContext =2, NormalMGExecutionTime =250,
        NormalMGCExecutionTime =250,
        ProvisionalResponseTimerValue = 200 }}
    }
  }
}
```

4.2.2 Audit Capability on non-Root Terminations.

This section illustrates the usage of Audit Capability command on a non-ROOT termination. The MGC in this example is generating Audit Capability for Events descriptor and signal descriptor. The MG should respond with the possible values of these descriptors.

Step 1:

```
Madhubabu, et al.                                             [Page 150]
```
The response has all the possible values. In this example for simplicity event parameters are not included. The packages that are realized on TermA are analog line supervision package, DTMF detection package call progress tone generation package, tone generation package, and generic package.

5. IVR using MEGACO

The Interactive Voice Response (IVR) is assumed to be a gateway with only ephemeral terminations. The IVR is capable of playing announcements, detecting digits and reporting the same to the MGC. In this example we assume a IVR, which is controlled by a MGC. The IVR is assumed to play some announcement and detects the DTMF digits and reports the same to MGC. This section presents few call scenarios where a residential user is connected to IVR, Trunking gateway connected to IVR, call disconnection from residential and Trunking gateway to the IVR.

5.1 Connecting Residential gateway to IVR.
Internet-Draft  Megaco/H.248 Call flow examples  July 2001

-------->|
  Modify to  
  check offhook
<--------|<--------|<--------|
  Modify Resp  Modify Resp

--------->
  UserA offhook
  Notify offhook
<--------|
  Notify Resp
<--------|
  Modify SG:dialtone
  ED:al/on,dd/ce{Dmap1}
  DM:Dmap1 = 2XXX

<--------|
  Dial Tone
  Modify Resp

--------->
  User Dials Digits
  Notify digits
<--------|
  Notify Response
<--------|
  Add TermA SD:ringbacktone
  Add $, Local SDP Info -underspecified

<--------|
  RingBack Tone
  Modify Resp TermA
  Add Resp Local SDP (Specified)
  <--------|
  Add $ Local(Underspecified)
  Remote SDP (Specified)
  <--------|
  Add Resp EphB Local Specified

---------->
  Modify TermA SendRecv
  Modify EphA Remote(Specified) SendRecv
<-------->
  Modify Resp

/------------------------
  RTP MEDIA
\------------------------/
The MGC initially generates a Modify command to the Residential gateway to which UserA is connected. The Modify command lists a offhook event in the Events descriptor. The embedded signal descriptor lists the dial tone and the embedded event descriptor lists the Digit completion event of the DTMF package.

Step 1
MGC to RGW:
    MEGACO/1 [216.33.33.61]:27000
    Transaction = 1234 {
        Context = - {
            Modify = TermA {
                Media {
                    LocalControl {
                        Mode = ReceiveOnly}
                } ,
                Events = 1111 {al/of { Signals {cg/dt},Embed = 1112 {al/on, dd/ce
                        (DigitMap=Dmap1)} }
                DigitMap= Dmap1{(2XXX)}
            }
        }
    }

The residential gateway responds the Modify command.

Step 2:
MG1 to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Reply = 1234 {
        Context = - (Modify = TermA)
    }

In this example User A goes off hook. This event is detected by the RGW and constructs the Notify message to the MGC. The MG uses the same request id (1111) sent by the MGC in its initial command. The timestamp of the event detected is also passed as parameter to the observed event.

Step 3
MG1 to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Transaction = 2000 {
        Context = - {
            Notify = TermA {ObservedEvents =1111 {

Madhubabu, et al.                                            [Page 153]
MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 4
MGC to MG1:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2000 {
      Context = - {Notify = TermA}
   }

The users dials digits and that takes the call to be terminated on the IVR. The digits dialed by the user are reported to the MGC in the Notify command.

Step 6
MG1 to MGC:
   MEGACO/1 [209.110.59.34]: 25000
   Transaction = 2001 {
      Context = - {
         Notify = TermA {ObservedEvents =1112 {
            20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
      }
   }

MGC after receiving the Notify command responds back with the Notify response.

Step 7
MGC to MG1:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 2001 {
      Context = - {Notify = TermA}
   }

MGC after receiving the Notify command starts analyzing the dialed digits. In this example the called subscriber is connected to the RGW2, which is again controlled by the same MGC. The MGC generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. As the MGC is aware that the destination user UserB is free it indicates MG1 to apply ringback tone to the termination of UserA. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created because of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used.

Step 8
In this example the connection fields IP address, the media field port number are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating the creation of a new context. In this example the MG creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. MG responds with the allocated IP address 209.110.59.33 and port number 30000.

Step 9

MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
    Context = 1 {
        Add = TermA,
        Add=EphA{
            Media {
                Local {
                    v=0
                    c=IN IP4 209.110.59.33
                    m=audio 30000 RTP/AVP 4
                        a=recvonly
                }
            }
        }
    }
}
MGC generates a similar transaction towards the IVR media gateway. The ContextID specified in the action is $. The Add command is meant for create an ephemeral termination. MGC has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the IVR. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 10
MGC to IVR:

```
MEGACO/1 [216.33.33.61]:27000
Transaction = 1236 {
  Context = $ {
    Add = $ {
      Media {
        LocalControl {
          Mode = Receiveonly,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        },
        Remote {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
        } ; RTP profile for G.723 is 4
      }
    Events = 1113{ streamid = 1 dd/ce { dmap2 }}
    Digits = Dmap2 {2XXX}
  }
}
```

IVR after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 2. The IVR creates a ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example IVR reserves IP address 207.176.47.90 and port number 40000. The IVR responds to MGC with the following transaction reply.

Step 11
MG2 to MGC:

```
```

Madhubabu, et al. [Page 156]
The MGC after receiving the response forwards the remote SDP information in Modify command to the Residential gateway. The MGC generates message to the RGW to stop the ringback tone and changes the mode of the two terminations TermA and EphA to send receive.

Step 12
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1237 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
        Signals { }
      },
      Modify = EphA {
        Media {
          LocalControl {
            Mode = sendrecv
          } Remote {
            v=0
c=IN IP4 207.176.47.90
m=audio 40000 RTP/AVP 4
          }
        }
      }
    }
  }
}

The empty signal descriptor in the Modify command for termination TermA, stop the ringback tone at the calling end. The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.
Step 13

MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000

Reply = 1237 {
  Context = 1 {Modify = TermA, Modify = EphA}
}

Now the RTP flow is established. The IVR plays an announcement. The User dials digits depending upon the announcement. The digits are detected on the RTP stream.

5.2 Disconnecting Residential User from IVR.

This section illustrates the case of disconnecting a residential user from IVR. The assumption is that the RTP media is already established and now the MGC has to act upon the user actions. The MGC waits for the user to go onhook. Once the UserB goes onhook, MG2 reports the notification of the onhook event to the MGC.

---

<table>
<thead>
<tr>
<th>USERA</th>
<th>RGW1</th>
<th>MGC</th>
<th>IVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERA goes OnHook</td>
<td>Notify OnHook</td>
<td>Notify Resp</td>
<td>Subtract TermA</td>
</tr>
<tr>
<td></td>
<td>Subtract EphA</td>
<td>Subtract Resp TermA</td>
<td>Subtract Resp EphA Statistics</td>
</tr>
<tr>
<td></td>
<td>Subtract EphB</td>
<td>Subtract Resp EphB Statistics</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1

MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Internet-Draft      Megaco/H.248 Call flow examples       July 2001

Transaction = 3000 {
  Context = 1 {
    Notify = TermA {ObservedEvents =1234 {
      20000202T10020000:al/on}}
  }
}
The MGC responds to the MG2 with the Notify response.

Step 2
MGC to MG2:
  MEGACO/1 [216.33.33.61]: 27000
  Reply = 3000 {
    Context = 1 {Notify = TermA}
  }
The MGC generates transactions with two subtracts commands one for
  physical and other for ephemeral terminations.

Step 3
MGC to MG1
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1234 {
    Context = 1 {
      Subtract = TermA {Audit{}},
      Subtract = EphA {Audit{Statistics}}
    }
  }
The MG subtracts the two terminations from the context. The context
  itself is deleted with the subtract of the last termination from it.
The MG1 responds to this transaction from MGC with statistics on
  ephemeral termination.

Step 4
MG1 to MGC:
  MEGACO/1 [209.110.59.34]:25000
  Reply = 1234 {
    Context = 1 {
      Subtract = TermA
      Subtract = EphA {
        Statistics {
          rtp/ps=1234, ; packets sent
          nt/os=56789, ; octets sent
          rtp/pr=987, ; packets received
          nt/or=65432, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }
The MGC generates similar command towards the IVR to subtract the ephemeral termination.

Step 5
MGC to MG2:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = 2 {
    Subtract = EphB {Audit(Statistics)}
  }
}
The IVR responds to the subtract commands generated by MGC.

Step 6
MG2 to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1235 {
  Context = 2 {
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

5.3 Connecting Trunking Gateway to IVR.
This section illustrates a call initiated from Trunking gateway towards IVR. It is assumption that the same MGC controls both the IVR and the Trunking gateway.

<table>
<thead>
<tr>
<th>SS7 Switch</th>
<th>TGW</th>
<th>MGC</th>
<th>IVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The MGC receives IAM message from the SS7 switch. In this example we assume that the Signaling gateway and the Media Gateway are together in one physical box. The MGC responds with the ACM message. The MGC also generates add command to the Trunking gateway for addition of a circuit group of specific trunk and also another Add for ephemeral termination. For the ephemeral termination the MGC specifies few SDP parameters in the Local descriptor and many of the parameters are underspecified. This facilitates the MG to assign values by its own.

Step 1
MGC to TGW

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = $ {
    Add = Trunk1/line1 {Media {
      LocalControl {Mode = SendRecv}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
    }
  }
```

Madhubabu, et al.
The Trunking gateway after responds with a contextID in this example. The ephemeral termination is created and added to the same context as the physical termination. The ephemeral termination added in this example is EPHA. The local parameters are specified in the response. The IP address chosen for the media transport in this example is 209.110.59.33, and the port number specified 30000.

Step 2
TGW to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1234 {
  Context = 1 {
    Add = Trunk1/line1,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
        }
      }
    }
  }
}

The MGC after receiving the response from the Trunking gateway uses the SDP information in the response sent from TG to the IVR. The command is for adding the ephemeral termination. The MGC requests creation of context and to the termination in the same.

Step 3
MGC to IVR

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = $ {
    Add = $ {Media {
      LocalControl {
        Mode = Receiveonly,
      },
      Local {
        v=0
        c=IN IP4 $
        m=audio $ RTP/AVP 4
      }
    }
  }
}
Remote{
  v=0
  c=IN IP4 209.110.59.33
  m=audio 30000 RTP/AVP 4
}

; RTP profile for G723 is 4

The IVR creates a context with ContextId 2. The ephemeral termination EPHB is created with the specified SDP information. The response from the IVR specifies the local SDP information.

Step 4
RGW to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1235 {
  Context = 2 {
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
        }
        } ; RTP profile for G723 is 4
      }
    }
  }
}

The MGC after receiving the response with the local SDP information conveys the same to the Trunking gateway as remote SDP information in the Modify command.

Step 5
MGC to TGW
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = 1 {
    Modify = EphA
    {Media {
      LocalControl {
        Mode = SendRecv,
      },
      Remote{
        v=0
        c=IN IP4 209.110.59.33
        m=audio 30000 RTP/AVP 4
      }
    }
  }
}

Madhubabu, et al.                                             [Page 163]
The Trunking gateway responds to the Modify command.

Step 6
TGW to MGC:

\[
\text{MEGACO/1 [207.176.47.89]: 26000}
\]
\[
\text{Reply = 1236} \{
\text{Context = 1} \{
\text{Modify = EphB}
\}
\}
\]

5.4 Disconnecting Trunking gateway from IVR

This section illustrates the disconnection of an IVR call from Trunking gateway. The Trunking gateway and the Signaling gateway are assumed to be present in the same physical box. The SS7 messages are received by the Signaling gateway and are forwarded to the MGC through the signaling gateway.
It is assumed that the RTP stream is already established. The REL message is received by the MGC through the Signaling gateway. The MGC initiates the terminating of the IVR call. The MGC initially generates a transaction with two subtracts towards the Trunking gateway. One subtract for removing the physical termination and other to remove the ephemeral termination.

Step 1
MGC to TGW:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234
Context = 2 {
    Subtract = Trunk2/line1{Audit{ }},
    Subtract = EphB {Audit{Statistics}}
}
```

The TGW responds to the subtract commands generated by MGC.

Step 2
TGW to MGC:

```
MEGACO/1 [209.110.59.34]:26000
Reply = 1234
Context = 2 {
    Subtract = Trunk2/line1
    Subtract = EphB {
        Statistics {
            rtp/ps=987, ; packets sent
            nt/os=65432, ; octets sent
            rtp/pr=1234, ; packets received
            nt/or=56789, ; octets received
            rtp/pl=10, ; % packets lost
            rtp/jit=30,
            rtp/delay=30 ; average latency
        }
    }
}
```

The MGC generates similar command towards the IVR.

Step 3
MGC to IVR:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235{
    Context = 1 {
        Subtract = EphA {Audit{Statistics}}
    }
}
```

The IVR responds to the subtract commands generated by MGC.

Step 4
IVR to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1235{
  Context = 1 {
    Subtract = EphA {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

Step 1:
MGC to MG:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234{
  Context = * {
    Subtract = *{Audit{Statistics}}
  }
}

The MG now subtracts all terminations in any of the contexts. There will be as many actions as the number of Contexts that are active in the MG. In this example we assume two contexts Context1 and Context2 with one two terminations in each of the context.

Step 2:
MG to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1234{
  Context = 1{
    Subtract = TermA {audit = {Statistics}}
  }
}

6. Wildcard ContextID usage

The protocol defines two types of wildcards. The CHOOSE wildcard and the ALL wildcard. The CHOOSE wildcard when used from MGC for the ContextID enables MG to create a new context. The ALL wildcard ContextID enables MGC to multiple contexts using a single Action. If the MGC needs to perform an operation common to all Contexts it can use the Wildcard ContextID for this purpose. For example if MGC needs to subtract all terminations irrespective of context they are in, it can use ContextId ‘*’ and termination ID ‘*’. This enables MG to perform the operation on all contexts that are active in the MG. The CHOOSE wildcard had already been in used in earlier call flows. This section shows a scenario where MGC uses * wildcard in ContextID.
Subtract = EphA
Statistics {
  rtp/ps=987, ; packets sent
  nt/os=65432, ; octets sent
  rtp/pr=1234, ; packets received
  nt/or=56789, ; octets received
  rtp/pl=10, ; % packets lost
  rtp/jit=30,
  rtp/delay=30 ; average latency
}

Context = 2 {
  Subtract = TermB { audit = {statistics }}
  Subtract = EphB {
    Statistics {
      rtp/ps=987, ; packets sent
      nt/os=65432, ; octets sent
      rtp/pr=1234, ; packets received
      nt/or=56789, ; octets received
      rtp/pl=10, ; % packets lost
      rtp/jit=30,
      rtp/delay=30 ; average latency
    }
  }
}

7. Wildcard TerminationId Usage
The wildcards when used for TerminationId can represent CHOOSE, ALL or partial choose. The partial choose enables MGC to specify part of TerminationId and leaving the remaining part of the TerminationId either * or $. The CHOOSE wildcard usage is illustrated in earlier examples. The Partial wildcard usage is illustrated in the Trunking gateway examples. The * wildcard example is treated in this section.

In this example the MGC generates a command with wildcard ALL "*" TerminationId, to enable MG to processes the command for all the terminations in the Gateway. The Media gateway is assumed to be a residential gateway. The events descriptor requests MG to check for offhook and apply dial tone when the offhook event occurs.

Step 1:
MGC to RGW:
  MEGACO/1 [216.33.33.61]:27000
  Transaction = 1234 {
    Context = - {
      Modify = *{
        Media {
          LocalControl {

Madhubabu, et al.}
The MG responds with specifying each of the TerminationId for which the command has been processed.

Step 2:
MG1 to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1234 {
  Context = -
  {Modify = TermA}
  {Modify = TermB}
  {Modify = TermC}
  {Modify = TermD}
  {Modify = TermE}
  {Modify = TermF}
  {Modify = TermG}
  {Modify = TermH}
  {Modify = TermI}
  {Modify = TermJ}
}

8. Supplementary services support
8.1 Call Transfer

<table>
<thead>
<tr>
<th>MG1</th>
<th>MGC</th>
<th>MG2</th>
<th>MG3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;----------------</td>
<td>Initial Modify</td>
<td>Initial Modify</td>
<td>&lt;----------------</td>
</tr>
<tr>
<td>Notify OffHook</td>
<td>Modify Response</td>
<td>Modify Resp</td>
<td>Modify Response</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Internet-Draft  Megaco/H.248 Call flow examples       July 2001

----------
Notify Response
----------
Modify ED:dd/ce(DigitMap=dmap1),al/on SD:cg/dt

<------
Modify Resp

Dial

Tone
<------>

Modify ED:dd/ce(DigitMap=dmap1),al/on SD:cg/dt

<-------
Notify Digits

<--------
Notify Response
<------------------->
Modify ringback

<--------
Modify Resp

Digit

<--------
<------>

Modify Resp

Ringback
<------>

Ringback

----------
Notify Resp

Add Phy

Add Eph Local Unspecified
<-------->

Add Phy

Add Eph Resp

Add Eph Resp Local specified
<--------

Add Phy

Add Eph Local unspecified

Remote Specified
<--------

Add Phy

Add Eph Resp Local specified
<--------

Modify Eph Remote Specified
<------------------->

Modify Resp

RTP MEDIA

----------------/

Notify Flash

<--------

Modify Resp

Routine Only

<--------

Notify Resp

DialTone

----------

Modify Resp

<--------

Notify Digits

<--------

Notify Resp

<--------

Modify Ring

<--------
The call Transfer feature in PSTN allows a user to transfer a call that he has received to another phone. This feature should be supported by MGC so that it is capable of generating required messages towards MG. In this example we assume that the MGC is capable of supporting Call Transfer. UserB, the Called party press flash hook and initiates call towards UserC. After UserC responds to the call, UserB goes onhook to connect UserA with UserC.

The MGC generates the Modify message towards all the three Residential gateways to check for off hook on the terminations. (A wildcard command may also be used in this scenario but for simplicity we consider only command to specific terminations). We are not considering the embedded signal and event descriptors here.

The MGC in NULL context generates the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified in this example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control descriptor.

**Step 1**

MGC to RGW1:

```
MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = Receiveonly
        },
        Events = 1111 {al/of}
      }
    }
  }
}
```

Madhubabu, et al.
MG, after receiving the command from MGC, accepts it and responds with the transaction reply. Here for only MG1 is shown to generate the response. In fact all the RGW generate the responses.

Step 2

MG1 to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1234 {
   Context = - (Modify = TermA)
}

In this example User A goes off hook. This event is detected by the RGW1 and it constructs the Notify message to the MGC. The MG uses the same request id (1111) sent by the MGC in its initial command. The timestamp of the event detected is also passed as a parameter to the observed event.

Step 3

MG1 to MGC:
MEGACO/1 [209.110.59.34]:25000
Transaction = 2000 {
   Context = - {
      Notify = TermA {ObservedEvents =1111 {
         20010202T10000000:al/of}}
   }
}

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 4

MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
   Context = - (Notify = TermA)
}

The MGC in the following command issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial patter is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 5

MGC to MG1:
MG validates the Modify command and responds to the MGC and then starts processing the descriptors listed.

Step 6
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
  Context = - {Modify = TermA}
}

The descriptors are processed in the order that is specified by the MGC. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the termination. The Digit map will be ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself contains a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. Its assumed that the digits dialed by the user, match with the digit map pattern. Lets assume that the user has dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is generated from MG1 to MGC. The MG again uses the same request identifier as specified by the MGC.

Step 7
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
  Context = - {
    Notify = TermA {ObservedEvents = 1112 {
      20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
  }
}
MGC after receiving the Notify command responds back with the Notify response.

Step 8  
MGC to MG1:  
MEGACO/1 [216.33.33.61]: 27000  
Reply = 2001 {  
  Context = - {Notify = TermA}  
}

MGC after receiving the Notify command starts analyzing the dialed digits. In this example the called subscriber is connected to the RGW2, which is again controlled by the same MGC. The MGC generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. As the MGC is aware that the destination user UserB is free it indicates MG1 to apply ringback tone to the termination of UserA. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created because of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used.

Step 9  
MGC to MG1:  
MEGACO/1 [216.33.33.61]: 27000  
Transaction = 1236 {  
  Context = $ {  
    Add = TermA {  
      Signals { cg/rt }  
    }  
    Add = $ {  
      Media {  
        LocalControl {  
          Mode = ReceiveOnly,  
        },  
        Local {  
          v=0  
          c=IN IP4 $  
          m=audio $ RTP/AVP 4  
        }  
      }  
    }  
  }
}

In this example the connection fields IP address, the media field port number are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating
the creation of a new context. In this example the MG creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. MG responds with the allocated IP address 209.110.59.33 and port number 30000.

Step 10
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        } ; RTP profile for G.723 is 4
      }
    }
  }
}

MGC generates a similar transaction towards the RGW2. The ContextID specified in the action is $. The first command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MGC has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the RGW2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MGC to MG2:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1237 {
  Context = $ {
    Add = TermB { Media {
        LocalControl {Mode = Receiveonly} },
        Signals {al/ri}
        Events=1234{al/of},
    }
  }
}
MG2 after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists "ring" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates an ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example MG2 reserves IP address 207.176.47.90 and port number 40000. The MG2 responds to MGC with the following transaction reply.

Step 12
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1237 {
  Context = 2 {
    Add = TermB,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
        ; RTP profile for G.723 is 4
      }
    }
  }
}

Madhubabu, et al.                                           [Page 176]
The MGC waits for the UserB to go offhook. Once the UserB goes offhook, MG2 reports the notification of the offhook event to the MGC.

Step 13
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Transaction = 3000 {
    Context = 2 {
        Notify = TermB {ObservedEvents =1234 {
            20000202T10020000:al/of}}
    }
}
```

The MGC responds to the MG2 with the Notify response.

Step 14
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 3000 {
    Context = 2 {Notify = TermB}
}
```

The MGC generates a transaction towards MG2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook and flashhook events.

Step 15:
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1238 {
    Context = 2 {
        Modify = TermB {
            Signals { } ; to turn off ringing
            Events = 1235 {al/on, al/fl { signals cg/dt,
                events dd/ce{dmap1}, al/on }},
            Media {
                LocalControl {
                    Mode = SendRecv,
                }
            }
        }
        Modify = EphB{
            Media {
                LocalControl {
                    Mode = SendRecv,
                }
            }
        }
    }
}
```
The MG2 responds to the request from MGC.

Step 16
MG2 to MGC:
    MEGACO/1 [207.176.47.89]: 26000
    Reply = 1238 {
        Context = 2 {Modify = TermB, Modify = EphB}
    }

The MGC generates message to the MG1 to stop the ringback tone and to report the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MGC to MG1:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1239 {
        Context = 1 {
            Modify = TermA {
                Media {
                    LocalControl {
                        Mode = sendrecv
                    }
                    Signals {
                    }
                }
            },
            Modify = EphA {
                Media {
                    LocalControl {
                        Mode = sendrecv
                    }
                    Remote {
                        v=0
c=IN IP4 207.176.47.90
m=audio 400000 RTP/AVP 4
                    }
                }
            }
        }
    }

The empty signal descriptor in the Modify command for termination TermA, stops the ringback tone at the calling end. The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.

Step 18
MG1 to MGC:
    MEGACO/1 [209.110.59.34]: 25000
The two users can exchange media, as the RTP streams are made bi-directional.

The UserB now presses flash to dial the UserC number.

The UserB flash event is reported to MGC using the Notify message.

Step 19
MG2 to MGC:

MEGACO/1 [209.110.59.34]:29000
Transaction = 3001 {
  Context = 2 {
    Notify = TermB {ObservedEvents = 1235 {
      20040202T10000000:al/fl}}
  }
}

MGC generates the Notify response.

Step 20
MGC to MG2:

MEGACO/1 [216.33.33.61]:27000
Reply = 3001 {
  Context = 2 {Notify = TermB}
}

The UserB gets the dial tone and starts dialing the digits. In this example the UserB dials the number 2804 of UserC. The dialed digits are reported to MGC using digit map completion event. The digits are reported using the Notify command.

Step 21
MG2 to MGC:

MEGACO/1 [209.110.59.34]:27000
Transaction = 3002 {
  Context = 2 {
    Notify = TermB {ObservedEvents = 1235 {
      20040202T10010000:dd/ce{ds="2804",Meth=FM}}}
  }
}

MGC after receiving the Notify command responds back with the Notify response.

Step 22
MGC to MG2:

MEGACO/1 [216.33.33.61]:27000
Reply = 3002 {
  Context = 2 {Notify = TermB}
}

The UserC is alerted with a ring signal to indicate that a call is to be made.
received. The Add command for the physical termination TermC with
signal descriptor allows the ring signal to be applied on the
termination. The ephemeral termination is also requested to be created
with under specified Local SDP information and fully specified Remote
SDP information.

Step 23:
MGC to MG3:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1240 {
    Context = $ {
      Add = TermC {
        Signals { al/ri }
        Events = 1111{ al/of embedded { al/on } }
      }
      Add = $ {
        Media {
          LocalControl {
            Mode = ReceiveOnly,
          },
          Local {
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 4
          }
        }
        Remote {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      } ; RTP profile for G723 is 4
    }
  }

In this example the SDP local information connection fields IP
address, the media field port numbers are unspecified. The MG3 in its
response indicates the IPAddress and port number used. The contextID
is also not specified indicating the creation of a new context. In
this example the MG3 creates a context with contextID 3. The physical
termination TermA is added to context 1. The mode of the physical
termination was earlier set to Receiveonly and in this message the
ephemeral termination is requested to create with Receiveonly mode.
The ephemeral termination created in this example is EphC. MG3 responds
with the allocated IP address 192.168.0.160 and port number 50000.

Step 24
MG3 to MGC:
  MEGACO/1 [209.110.59.35]: 25000

Madhubabu, et al.                                             [Page 180]
The MGC generates ring back tone towards the UserB to indicate that altering signal has been sent to the called party UserC.

Step 25
MGC to MG2:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1241 {
  Context = 2 {
    Modify = TermB {
      Signals { cg/rt }
    }
  }
}
The MG2 after receiving the Modify command applies the ring back tone specified in the signals descriptor. The Modify response is sent back to MGC.

Step 26
MG2 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1241 {
  Context = 2 {
    Modify= TermB,
  }
}
The UserC after receiving the ring signal goes offhook. The offhook event is reported to MGC in the Notify command.

Step 27
MG3 to MGC:
MEGACO/1 [209.110.59.35]:25000
Transaction = 4001 {
  Context = 3 {
    Notify = TermC {ObservedEvents =1111 {

Madhubabu, et al.                                             [Page 181]
MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 28
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Reply = 4001 {
  Context = 3 {Notify = TermC}
}
The MGC now updates the UserC connected to the Residential gateway 3 with modify command to change the mode of the terminations set to sendrecv.

Step 29:
MGC to MG3:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1242 {
        Context = 3 {
            Modify = TermC {
                Signals { } ; to turn off ringing
                Events = 1235 {al/on},
                Media {
                    LocalControl {
                        Mode = SendRecv,
                    }
                }
            }
            Modify = EphC{
                Media {
                    LocalControl {
                        Mode = SendRecv,
                    }
                }
            }
        }
    }
The Residential gateway responds with the Modify response command.

Step 30
MG3 to MGC:
    MEGACO/1 [209.110.59.35]: 25000
    Reply = 1242 {
        Context = 3 {Modify = TermC , Modify = EphC}
    }
The MGC now updates the UserB connected to the Residential gateway 2 with the local SDP information of UserC as remote SDP information. The Modify command with the remote SDP information is sent to RGW2, for the ephemeral termination.
Step 31
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1243 {
  Context = 2 {
    Modify = TermB {
      Media{
        LocalControl{ mode = sendrecv },
        Remote {
          v=0
          c=IN IP4 192.168.0.160
          m=audio 50000 RTP/AVP 4
        } ; RTP profile for G.723 is 4
      }
    }
  }
}
```

The RGW2 responds with the Modify response.

Step 32
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Reply = 1243 {
  Context = 2 {Modify = TermB }
}
```

The RGW2 updates the remote SDP information and generates the Modify response towards MGC. The Media path is established between UserB and UserC. The two users can be in conversation. The intention of this call scenario is to illustrate the call that is initiated from UserA to UserC. Now the UserB is in connection with UserC and after UserB goes onhook the MGC modifies the remote SDP information of both UserA and UserB. This makes the remote SDP of UserA point to UserC and remote SDP information of UserA point to UserC. When the UserB goes onhook the event is reported to MGC in the Notify command.

Step 33
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Transaction = 3003 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1235 {
      20060202T10000000:al/on}}
  }
}
```

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 34
The MGC also subtracts the terminations TermB and EphB from context2. The context also gets destroyed after deletion of the last termination. The Subtract commands are generated in a single transaction.

Step 35
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1244 {
    Context = 2 {
        Subtract = TermB {Audit{ }},
        Subtract = EphB {Audit{Statistics}}
    }
}

The MG2 responds to the subtract commands generated by MGC.

Step 36
MG2 to MGC:

MEGACO/1 [209.176.47.89]: 26000
Reply = 1244 {
    Context = 2 {
        Subtract = TermB
        Subtract = EphB {
            Statistics {
                rtp/ps=987, ; packets sent
                nt/os=65432, ; octets sent
                rtp/pr=1234, ; packets received
                nt/or=56789, ; octets received
                rtp/pl=10, ; % packets lost
                rtp/jit=30,
                rtp/delay=30 ; average latency
            }
        }
    }
}

UserB is not in connection with UserA and UserC. The MGC now initiates the Modify command towards UserA and UserC.

Step 37
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1245 {
    Context = 1 {
        Modify = TermA {
            Media {
                LocalControl {
                    Mode = sendrecv
                }
            }
        }
    }
}
The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.

Step 38
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1245 {
    Context = 1 {Modify = TermA, Modify = EphA}
}

Similar command is generated towards UserC connected to RGW3.

Step 39
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1246 {
    Context = 3 {
        Modify = TermC {
            Media {
                LocalControl {
                    Mode = sendrecv
                }
            }
        },
        Modify = EphC {
            Media {
                LocalControl {
                    Mode = sendrecv
                }
            }
        }
    }
}

; RTP profile for G723 is 4
The remote SDP information is updated for the ephemeral termination EphC. The mode is changed to send receive. MG3 responds to the MGC with the response for the Modify commands.

Step 40
MG3 to MGC:

MG3 to MGC:

MEGACO/1 [209.110.59.35]: 25000
Reply = 1246 {
    Context = 3 {Modify = TermC, Modify = EphC}
}

The users UserA and UserC can be in conversation as the modes are changed to sendrecv. The call is transferred completely from UserB to UserC. The call can be terminated by any of the users. The UserA after the conversation goes onhook indicating the tearing down of the call. The same is reported in the Notify command from MG1 to MGC.

Step 41
MG1 to MGC:

MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Transaction = 2003 {
    Context = 1 {
        Notify = TermA {ObservedEvents =1112 {
            20010202T10030000:al/on}
        }
    }
}

The MGC responds to the MG1s Notify message.

Step 42
MGC to RGW1:

MGC to RGW1:

MEGACO/1 [216.33.33.61]:27000
Reply = 2003 {
    Context = 1 {
        Notify = TermA
    }
}

The MGC generates a Modify command towards the RGW3 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 43
MGC to RGW3:

MGC to RGW3:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1247 {
    Context = 3 {
        Modify = TermC {
    
Madhubabu, et al.                                             [Page 186]
Signals {cg/bt}
Media {
    LocalControl {
        Mode = recvonly
    }
},
Modify = EphC {
    Media {
        LocalControl {
            Mode = recvonly
        }
    }
}
}
The MG3 responds to this modify request.

Step 44
MG3 to MGC:
    MEGACO/1 [209.110.59.35]: 25000
    Reply = 1247 {
        Context = 3 {
            Modify= TermC, Modify = EphC
        }
    }
The MGC generates a transaction with two subtracts commands one for physical and other for ephemeral terminations

Step 45:
MGC to MG1
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1248 {
        Context = 1 {
            Subtract = TermA {Audit{ }},
            Subtract = EphA {Audit(Statistics)}
        }
    }
The MG1 subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it.
The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 46
MG1 to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Reply = 1248 {
        Context = 1 {
            Subtract = TermA
            Subtract = EphA {

Madhubabu, et al.  [Page 187]
Statistics {
    rtp/ps=1234, ; packets sent
    nt/os=56789, ; octets sent
    rtp/pr=987, ; packets received
    nt/or=65432, ; octets received
    rtp/pl=10, ; % packets lost
    rtp/jit=30,
    rtp/delay=30 ; average latency
}

Step 47
MG3 to MGC:
    MEGACO/1 [209.110.59.35]:25000
    Transaction = 4002 {
        Context = 3 {
            Notify = TermC {ObservedEvents =1235 {
                20050202T10000000:al/on}}
        }
    }

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 48
MGC to MG3:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 4002 {
        Context = 3 {Notify = TermC}
    }

The MGC generates Subtract command towards MG3.

Step 49
MGC to MG3:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1249 {
        Context = 3 {
            Subtract = TermC {Audit{ }},
            Subtract = EphC {Audit{Statistics}}
        }
    }

The MG3 responds to the subtract commands generated by MGC.

Step 50
MG3 to MGC:
    MEGACO/1 [209.110.59.34]:28000
    Reply = 1249 {
        Context = 3 {

Madhubabu, et al.                                             [Page 188]
Subtract = TermC
Subtract = EphC {
  Statistics {
    rtp/ps=987, ; packets sent
    nt/os=65432, ; octets sent
    rtp/pr=1234, ; packets received
    nt/or=56789, ; octets received
    rtp/pl=10, ; % packets lost
    rtp/jit=30,
    rtp/delay=30 ; average latency
  }
}

The MGC generates the message as shown in step 1 to all the three gateways, to enable the users to participate/initiate in further calls.

8.2 Call waiting

The call waiting feature in Telephone networks enables a user to receive two calls simultaneously. The user can switch between the calls. In this example UserA calls UserB and when the conversation is taking place UserC calls UserB. The UserB hears a call waiting tone and switches to this new call using flash hook. UserC disconnects the call and the UserB continues his conversation with UserA. The MGC should support such features and UserB should subscribe for this feature. The figure above suggests that all the three MG’s are controlled by the same MGC. Even though this many not true in real world this assumption holds good for illustration purposes.

<table>
<thead>
<tr>
<th>MG1</th>
<th>MGC</th>
<th>MG2</th>
<th>MG3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;------------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Initial Modify</td>
<td>Initial Modify</td>
<td>Initial Modify</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Modify Response</td>
<td>Modify Resp</td>
<td>Modify Response</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Notify OffHook</td>
<td>Notify Response</td>
<td>Modify Response</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Modify ED:dd/ce(DmapName=dmap1),al/on SD:cg/dt</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al.
The MGC generates the Modify message towards all the three Residential gateways to check for off hook on the terminations. (A wildcard command may also be used in this scenario but for simplicity we consider only command to specific terminations). The MGC in NULL context generated the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified here in the example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control descriptor.

Step 1
MGC to RGW1:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = ReceiveOnly}
      },
      Events = 1111 {al/of}
    }
  }
}

MG after receiving the command from MGC accepts it and responds with the transaction reply. Here for only MG1 is shown to generate the response. In fact all the three RGW generates the response.

Step 2
MG1 to MGC:
MEGACO/1 [209.110.59.34]:25000
Reply = 1234 {
  Context = - (Modify = TermA)
}
In this example User A goes off hook. This event is detected by the 
RGW1 and constructs the Notify message to the MGC. The MG uses the same 
request id (1111) sent by the MGC in its initial command. The 
timestamp of the event detected is also passed as a parameter to the 
observed event.

Step 3
MGI to MGC:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2000 {
  Context = - {
    Notify = TermA {ObservedEvents = 1111 {
      20010202T10000000: al/of}}
  }
}

MGC generates the Notify response and responds with further messages 
towards the MG that generated the Notify command.

Step 4
MGC to MGI:
MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = TermA}
}

The MGC in the following command issues a MODIFY command. The Modify 
command contains a signal descriptor for the application of dial tone 
to the user. The digit map descriptor here is used to configure a 
digit map on the termination. The digit map name used in the example 
is Dmap1 and the dial patter is 2XXX. The event descriptor lists digit 
map completion event of the DTMF detection package and onhook of the 
analog line supervision package. The request id specified in the event 
descriptor is 1112.

Step 5
MGC to MGI:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = - {
    Modify = TermA {
      Signals {cg/dt},
      DigitMap= Dmap1{(2XXX)}
      Events = 1112 {
        al/on, dd/ce {DigitMap=Dmap1}
      },
    }
  }
}

MG after receiving validation responds the Modify command responds the
Step 6
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
    Context = - {Modify = TermA}
}
```

The descriptors are processed in the order that is specified by the MGC. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the termination. The Digit map is ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself contains a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. The digits dialed by user match with the digitmap pattern. Lets assume that the user has dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is generated from MG1 to MGC. The MG again used the same request identifier as specified by the MGC.

Step 7
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
    Context = - {
        Notify = TermA {ObservedEvents =1112 {
            20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
    }
}
```

MGC after receiving the Notify command responds back with the Notify response.

Step 8
MGC to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 2001 {
    Context = - {Notify = TermA}
}
```

MGC after receiving the Notify command starts analyzing the dialed digits. In this example the called subscriber is connected to the RGW2, which is again controlled by the same MGC. The MGC generates a transaction with two commands clubbed into the same Action. The first
command is to create a new context and add the physical termination TermA into it. As the MGC is aware that the destination user UserB is free it indicates MG1 to apply ringback tone to the termination of UserA. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created as a result of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used.

Step 9
MGC to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = $ {
    Add = TermA {
      Signals { cg/rt }
    }
    Add = $ {
      Media {
        LocalControl {
          Mode = ReceiveOnly,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        }
      }
    }
  }
}
```

In this example the connection fields IP address, the media field port number are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating the creation of a new context. In this example the MG creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. MG responds with the allocated IP address 209.110.59.33 and port number 30000.

Step 10
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = TermA,
  }
}
```
Internet-Draft      Megaco/H.248 Call flow examples       July 2001

Add=EphA{
    Media {
        Local {
            v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
a=recvonly
        } ; RTP profile for G.723 is 4
    }
}
}

MGC generates a similar transaction towards the RGW2. The ContextID specified in the action is $. The first command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MGC has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the RGW2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MGC to MG2:
    MEGACO/1 [216.33.33.61]:27000
    Transaction = 1237 {
        Context = $ {
            Add = TermB { Media {
                LocalControl {Mode = Receiveonly} },
                Signals {al/ri}
                Events=1234{al/of},
            },
            Add = $ {Media {
                LocalControl {
                    Mode = Receiveonly,
                },
                Local {
                    v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
            }},
            Remote {
                v=0
c=IN IP4 209.110.59.33
m=audio 30000 RTP/AVP 4
        }
    }
MG2, after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists "ring" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates an ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example MG2 reserves IP address 207.176.47.90 and port number 40000. The MG2 responds to MGC with the following transaction reply.

Step 12
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1237 {
  Context = 2 {
    Add = TermB,
    Add = EphB{
      Media {
        Local {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        } ; RTP profile for G.723 is 4
      }
    }
  }
}

The MGC waits for the UserB to go offhook. Once the UserB goes offhook, MG2 reports the notification of the offhook event to the MGC.

Step 13
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Transaction = 3000 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20000202T10020000:al/of}}
  }
}

The MGC responds to the MG2 with the Notify response.
Step 14
MGC to MG2:
  MEGACO/1 [216.33.33.61]: 27000
  Reply = 3000 {
    Context = 2 {Notify = TermB}
  }
The MGC generates a transaction towards MG2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook event.

Step 15:
MGC to MG2:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1238 {
    Context = 2 {
      Modify = TermB {
        Signals { } ; to turn off ringing
        Events = 1235 {al/on},
        Media {
          LocalControl {
            Mode = SendRecv,
          }
        }
      }
      Modify = EphB{
        Media {
          LocalControl {
            Mode = SendRecv,
          }
        }
      }
    }
  }
The MG2 responds to the request from MGC.

Step 16
MG2 to MGC:
  MEGACO/1 [207.176.47.89]: 26000
  Reply = 1238 {
    Context = 2 {Modify = TermB , Modify = EphB}
  }
The MGC generates a message to the MG1 to stop the ringback tone and to report the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MGC to MG1:

Madhubabu, et al.    [Page 198]
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
        Signals { }
      }
    },
    Modify = EphA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
        Remote {
          v=0
          c=IN IP4 207.176.47.90
          m=audio 40000 RTP/AVP 4
        }
      }
    }
  }
}

The empty signal descriptor in the Modify command for termination TermA, stops the ringback tone at the calling end. The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.

Step 18
MG1 to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1239 {
    Context = 1 {Modify = TermA, Modify = EphA}
  }

The two users can exchange media as the RTP streams are made bi-directional. Now UserC goes offhook to initiate a call towards UserB. The Residential gateway reports the Offhook event to MGC through the Notify message.

Step 19
MG3 to MGC:
  MEGACO/1 [209.110.60.35]:25000
  Transaction = 4000 {
    Context = - {
      Notify = TermC {ObservedEvents =1111 {
        20040202T10000000:al/of} }
    }
  }

Madhubabu, et al.
MGC generates the Notify response.

Step 20
MGC to MG3:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 4000 {
    Context = - {Notify = TermC}
}
```

The MGC in the following command issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial pattern is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 21
MGC to MG3:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
    Context = - {
        Modify = TermC {
            Signals {cg/dt},
            DigitMap= Dmap1{(2XXX)}
            Events = 1112 {
                al/on, dd/ce {DigitMap=Dmap1}
            },
        }
    }
}
```

MG after receiving the Modify command after validation responds to the MGC and starts processing the descriptors listed.

Step 22
MG3 to MGC:

```
MEGACO/1 [209.110.60.35]: 25000
Reply = 1240 {
    Context = - {Modify = TermC}
}
```

The Media gateway applies dial tone and waits for the user to enter the digits. The UserC dials digits 2992. The same is reported to MGC through the Notify command.

Step 23
MG3 to MGC:

```
MEGACO/1 [209.110.60.35]: 25000
```

Madhubabu, et al.                                             [Page 200]
Transaction = 4001 {
    Context = - {
        Notify = TermC {ObservedEvents =1112 {
            20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
    }
}

MGC after receiving the Notify command responds back with the Notify response.

Step 24
MGC to MG3:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 4001 {
        Context = - {Notify = TermC}
    }

The MGC after analyzing the finds that another call is waiting for UserB. Before generating any further commands to UserB, MGC generates ADD command for physical and to create ephemeral termination towards UserC in RGW3. In the Remote SDP information MGC provides the Local SDP information of UserB. The Local SDP information of UserC is left underspecified to enable the RGW3 choose those values.

Step 25
MGC to MG3:
    MEGACO/1 [216.33.33.61]:27000
    Transaction = 1241 {
        Context = $ {
            Add = TermC  { Media {
                LocalControl {Mode = Receiveonly} },
                Signals {al/rt}
                Events=1234{al/of},
            },
            Add  = $ {Media {
                LocalControl {
                    Mode = Receiveonly,
                },
                Local {
                    v=0
                    c=IN IP4 $
                    m=audio $ RTP/AVP 4
                    },
                    Remote {
                    v=0
                    c=IN IP4 207.176.47.90
                    m=audio 40000 RTP/AVP 4
                    } ; RTP profile for G.723 is 4
            }
        }
    }

Madhubabu, et al.                                             [Page 201]
MG3 after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 3. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists "ringback" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW3 creates a ephemeral termination with TerminationId EphC. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example MG2 reserves IP address 192.168.0.160 and port number 50000.

The MG3 responds to MGC with the following transaction reply.

Step 26
MG3 to MGC:

```
MEGACO/1 [209.110.60.35]: 25000
Reply = 1241 {

  Context = 3 {
    Add = TermC,
    Add = EphC{
      Media {
        Local {
          v=0
          c=IN IP4 192.168.0.160
          m=audio 50000 RTP/AVP 4
        }
      }
    }
  }
}
```

If generates a Modify command to the UserB with call waiting tone in the Signal Descriptor.

Step 27
MGC to MG2:

```
MEGACO/1 [216.33.33.61]:26000
Transaction = 1242 {

  Context = 2 {
    Modify = TermB { Media {
      LocalControl {Mode = SendRecv},
      Signals {cg/cw}
      Events=1234{al/f1, al/on},
    }
  }
}
```
MG2 generates the response for the Modify command generated by MGC.

Step 28
MG2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Reply = 1242 {
    Context = 2 {
        Modify= TermB }
}

The UserB press flash button on the phone and the Residential gateway generates Notify command towards the MGC.

Step 29
MG2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Transaction = 3001 {
    Context = 2 {
        Notify = TermB {ObservedEvents =1234 {
            20040202T10000000:al/fl}}
    }
}

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 30
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Reply = 3001 {
    Context = 2 {Notify = TermB}
}

The MGC now is aware that the UserB should be in voice conversation with UserC instead of UserA. The MGC now has to update the remote SDP information of UserC. Such that any the same local SDP information is used by UserB to continue calling UserC. The Local SDP information of UserC is provided as the remote SDP information to UserB. Since the Ephemeral termination is already in context, the Modify command is used by MGC to update the remote SDP information.

Step 31
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1243 {
    Context = 2 {
        Modify = TermB {
            Media {
                LocalControl {
                    Mode = sendrecv}
            }
        }
    }

Madhubabu, et al.
The empty signal descriptor in the Modify command for termination TermB, stop the Call waiting tone at the calling end. The remote SDP information is updated for the ephemeral termination EphB. The mode is changed to send receive. MG2 responds to the MGC with the response for the Modify commands.

Step 32
MG2 to MGC:
    MEGACO/1 [207.176.47.89]: 26000
    Reply = 1243 {
        Context = 2 {Modify = TermB, Modify = EphB}
    }

The MGC now generates a Modify command to change the mode of the termination from receive only to send receive. At the same time MGC also sends an empty signal descriptor to stop the ring back tone that was earlier applied on termination TermC of UserC.

Step 33
MGC to MG3:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1244 {
        Context = 3 {
            Modify = TermC {
                Media {
                    LocalControl {
                        Mode = sendrecv
                    }
                    v=0
                    c=IN IP4 192.168.0.160
                    m=audio 50000 RTP/AVP 4
                }
            }
        }
        Signals { }
    },
    Modify = EphC {
        Media {
            LocalControl {
                Mode = sendrecv
            }
        }
    }
The empty signal descriptor in the Modify command for termination TermC, stop the ringback tone at the calling end. The mode is changed to send receive. MG3 responds to the MGC with the response for the Modify commands.

Step 34
MG3 to MGC:

MEGACO/1 [209.110.60.34]: 25000
Transaction = 4002 { Context = 3 {Notify = TermC, Notify = EphC}

Now the UserB and UserC are connected (through RTP Media). After the conversation in the example UserC goes onhook to termination its call with UserB. The Onhook event is reported to MGC though the Notify command.

Step 35
MG3 to MGC:

MEGACO/1 [209.110.60.34]:25000
Transaction = 4002 { Context = 3 {Notify = TermC {ObservedEvents =1234 {20050202T10030000:al/on}}}}

The MGC responds to the MG3s Notify message.

Step 36
MGC to RGW3:

MEGACO/1 [216.33.33.61]:27000
Transaction = 4002 { Context = 3 {Notify = TermC}

The MGC generates Subtract command towards RGW3.

Step 37
MGC to MG3:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1245 { Context = 3 { Subtract = TermC {Audit{ }}, Subtract = EphC {Audit(Statistics)}

Madhubabu, et al. [Page 205]
The MG3 responds to the subtract commands generated by MGC.

Step 38
MG3 to MGC:
  MEGACO/1 [209.110.59.35]:25000
  Reply = 1245 {
    Context = 3 {
      Subtract = TermC
      Subtract = EphC {
        Statistics {
          rtp/ps=987, ; packets sent
          nt/os=65432, ; octets sent
          rtp/pr=1234, ; packets received
          nt/or=56789, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }
}

The MGC generates a Modify command towards the RGW2 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 39
MGC to RGW2:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1246 {
    Context = 2 {
      Modify = TermB {
        Signals {cg/bt}
        Events = 1235 { al/fl, al/on }
        Media {
          LocalControl {
            Mode = recvonly
          }
        }},
      Modify = EphB {
        Media {
          LocalControl {
            Mode = recvonly
          }
        }
      }
    }
  }
}

The MG2 responds to this modify request.
Step 40
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1246 {
  Context = 2 {
    Modify = TermB, Modify = EphB
  }
}
The User B press flash to continue its call with UserA. The flash event
is reported to MGC in the Notify command.

Step 41
MG2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Transaction = 3002 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20060202T10000000:al/fl}}
  }
}
MGC generates the Notify response and responds with further messages
towards the MG that generated the Notify command.

Step 42
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Reply = 3002 {
  Context = 2 {Notify = TermB}
}
The MGC now generates a Modify command towards the UserB with the
Local SDP information of UserA as Remote SDP information for the
ephemeral termination EphB. This enables UserB to continue the call
with UserA.

Step 43
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1247 {
  Context = 2 {
    Modify = TermB {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    },
    Modify = EphB {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    }
  }
}
The empty signal descriptor in the Modify command for termination TermB, stop the busy tone at the calling end. The remote SDP information is updated for the ephemeral termination EphB. The mode is changed to send receive. MG2 responds to the MGC with the response for the Modify commands.

Step 44
MG2 to MGC:
    MEGACO/1 [207.176.47.89]: 26000
    Reply = 1247 {
        Context = 2 {Modify = TermB, Modify = EphB}
    }

The UserB and UserA can continue their conversation. The call can be tear down either by UserA or UserB. In this example we assume that UserA terminates the Call. The UserA goes onhook and the users action is reported to MGC using the Notify command.

Step 45
RGW1 to MGC:
    MEGACO/1 [209.110.59.34]:25000
    Transaction = 2002 {
        Context = 1 {
            Notify = TermA {ObservedEvents =1112 {
                20010202T10030000:al/on}
            }
        }
    }

The MGC responds to the MG1s Notify message.

Step 46
MGC to RGW1:
    MEGACO/1 [216.33.33.61]:27000
    Reply = 2002 {
        Context = 1 {
            Notify = TermA
        }
    }

The MGC generates a Modify command towards the RGW2 for applying busy tone to the called subscriber. The mode of both terminations is set to
Step 47
MGC to RGW2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1248 {
  Context = 2 {
    Modify = TermB {
      Signals {cg/bt}
      Media {
        LocalControl {
          Mode = recvonly}
      }
    },
    Modify = EphB {
      Media {
        LocalControl {
          Mode = recvonly}
      }
    }
  }
}

The MG2 responds to this modify request.

Step 48
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1248 {
  Context = 2 {
    Modify= TermB, Modify = EphB}
}

The MGC generates transactions with two subtracts commands one for
physical and other for ephemeral terminations. The MGC does the same
for both the Contexts one at RGW1 and the other at RGW2.

Step 49:
MGC to MG1

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1249 {
  Context = 1 {
    Subtract = TermA {Audit {}},
    Subtract = EphA {Audit {Statistics}}
  }
}

The MG subtracts the two terminations from the context. The context
itself is deleted with the subtract of the last termination from it.
The MG1 responds to this transaction from MGC with statistics on
ephemeral termination.
Step 50
MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1249 {
  Context = 1 {
    Subtract = TermA
    Subtract = EphA {
      Statistics {
        rtp/ps=1234, ; packets sent
        nt/os=56789, ; octets sent
        rtp/pr=987, ; packets received
        nt/or=65432, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

The UserB after hearing the busy tone goes onhook, the same is recognized by the Media gateway and generates Notify command towards the MGC.

Step 51
MG2 to MGC:

MEGACO/1 [207.176.47.89]:26000
Transaction = 3003 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20060202T10000000:al/on}}
  }
}

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 52
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Reply = 3002 {
  Context = 2 {Notify = TermB}
}

The MGC then generates subtract commands towards RGW2.

Step 53
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1250 {

Madhubabu, et al.
The MG2 responds to the subtract commands generated by MGC.

Step 54
MG2 to MGC:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1250 {
    Context = 2 {
        Subtract = TermB
        Subtract = EphB {
            Statistics {
                rtp/ps=987, ; packets sent
                nt/os=65432, ; octets sent
                rtp/pr=1234, ; packets received
                nt/or=56789, ; octets received
                rtp/pl=10, ; % packets lost
                rtp/jit=30,
                rtp/delay=30 ; average latency
            }
        }
    }
}

The MGC generates the message as shown in step 1 to all the Media Gateways, to enable the users to participate/initiate in further calls.

9. Conferencing

A Media Gateway optionally performs media conferencing. Media Gateways that support multipoint conferences might allow three or more terminations in a context. In this section we will illustrate conferencing between three users. These call flows make use of the Topology descriptor. A topology descriptor is used to specify flow directions between terminations in a Context.

<table>
<thead>
<tr>
<th>MG1</th>
<th>MGC</th>
<th>MG2</th>
<th>MG3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------------------</td>
<td>-----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Modify</td>
<td>Initial Modify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------------------</td>
<td>-----------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Madhubabu, et al. [Page 211]
Modify Response | Modify Resp | Modify Response
--------------|------------|-----------------
                | Notify OffHook | Modify Response
                | Notify Response | Modify Response
                | Modify ED:dd/cd,al/on SD:cg/dt | Modify Response
<-----|----------------|---------------|
Dial  | Modify Resp | Modify Resp
Tone | --------------|---------------|
-----|---------------|---------------|
Digits | Notify OffHook | Modify Resp
      | Notify Digits | Modify Resp
      | Notify Response | Modify Resp
      | Modify ringback | Modify Resp
      | Modify Ring | Modify Resp
<-----|----------------|---------------|
Ring  | Modify Resp | Modify Resp
back | --------------|---------------|
<-----|---------------|---------------|
Add Phy | Notify OffHook | Modify Resp
Local Unspecified | Modify Resp
<-----|---------------|---------------|
Add Phy | Modify Resp | Modify Resp
Resp | Modify Resp | Modify Resp
Add EphAResp Local specified | Modify Resp
<-----|---------------|---------------|
 Modify Resp | Modify Resp
Add Phy | Modify Resp
Add $   Local Unspecified Remote Specified | Modify Resp
Add Phy Resp Add EphB Resp Local Specified | Modify Resp
<-----|---------------|---------------|
 Modify Resp | Modify Resp
Add Phy | Modify Resp
Add $   Local unspecified Remote Specified | Modify Resp
Add Phy Resp Add EphB Resp Local Specified | Modify Resp
<-----|---------------|---------------|
 Modify Resp | Modify Resp
RTP MEDIA | Modify Resp
\-------------------------------/ | Modify Resp
--------------|---------------|
Notify Flash | Modify Resp
--------------|---------------|
Notify Resp | Modify Resp
RecvOnly | Modify Resp
--------------|---------------|
DialTone | Modify Resp
--------------|---------------|
Notify Digits | Modify Resp
--------------|---------------|
Notify Resp | Modify Resp

Madhubabu, et al. [Page 212]
The conferencing feature in PSTN allows a user to speak with multiple users at the same time. This feature
should be supported by MGC so that it is capable of generating required messages towards MG. In this example we assume that the MGC is capable of supporting Conferencing. UserB, the called party, initially receives a call from UserA. For adding UserC, UserB press flash hook and dials UserC number and after UserC answers the call goes flash hook again to connect UserA and UserC. Now UserA, User B and UserC are in the call.

The MGC generates the Modify message towards all the three Residential gateways to check for off hook on the terminations. (A wildcard command may also be used in this scenario but for simplicity we consider only command to specific terminations). We are not considering the embedded signal and event descriptors here. The MGC in NULL context generates the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified here in the example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control descriptor.

Step 1
MGC to RGW1:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = ReceiveOnly
        }
      },
      Events = 1111 {al/of}
    }
  }
}

MG after receiving the command from MGC accepts it and responds with the transaction reply. Here only MG1 is shown to generate the response. In fact all the RGW generates the response.

Step 2

MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1234 {
  Context = - (Modify = TermA)
}

In this example User A goes off hook. This event is detected by the RGW1 and constructs the Notify message to the MGC. The MG uses the same request id (1111) sent by the MGC in its initial command. The timestamp of the event detected is also passed as a parameter to the observed event.
Step 3
MGC to MG1:

MEGACO/1 [209.110.59.34]:25000
Transaction = 2000 {
  Context = - {
    Notify = TermA {ObservedEvents =1111 {
      20010202T10000000:al/of}}
  }
}

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 4
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Reply = 2000 {
  Context = - {Notify = TermA}
}

The MGC in the following command issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial pattern is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 5
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = - {
    Modify = TermA {
      Signals {cg/dt},
      DigitMap= Dmap1(2XXX)
      Events = 1112 {
        al/on, dd/ce {DigitMap=Dmap1}
      },
    }
  }
}

MG after validating the Modify command responds to the MGC and starts processing the descriptors listed.

Step 6
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {

The descriptors are processed in the order that is specified by the MGC. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the Termination. The Digit map will be made ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself may contain a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. The digits dialed by the user match with the digit map pattern. Let's assume that the user has dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is generated from MG1 to MGC. The MG again used the same request identifier as specified by the MGC.

Step 7
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
  Transaction = 2001 {
    Context = - {
      Notify = TermA {ObservedEvents =1112 {
        20010202T10010000:dd/ce{ds="2992",Meth=FM}}}
    }
  }

MGC after receiving the Notify command responds back with the Notify response.

Step 8
MGC to MG1:
  MEGACO/1 [216.33.33.61]: 27000
    Reply = 2001 {
      Context = - {Notify = TermA}
    }

MGC after receiving the Notify command starts analyzing the dialed digits. In this example the called subscriber is connected to the RGW2, which is again controlled by the same MGC. The MGC generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. As the MGC is aware that the destination user UserB is free it instructs MG1 to apply ringback tone to the termination of UserA. The second command is generated to create an ephemeral termination and add the created termination in the same
context that was created as a result of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used.

Step 9
MGC to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1236 {
  Context = $ {
    Add = TermA {
      Signals { cg/rt }
    }
    Add = $ {
      Media {
        LocalControl {
          Mode = ReceiveOnly,
        },
        Local {
          v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
        }
      }
    }
  }
}

In this example the connection fields IP address, the media field port number are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating the creation of a new context. In this example the MG creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. MG responds with the allocated IP address 209.110.59.33 and port number 30000.

Step 10
MG1 to MGC:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          v=0
c=IN IP4 209.110.59.33
        }
      }
    }
  }
}

Madhubabu, et al.                                             [Page 218]
MGC generates a similar transaction towards the RGW2. The ContextID specified in the action is $$. The first command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MGC has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the RGW2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MGC to MG2:

MEGACO/1 [216.33.33.61]:27000
Transaction = 1237 {
Context = $$ {
    Add = TermB { Media {
        LocalControl {Mode = Receiveonly} },
        Signals {al/ri}
        Events=1234{al/of},
    },
    Add = $$ {Media {
        LocalControl {
            Mode = Receiveonly,
        },
        Local {
            v=0
            c=IN IP4 $ 
            m=audio $ RTP/AVP 4
        },
        Remote {
            v=0
            c=IN IP4 209.110.59.33
            m=audio 30000 RTP/AVP 4
        } ; RTP profile for G.723 is 4
    }
}
}
MG2 after receiving the new transaction from MGC starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists “ring” signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates a ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC. The MG allocates the necessary resources for processing the media descriptor for the ephemeral termination. The MG responds to the MGC by specifying the IP address reserved for the local connection. In this example MG2 reserves IP address 207.176.47.90 and port number 40000. The MG2 responds to MGC with the following transaction reply.

Step 12
MG2 to MGC:
   MEGACO/1 [207.176.47.89]: 26000
   Reply = 1237 {
      Context = 2 {
         Add = TermB, 
         Add = EphB{
               Media {
                  Local {
                     v=0
                     c=IN IP4 207.176.47.90
                     m=audio 40000 RTP/AVP 4
                  }
               } ; RTP profile for G723 is 4
         }
      }
   }

The MGC waits for the UserB to go offhook. Once the UserB goes offhook, MG2 reports the notification of the offhook event to the MGC.

Step 13
MG2 to MGC:
   MEGACO/1 [207.176.47.89]: 26000
   Transaction = 3000 {
      Notify = TermB {ObservedEvents =1234 {
                     20000202T10020000:al/of}}
   }

The MGC responds to the MG2 with the Notify response.

Step 14
MGC to MG2:
   MEGACO/1 [216.33.33.61]: 27000
   Reply = 3000 {

The MGC generates a transaction towards MG2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook event, flash hook and the dd/ce event.

Step 15:
MGC to MG2:
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1238 {
      Context = 2 {
         Modify = TermB {
            Signals { } ; to turn off ringing
            Events = 1235 {al/on, al/fl { signals cg/dt, events
                           dd/ce{dmap1}, al/on }},
            Media {
               LocalControl {
                  Mode = SendRecv,
               }
            }
         }
         Modify = EphB{
            Media {
               LocalControl {
                  Mode = SendRecv,
               }
            }
         }
      }
   }

The MG2 responds to the request from MGC.

Step 16
MG2 to MGC:
   MEGACO/1 [207.176.47.89]: 26000
   Reply = 1238 {
      Context = 2 {Modify = TermB , Modify = EphB}
   }

The MGC generates message to the MG1 to stop the ringback tone and to report the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MGC to MG1:
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1239 {
      Context = 1 {

Modify = TermA {
    Media {
        LocalControl {
            Mode = sendrecv
        }
        Signals { }
    },
    Modify = EphA {
        Media {
            LocalControl {
                Mode = sendrecv
            }
            Remote {
                v=0
                c=IN IP4 207.176.47.90
                m=audio 40000 RTP/AVP 4
            }
        }
    }
}

The empty signal descriptor in the Modify command for termination TermA, stop the ringback tone at the calling end. The remote SDP information is updated for the ephemeral termination EphA. The mode is changed to send receive. MG1 responds to the MGC with the response for the Modify commands.

Step 18
MG1 to MGC:
    MEGACO/1 [209.110.59.34]: 25000
   Reply = 1239 {
        Context = 1 {Modify = TermA, Modify = EphA}
    }

The two users can exchange media, as the RTP streams are made bi-directional. The figure below shows the terminations at both in both the Contexts.
The UserB now presses flash to dial the UserC number. The UserB flash event is reported to MGC using the Notify message.

Step 19
MG2 to MGC:

```
MEGACO/1 [209.110.59.34]:25000
Transaction = 3001 {
    Context = 2 {
        Notify = TermB {ObservedEvents =1235 {
            20040202T10000000:al/fl}
        }
    }
}
```

MGC generates the Notify response.

Step 20
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 3001 {
    Context = 2 {Notify = TermB}
}
```

The UserB gets the dial tone and starts dialing the digits. In this example the UserB dials the number 2804 of UserC. The dialed digits are reported to MGC using digit map completion event. The digits are reported using the Notify command.

Step 21
MG2 to MGC:

```
MEGACO/1 [209.110.59.34]: 27000
Transaction = 3002 {
    Context = 2 {
        Notify = TermB {Observed Events =1235 {
```
MGC after receiving the Notify command responds back with the Notify response.

Step 22
MGC to MG2:
MEGACO/1 [216.33.33.61]: 27000
Reply = 3002 {
   Context = 2 {Notify = TermB}
}
The UserC is alerted with ring signal to indicate that a call is to be received. The Add command for the physical termination TermC with signal descriptor allows the ring signal to be applied on the termination. The ephemeral termination is also requested to be created with under specified Local SDP information and fully specified Remote SDP information.

Step 23:
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1240 {
   Context = $ {
      Add = TermC {
         Signals { al/ri }
         Events = 1111{ al/of embedded { al/on } } }
      Add = $ {
         Media {
            LocalControl {
               Mode = ReceiveOnly,
            },
            Local {\n               v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 4
         }
         }
         Remote {
            v=0
            c=IN IP4 207.176.47.90
            m=audio 40000 RTP/AVP 4
         }
      } ; RTP profile for G723 is 4
   }
}

Madhubabu, et al.                                  [Page 224]
In this example the SDP local information connection fields IP address, the media field port numbers are unspecified. The MG in its response indicates the IPAddress and port number used. The contextID is also not specified indicating the creation of a new context. In this example the MG creates a context with contextID 3. The physical termination TermC is added to context 3. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphC. MG responds with the allocated IP address 192.168.0.160 and port number 50000.

Step 24
MG1 to MGC:
    MEGACO/1 [209.110.59.34]: 25000
    Reply = 1240 {
    Context = 3 {
    Add = TermC,
    Add=EphC{
    Media {
    Local {
  v=0
  c=IN IP4 192.168.0.160
  m=audio 50000 RTP/AVP 4
  a=recvonly
        } ; RTP profile for G.723 is 4
    }
    }
    }
    }
    
The MGC generates ring back tone towards the UserB to indicate that altering signal has been sent to the called party UserC.

Step 25
MGC to MG2:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1241 {
    Context = 2 {
    Modify = TermB {
    Signals { cg/rt }
    }
    }
    }
    
The MG2 after receiving the Modify command applies the ring back tone specified in the signals descriptor. The Modify response is sent back to MGC.

Step 26
MG2 to MGC:
    MEGACO/1 [207.176.47.89]: 26000
The UserC after receiving the ring signal goes offhook. The offhook event is reported to MGC in the Notify command.

Step 27
MG3 to MGC:
MEGACO/1 [209.110.59.34]:28000
Transaction = 4001 {
    Context = 3 {
        Notify = TermC {ObservedEvents = 1111 {
            20050202T10000000:al/of}}
    }
}

MGC generates the Notify response and responds with further messages towards the MG that generated the Notify command.

Step 28
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Reply = 4001 {
    Context = 3 {Notify = TermC}
}
The MGC now updates the UserC connected to the Residential gateway 3 with modify command to change the mode of the terminations set to sendrecv.

Step 29:
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1242 {
    Context = 3 {
        Modify = TermC {
            Signals { } ; to turn off ringing
            Events = 1235 {al/on},
            Media {
                LocalControl {
                    Mode = SendRecv,
                }
            }
        }
        Modify = EphC{
            Media {
                LocalControl {
                    Mode = SendRecv,
                }
            }
        }
    }
}

Madhubabu, et al.
The Residential gateway responds with the Modify response command.

Step 30
MG3 to MGC:

MEGACO/1 [207.176.47.89]: 26000
Reply = 1242 {
  Context = 3 {Modify = TermC , Modify = EphC}
}

The MGC now updates the UserB connected to the Residential gateway 2 with the local SDP information of UserC as remote SDP information. The Modify command with the remote SDP information is sent to RGW2, for the ephemeral termination.

Step 31
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1243 {
  Context = 2 {
    Modify = TermB { Event = 1234 { al/fl, al/on }}
    Modify = EphB {
      Media{
        LocalControl{ mode = sendrecv },
        Remote {
          v=0
          c=IN IP4 192.168.0.160
          m=audio 50000 RTP/AVP 4
        }
      }
    }
  }
}

The RGW2 responds with the Modify response.

Step 32
MG2 to MGC:

MEGACO/1 [207.176.47.89]: 28000
Reply = 1243 {
  Context = 2 {Modify = TermB, Modify = EphB }
}

The RGW2 updates the remote SDP information and generates the Modify response towards MGC. The Media path is established between UserB and UserC. The two users can be in conversation. The following figure shows the different context created and the terminations in these Contexts.

The MGC then generates a Modify command towards MG1. The UserA mode is set to Receive only so that UserA generate RTP is not directed towards UserB.
Step 33
MGC to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1244 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = recvonly
        }
      }
      Signals {}
    },
    Modify = EphA {
      Media {
        LocalControl {
          Mode = recvonly
        }
      }
    }
  }
}
```

Step 34
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1244 {
  Context = 1 {Modify = TermA, Modify = EphA}
}
```

The Ephemeral termination EphB now has the Remote SDP information of UserC. This enables the media flow between UserB and UserC. UserA, even though is in valid context, doesn’t generate any RTP media.

The UserB after the establishment of RTP media with UserC goes flash hook to indicate the conference creation with UserA. The Notify event from RGW2 is reported to MGC using the Notify message. The UserB press flash button on the phone and the Residential gateway generates Notify command towards the MGC.

Step 35
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]: 26000
Transaction = 3003 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20050202T10000000:al/fl}}
  }
}
```
MGC generates the Notify.

Step 36
MGC to MG2:
MEGACO/1 [216.33.33.61]: 27000
Reply = 3003 {
  Context = 2 {Notify = TermB}
}
This call flow section illustrates the call between three users. Already UserB and UserC are connected. The UserA remote SDP information points to the UserB. Now the MGC has to add one more ephemeral termination in each of the contexts created at the three different user Gateways. In this scenario we show that EphA, EphB and EphC are initially created. EphA of UserA points to EphB of UserB, but since the mode of the termination is "recvonly", this can be modified as required. The EphB or UserB is virtually connected to EphC of UserC. The MGC for connecting UserA and UserC generates Add command towards RGW1 for creating an Ephemeral termination. After receiving the response of the Add command indicating the creation of EphD, the MGC passes the local SDP information of EphD in the Add command towards UserC for creation of another Ephemeral termination. RGW3 creates Ephemeral Termination EphF with remote information pointing towards EphD of UserA. The MGC now after receiving response from UserC for EphF creation "Modifies" the EphD with the remote SDP information of EphF. Thus connecting the UserA and UserC. The topology descriptor is used to illustrate the control of media flows between the terminations in the context.

The MGC generates Add command towards UserB for creating an Ephemeral termination. The Local SDP information of EphA is passed as remote SDP information for this newly created termination. After receiving the response from UserB indicating the successful creation of EphE, the local SDP information of EphE is passed to UserA as remote SDP information for EphA in the "Modify" command. UserA’s EphA is modified to point towards the EphE of UserB. Now all the Users have 3 termination in their contexts. Thus enabling them to participate in conference. The following paragraphs illustrates the different messages exchanged in detail, to illustrate the addition of new user in conference. The same can be extended to any number of users but for simplicity, only 3 users are considered.

The MGC generates an add command towards MG1 for creation of another ephemeral termination so that the media information is directed towards UserC. The ephemeral termination is requested to be added in context 1. The topology descriptor that shows the directions of media flow inside the context is initially set to ISOLOATE so that there is no media immediately transferred between these termination within the context.

Step 37
MGC to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1244 {
  Context = 1
  Topology TermA,EphA,isolat, TermA, $, isolate, EphA,$, isolate{
```

Madhubabu, et al. [Page 230]
The Residential gateway now generates an ephemeral termination, with newly allocated resources for it. The IP address that is allocated is 192.168.0.155 and the port number is 35000. The response is sent back to MGC.

Step 38
MG1 to MGC:
MEGACO/1 [209.110.59.34]: 25000
Reply = 1244 {
  Context = 1 {
    Add=EphD{
      Media {
        Local {
          v=0
          c=IN IP4 $192.168.0.155
          m=audio $ RTP/AVP 4
          a=recvonly
        }
      }
    }
  }
}
The MGC after receiving this information generates another ADD command towards the Residential gateway 3, such that the UserA SDP information is indicated to UserC.

Step 39
MGC to MG3:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1245 {
  Context = 3
  Topology TermC,EphC,bothway, TermC, $, bothway, EphC,$, bothway {
    Add = $ {
      Media {
        
      }
    }
  }
}
The Residential gateway now generates an ephemeral termination, with newly allocated resources for it. The IP address that is allocated is \texttt{192.168.0.100} and the port number is 55000. The response is sent back to MGC.

Step 40
MG3 to MGC:

\texttt{MEGACO/1 [209.110.59.34]: 25000}
Reply = 1245 {
Context = 3 {
Add=EphF{
Media {
Local {
  v=0
  c=IN IP4 $192.168.0.100$
  m=audio 55000 RTP/AVP 4
  a=recvonly
}
  ; RTP profile for G.723 is 4
}
}
}
}

The MGC after receiving the response from the UserC now indicates the same information towards the Residential gateway 1. Thus enabling media flow between the UserA and UserC.

Step 41
MGC to MG1:

\texttt{MEGACO/1 [216.33.33.61]: 27000}
Transaction = 1246 {
}

Madhubabu, et al.  
[Page 232]
Context = 1
Topology TermA,EphA, isolate, TermA, EphD, bothway, EphA,EphD, isolate{
  Modify = EphD {
    Media {
      {
        LocalControl {
          Mode = ReceiveOnly,
        },
        Remote {
          v=0
c=in IP4 192.168.0.100
m=audio 55000 RTP/AVP 4
        }
      }
    }
  }
}

The Residential gateway now generates the Modify response to the MGC.

Step 42
MG1 to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1246 {
    Context = 1 {
      Modify=EphD
    }
  }

Madhubabu, et al. [Page 233]
The UserA and UserC are in conversation now. The UserB and UserC were already in conversation. Now the UserA and UserB need to be connected through RTP media. The MGC now generates ADD command towards the UserB of the Residential gateway 2. The Local information of UserA is indicated as the remote information of the newly created ephemeral termination. This ephemeral termination is requested to be added to the earlier context 2 itself.

Step 43
MGC to MG2:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1247 {
  Context = 2
  Topology TermB, EphB, bothway, TermB, $, bothway, EphB, $, bothway {
    Add = $ {
      Media {
        LocalControl {
          Mode = sendrecv,
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 4
        }
      }
      Remote {

      }
  }
}
```
The Residential gateway 2 now creates an ephemeral termination, with newly allocated resources for it. The IP address that is allocated is 192.168.0.110 and the port number is 45000. The response is sent back to MGC.

Step 44
MG2 to MGC:
MEGACO/1 [209.110.59.34]: 26000
Reply = 1247 {
  Context = 2 {
    Add=EphE{
      Media {
        Local {
          v=0
          c=IN IP4 192.168.0.110
          m=audio 45000 RTP/AVP 4
          a=recvonly
        } ; RTP profile for G.723 is 4
      }
    }
  }
}

The MGC after receiving the local SDP information update the Ephemeral termination EphA of the Residential Gateway 1. The Modify command is generated towards RGW1.

Step 45
MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1248 {
  Context = 1
    Modify = EphA {
      Media {
        LocalControl {
          Mode = sendrecv,
        },
        Remote {
          v=0
        }
      }
    }
  }
}
The Residential gateway now generates the Modify response to the MGC.

Step 46
MGI to MGC:

   MEGACO/1 [209.110.59.34]: 25000
   Reply = 1248 {
       Context = 1 {
           Modify=EphA
       }
   }

All the terminations in all the contexts are made to sendrecv, thus all
the three users are effectively in conference. The following figure
shows the direction of media transfer between terminations inside each
of the contexts.

Madhubabu, et al.                                             [Page 236]
In this example UserB goes onhook. The onhook event is reported to MGC using the Notify message.

Step 47
MG2 to MGC:

```
MEGACO/1 [207.176.47.89]:26000
Transaction = 3006 {
   Context = 2 {
      Notify = TermB {ObservedEvents =1234 {
         20010202T10030000:al/on}
      }
   }
}
```

The MGC responds to the MG’s notify message.

Step 48
MGC to MG2:

```
MEGACO/1 [216.33.33.61]:27000
Reply = 3006 {
   Context = 2 {
      Notify = TermB
   }
}
```

The MGC generates Subtract command towards the Residential Gateway 2 to delete all the terminations in the context 2. The statistics are requested only for the ephemeral terminations.

Step 49
MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1249 {
  Context = 2 {
    Subtract = TermB {Audit{}},
    Subtract = EphB {Audit{Statistics}}
    Subtract = EphE {Audit{Statistics}}
  }
}

The MG2 responds to the subtract commands generated by MGC.

Step 50

MG2 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1249 {
  Context = 2 {
    Subtract = TermB
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    Subtract = EphE {
      Statistics {
        rtp/ps=1987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}

The MGC generates Subtract command for the ephemeral termination EphA towards the RGW1.

Step 51

MGC to MG2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1250 {
  Context = 1 {
    Subtract = EphA {Audit{Statistics}}
  }
}
The MG1 responds to the subtract commands generated by MGC.

Step 52
MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1250 {
    Context = 2 {
        Subtract = EphA {
            Statistics {
                rtp/ps=987, ; packets sent
                nt/os=65432, ; octets sent
                rtp/pr=1234, ; packets received
                nt/or=56789, ; octets received
                rtp/pl=10, ; % packets lost
                rtp/jit=30,
                rtp/delay=30 ; average latency
            }
        }
    }
}

The MGC generates similar command towards UserC at RGW2, for subtracting the ephemeral termination EphC.

Step 53
MGC to MG3:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1251 {
    Context = 3 {
        Subtract = EphC {Audit{Statistics}}
    }
}

The MG3 responds to the subtract commands generated by MGC.

Step 54
MG3 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1251 {
    Context = 3 {
        Subtract = EphC {
            Statistics {
                rtp/ps=987, ; packets sent
                nt/os=65432, ; octets sent
                rtp/pr=1234, ; packets received
                nt/or=56789, ; octets received
                rtp/pl=10, ; % packets lost
                rtp/jit=30,
                rtp/delay=30 ; average latency
            }
        }
    }
}
After completing the conversation with UserA, UserC goes onhook. The same is indicated towards MGC in the Notify command.

Step 55
MG3 to MGC:
   MEGACO/1 [209.110.59.34]:25000
   Transaction = 4004 {
      Context = 3 {
         Notify = TermC {ObservedEvents =1111 {
            20060202T10030000:al/on}
         }
      }
   }

The MGC responds to the MG’s notify message.
Step 56
MGC to MG3:

```
MEGACO/1 [216.33.33.61]: 27000
Reply = 4004 {
  Context = 3 {
    Notify = TermC
  }
}
```

The MGC generates a Modify command towards the RGW1 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 57
MGC to MG1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1252 {
  Context = 1 {
    Modify = TermA {
      Signals {cg/bt}
      Media {
        LocalControl {
          Mode = recvonly}
      }
    },
    Modify = EphD {
      Media {
        LocalControl {
          Mode = recvonly}
      }
    }
  }
}
```

The MG1 responds to this modify request.

Step 58
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1252 {
  Context = 1 {
    Modify = TermA, Modify = EphD
  }
}
```

The MGC generates Subtract command RGW3.

Step 59
MGC to MG3:

```
```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1254 {
  Context = 3 {
    Subtract = TermC {Audit{ }},
    Subtract = EphF {Audit{Statistics}}
  }
}
The MG3 responds to the subtract commands generated by MGC.

Step 60
MG3 to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1254 {
    Context = 3 {
      Subtract = TermC
      Subtract = EphF {
        Statistics {
          rtp/ps=987, ; packets sent
          nt/os=65432, ; octets sent
          rtp/pr=1234, ; packets received
          nt/or=56789, ; octets received
          rtp/pl=10, ; % packets lost
          rtp/jit=30,
          rtp/delay=30 ; average latency
        }
      }
    }
  }
}
The UserA goes onhook after hearing the busy tone. The same is indicated to MGC using the Notify command.

Step 61
MG1 to MGC:
  MEGACO/1 [209.110.59.34]: 25000
  Transaction = 2002 {
    Context = 1 {
      Notify = TermA {ObservedEvents = 1112 {
        20010202T10010000:al/on}
    }
  }
}
MGC after receiving the Notify command responds back with the Notify response.

Step 62
MGC to MG1:
  MEGACO/1 [216.33.33.61]: 27000
  Reply = 2002 {
    Context = 1 {Notify = TermA}
Step 63:
MGC to MG1

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1253 {
  Context = 1 {
  Subtract = TermA {Audit{ }},
  Subtract = EphD {Audit{Statistics}}
  }
}

The MG subtracts the two terminations from the context. The context itself is deleted with the "subtract" of the last termination from it. The MG1 responds to this transaction from MGC with statistics on ephemeral termination.

Step 64
MG1 to MGC:

MEGACO/1 [209.110.59.34]:25000
Reply = 1253 {
  Context = 1 {
  Subtract = TermA
  Subtract = EphD {
  Statistics {
  rtp/ps=1234, ; packets sent
  nt/os=56789, ; octets sent
  rtp/pr=987, ; packets received
  nt/or=65432, ; octets received
  rtp/pl=10, ; % packets lost
  rtp/jit=30,
  rtp/delay=30 ; average latency
  }
  }
  }
}

10.0 SDP for ATM

10.1 This section illustrates the usage of SDP for ATM. The draft "ATM SDP" [ref] is taken as reference. The two call establishment methods namely "Forward Bearer Connection Set-up model" and the "Backward Connection Set-up model" are considered. In the first method, the ATM connection establishment is initiated by the Originating Media Gateway whereas in the second method the Terminating Media Gateway initiates the ATM connection establishment.

The SDP attribute "eecid" is used in both the methods. The "eecid" is a means of correlating service-level connections with underlying ATM bearer connections.

In this example we assume that the Originating Media Gateway is
controlled by MGC1 and terminating Gateway controlled by MGC2. The communication protocol between the two MGC’s is out-of-scope of this draft.

10.1.1 Forward Bearer Connection Set-up Method

In this call scenario we assume that the RGW1 and RGW2 are having ATM connectivity towards the PDN network. RGW1 is controlled by MGC1 and RGW2 is controlled by MGC2. The call is initiated by user connected to the RGW1.

<table>
<thead>
<tr>
<th>RGW1</th>
<th>MGC1</th>
<th>MGC2</th>
<th>RGW2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RGW1</td>
<td>MGC1</td>
</tr>
<tr>
<td>________________</td>
<td>________________</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----------------</td>
<td></td>
</tr>
<tr>
<td>Initial Modify</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>Modify Resp</td>
<td></td>
</tr>
<tr>
<td>&lt;---------------</td>
<td></td>
</tr>
<tr>
<td>OffHook</td>
<td>----------------&gt;</td>
</tr>
<tr>
<td>Notify OffHook</td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Notify Resp</td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Dial Tone</td>
<td>&lt;--------------</td>
</tr>
<tr>
<td>Notify Digits</td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Notify Resp</td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Modify PhyB</td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Add PhyA SD:cg/rt</td>
<td>Add $ Local Unspecified</td>
</tr>
<tr>
<td>&lt;----------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Add PhyA Resp</td>
<td>Add EphA Resp</td>
</tr>
<tr>
<td>&lt;---------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OffHook</td>
<td>Notify OffHook</td>
</tr>
<tr>
<td>&lt;--------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Notify Resp</td>
<td>MGC-MGC SDPexchange</td>
</tr>
<tr>
<td>&lt;--------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>

Madhubabu, et al. [Page 244]
In Step 1 the MGCl generates the Modify message towards the RGW1 and similarly MGCl generates Modify message towards the RGW2 to check for off hook on the terminations. (A wildcard command may also be used in this scenario but for simplicity we consider only command to specific terminations). Modify message generated only for Residential gateway 1 is shown, similar message is sent to the other Residential gateway also. We are not considering the embedded signal and event descriptors here. The MGCl in NULL context generates the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified here in the example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control descriptor.
Step 1  
MG1 to RGW1: 

MEGACO/1 [216.33.33.61]: 27000  
Transaction = 1234 {
  Context = - {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = Receiveonly}
      },
      Events = 1111 {al/of}
    }
  }
}

MG after receiving the command from MG1 accepts it and responds with the transaction reply.

Step 2  
MG1 to MG2:  

MEGACO/1 [209.110.59.34]: 25000  
Reply = 1234 {
  Context = - (Modify = TermA)
}

In this example User A goes off hook. This event is detected by the RGW1 and constructs and sends the Notify message towards the MG1. The MG uses the same request ID (1111) sent by the MG1 in its initial command. The timestamp of the detected event is also passed as parameter to the observed event.

Step 3  
MG1 to MG1:  

MEGACO/1 [209.110.59.34]: 25000  
Transaction = 2000 {
  Context = - {
    Notify = TermA {ObservedEvents = 1111 {
      20010202T10000000:al/of}}
  }
}

MG1 generates the Notify response and responds with more messages towards the MG that generated the Notify command.

Step 4  
MG1 to MG1:  

MEGACO/1 [216.33.33.61]: 27000  
Reply = 2000 {
  Context = - (Notify = TermA)
}
The MGC1 in the present example issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial pattern is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 5
MGC1 to MG1:

    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1235 {
        Context = - {
            Modify = TermA {
                Signals = cg/dt,
                DigitMap = Dmap1 {{2XXX}}
                Events = 1112 {
                    al/on, dd/ce {DigitMap=Dmap1}
                }
            }
        }
    }

MG after receiving the Modify command after validation responds to the MGC1 and starts processing the descriptors listed.

Step 6
MG1 to MGC1:

    MEGACO/1 [209.110.59.34]: 25000
    Reply = 1235 {
        Context = - {Modify = TermA}
    }

The descriptors are processed in the order that is specified by the MGC1. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the termination. The Digit map said to be ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself contains a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. It’s assumed that the user dials digits that match the pattern specified in the digit map. Lets assume that the user has dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is
generated from MG1 to MGC1. The MG again used the same request identifier as specified by the MGC1.

Step 7
MG1 to MGC1:
MEGACO/1 [209.110.59.34]: 25000
  Transaction = 2001 {
    Context = - {
      Notify = TermA {ObservedEvents =112 {
        20010202T10010000:dd/ce {ds="2992", Meth=FM}}
      }
    }
  }

MGC1 after receiving the Notify command responds back with the Notify response.

Step 8
MGC1 to MG1:
MEGACO/1 [216.33.33.61]: 27000
  Reply = 2001 {
    Context = - {Notify = TermA}
  }

MGC after receiving the Notify command starts analyzing the dialed digits. In this example it is assumed that the called subscriber is connected to the RGW2, which is controlled by MGC2. The MGC1 communicates to MGC2 that there is a call initiation for UserB.

The MGC1 generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created because of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used. The MGC1 thus initiates service-level call establishment by sending the following control message to the RGW1.

Step 9
MGC to MG1:
MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1236 {
    Context = $ {
      Add = TermA {
        Signals {cg/rt}
      }
      Add = $ {
        Media {
          LocalControl {
        }
      }
    }
  }

Madhubabu, et al.
In this example the connection field specifies only the connection type (ATM). The other fields are unspecified as they are not needed. The RGW1 in its response indicates its NSAP address. In this example the RGW1 creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA.

Step 10
RGW1 to MGC1:
  MEGACO/1 [209.110.59.34]: 25000
  Reply = 1236 {
    Context = 1 {
      Add = TermA,
      Add=EphA{
        Media {
          Local {
            v=0
            o=- 2873397496 0 ATM - -
            s=-
            c=ATM NSAP
            t=0 0 m=audio $ - -
              }
          }
        }
      }
    }
  }
}

MGC2 after receiving the message from MGC1 generates an Add command towards the RGW2. The ContextID specified in the action is $. The first
command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MGC2 has the local information for the ephemeral termination EphA in the RGW1. This information is passed as remote information to the RGW2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MGC2 to RGW2:

```
MEGACO/1 [216.33.33.62]:27000
Transaction = 1237 {
    Context = $ {
        Add = TermB { Media {
            LocalControl {Mode = Receiveonly} },
            Signals {al/ri}
            Events =1234{al/of {events = 1235 {al/on}}},
        },
        Add = $ {Media {
            LocalControl {
                Mode = sendrecv,
            },
            Local {
                v=0
                o=- 2873397497 0 ATM - -
                s=-
                c=ATM - -
                t=0 0
                m=audio $ - -
            },
            Remote {
                v=0
                o=- 2873397496 0 ATM
                NSAP 47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
                s=-
                c=ATM NSAP
                47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
                t=0 0 m=audio - AAL2/ITU 8
            }
        }
    }
}
```

RGW2 after receiving the new transaction from MGC2 starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor
specified in the command. The signal descriptor lists "ring" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates an ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC2. The RGW2 allocates the necessary resources for processing the media descriptor for the ephemeral termination. The RGW2 responds to the MGC2 by providing an SDP descriptor with a locally assigned "eecid". The MG2 responds to MGC with the following transaction reply.

Step 12
RGW2 to MGC2:
  MEGACO/1 [207.176.47.90]: 26000
  Reply = 1237 {
    Context = 2 {
      Add = TermB,
      Add = EphB{
        Media {
          Local {
            
            o= 2873397714 0 ATM
            NSAP 47.0091.8100.0000.0040.2A74.EB03.0020.4421.2A04.00
            s= -
            c= ATM NSAP
              47.0091.8100.0000.0040.2A74.EB03.0020.4421.2A04.00
            t= 0 0
            m= audio - AAL2/ITU 8
            a= eecid:B3D58E32
          }
        }
      }
    }
  }

The MGC2 waits for the UserB to go offhook. Once the UserB goes offhook, RGW2 reports the notification of the offhook event to the MGC2.

Step 13
RGW2 to MGC2:
  MEGACO/1 [207.176.47.90]: 26000
  Transaction = 3000 {
    Notify = TermB {ObservedEvents =1234 {
      20000202T10020000:al/of}}
  }

The MGC2 responds to the RGW2 with the Notify response.

Step 14
MGC2 to RGW2:
  MEGACO/1 [216.33.33.60]: 27000
  Reply = 3000 {

The MGC2 generates a transaction towards RGW2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook event.

Step 15:
MGC to MG2:

```
MEGACO/1 [216.33.33.60]: 27000
Transaction = 1238 {
   Context = 2 {
      Modify = TermB {
         Signals { } ; to turn off ringing
         Events = 1235 {al/on},
         Media {
            LocalControl {
               Mode = SendRecv,
            }
         }
      }
      Modify = EphB{
         Media {
            LocalControl {
               Mode = SendRecv,
            }
         }
      }
   }
}```

The MG2 responds to the request from MGC.

Step 16
RGW2 to MGC2:

```
MEGACO/1 [207.176.47.90]: 26000
Reply = 1238 {
   Context = 2 {Modify = TermB , Modify = EphB}
}```

The MGC2 generates message to the MGC1 to indicate the "eecid" generated by the RGW2. The MGC1 generates a Modify message towards the RGW1 to stop the ringback tone and to report the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MGC1 to RGW1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239 {
   Context = 1 {
```

Madhubabu, et al.                                            [Page 252]
Modify = TermA {
    Media {
        LocalControl {
            Mode = sendrecv
        }
        Signals {
        }
    }
}
Modify = EphA {
    Media {
        LocalControl {
            Mode = sendrecv
            Local {
                v=0
                o=- 2873397874 0 ATM - -
                s=-
                c=ATM - -
                t=0 0
                m=audio $ - -
                a=BearerType:SVC on
            }
            Remote {
                v=0
                o=- 2873397714 0 ATM
                c=ATM NSAP 47.0091.8100.0000.0040.2A74.EB03.0020.4421.2A04.00
                s=-
                t=0 0
                m=audio $ AAL2/ITU 8
                a=eecid:B3D58E32
            }
        }
    }
}
The empty signal descriptor in the Modify command for termination TermA, specifies to stop the ringback tone at the calling end. The remote SDP information specifies the "eecid" to be used for the ephemeral termination EphA. The RGW1 using this "eecid" initiates the connection establishment. The Local SDP information with the media attribute "BearerType" indicates that an SVC is to be used and that the <localInitiation> flag is on i.e. the SVC is to be set up by the terminating Media Gateway. The mode is changed to send receive. RGW1 responds to the MGC with the response for the Modify commands.

Step 18
MG1 to MGC:
    MEGACO/1 [209.110.59.34]: 25000
The two users can exchange the media. The call can be termination either by the calling user or the called user. In this example it is assumed that the calling party has gone on-hook. The UserA after the conversation goes onhook indicating the tearing down of the call. The same is reported in the Notify command from RGW1 to MGC1.

Step 19
RGW1 to MGC1:

\[
\text{MEGACO/1 [209.110.59.34]:25000} \\
\text{Transaction = 2002 { } } \\
\text{Context = 1 { } } \\
\text{Notify = TermA {ObservedEvents =1112 { } } } \\
\text{20010202T10030000:a1/on} \\
\}
\]

The MGC1 responds to the MG1's Notify message.

Step 20
MGC1 to RGW1:

\[
\text{MEGACO/1 [216.33.33.61]:27000} \\
\text{Reply = 2002 { } } \\
\text{Context = 1 { } } \\
\text{Notify = TermA} \\
\}
\]

The MGC2 after receiving the information from MGC1, generates a Modify command towards the RGW2 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 21
MGC2 to RGW2:

\[
\text{MEGACO/1 [216.33.33.60]: 27000} \\
\text{Transaction = 1240 { } } \\
\text{Context = 2 { } } \\
\text{Modify = TermB { } } \\
\text{Signals {cg/bt} } \\
\text{Media { } } \\
\text{LocalControl { } } \\
\text{Mode = recvonly} \\
\}}, \\
\text{Modify = EphB { } } \\
\text{Media { } } \\
\text{LocalControl { } } \\
\text{Mode = recvonly} \\
\}
\]
The RGW2 responds to this modify request.

Step 22
RGW2 to MGC2:
   MEGACO/1 [207.176.47.90]: 26000
   Reply = 1240 {
      Context = 2 {
         Modify= TermB, Modify = EphB
      }
   }

The MGC1 generates transactions with two subtracts commands one for physical and other for ephemeral terminations.

Step 23:
MGC1 to RGW1
   MEGACO/1 [216.33.33.61]: 27000
   Transaction = 1241 {
      Context = 1 {
         Subtract = TermA {Audit{ }},
         Subtract = EphA {Audit(Statistics)}
      }
   }

The RGW1 subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The RGW1 responds to this transaction from MGC1 with statistics on ephemeral termination.

Step 24
RGW1 to MGC1:
   MEGACO/1 [209.110.59.34]:25000
   Reply = 1241 {
      Context = 1 {
         Subtract = TermA
         Subtract = EphA {
            Statistics {
               rtp/ps=1234, ; packets sent
               nt/os=56789, ; octets sent
               rtp/pr=987, ; packets received
               nt/or=65432, ; octets received
               rtp/pl=10, ; % packets lost
               rtp/jit=30,
               rtp/delay=30 ; average latency
            }
         }
      }
   }
The User B after going onhook, the RGW2 generates Notify command towards the MGC2.

Step 25
RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000
Transaction = 3001 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1235 {
      20000202T10070000:al/on}}
  }
}

The MGC2 responds to the RGW2 with the Notify response.

Step 26
MGC2 to RGW2:

MEGACO/1 [216.33.33.61]: 27000
Reply = 3001 {
  Context = 2 {Notify = TermB}
}

The MGC2 generates subtract command towards RGW2 for removing TermB from valid context.

Step 26
MGC2 to RGW2:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1242 {
  Context = 2 {
    Subtract = TermB {Audit{ }},
    Subtract = EphB {Audit{Statistics}}
  }
}

The RGW2 responds to the subtract commands generated by MGC2.

Step 27
RGW2 to MGC2:

MEGACO/1 [207.176.47.89]:26000
Reply = 1242 {
  Context = 2 {
    Subtract = TermB
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}
The two users UserA and UserB are ready for initiating/receiving further calls.

### 10.1.2 Backward Bearer Connection Set-up Method

In this call scenario we assume that the RGW1 and RGW2 are having ATM connectivity towards the PDN network. RGW1 is controlled by MGC1 and RGW2 is controlled by MGC2. The call is initiated by user connected to the RGW1.

<table>
<thead>
<tr>
<th>RGW1</th>
<th>MGC1</th>
<th>MGC2</th>
<th>RGW2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial Modify</td>
<td>Initial Modify</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Resp</td>
<td>Modify Resp</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>---------------</td>
<td>-----</td>
</tr>
<tr>
<td>OffHook</td>
<td>Notify OffHook</td>
<td>Notify Resp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notify Resp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dial Tone</td>
<td>Notify Digits</td>
<td>Notify Resp</td>
<td>MGC-MGC protocol</td>
</tr>
<tr>
<td></td>
<td>Notify Resp</td>
<td></td>
<td>Modify PhyB</td>
</tr>
<tr>
<td></td>
<td>Modify Resp</td>
<td></td>
<td>Ring</td>
</tr>
<tr>
<td></td>
<td>Notify OffHook</td>
<td></td>
<td>MGC-MGC Ring indication</td>
</tr>
<tr>
<td></td>
<td>Add PhyA SD:cg/rt</td>
<td>Add PhyA Resp</td>
<td>Add EphA Resp</td>
</tr>
<tr>
<td></td>
<td>Add $ Local Unspecified</td>
<td></td>
<td>Notify OffHook</td>
</tr>
</tbody>
</table>

Madhubabu, et al. [Page 257]
In Step 1 the MGC1 generates the Modify message towards the RGW1 and similarly MGC2 generates Modify message towards the RGW2 to check for off hook on the terminations. (A wildcard command may also be used in this scenario but for simplicity we consider only command to specific terminations). Modify message generated only for Residential gateway 1 is shown, similar message is sent to the other Residential gateway also. We are not considering the embedded signal and event descriptors here. The MGC in NULL context generates the command to the specific termination TermA. The off hook event of the analog supervision package is used here. The request identifier specified here in the example is 1111. The mode of the termination is set to receive only. The stream parameter is used with only the Local control.
Internet-Draft      Megaco/H.248 Call flow examples       July 2001

descriptor.

Step 1
MGC1 to RGW1:
    MEGACO/1 [216.33.33.61]: 27000
    Transaction = 1234 {
        Context = - {
            Modify = TermA {
                Media {
                    LocalControl {
                        Mode = Receiveonly
                    },
                    Events = 1111 {al/of}
                }
            }
        }
    }

MG after receiving the command from MGC1 accepts it and responds with the transaction reply.

Step 2

MG1 to MGC2:
    MEGACO/1 [209.110.59.34]: 25000
    Reply = 1234 {
        Context = - (Modify = TermA)
    }

In this example User A goes off hook. This event is detected by the RGW1 and constructs and sends the Notify message towards the MGC1. The MG uses the same request id (1111) sent by the MGC1 in its initial command. The timestamp of the detected event is also passed as parameter to the observed event.

Step 3
MG1 to MGC1:
    MEGACO/1 [209.110.59.34]: 25000
    Transaction = 2000 {
        Context = - {
            Notify = TermA {ObservedEvents =1111 {
                20010202T10000000:al/of}}
        }
    }

MGC1 generates the Notify response and responds with more messages towards the MG that generated the Notify command.

Step 4
MGC1 to MG1:
    MEGACO/1 [216.33.33.61]: 27000
    Reply = 2000 {
        Context = - (Notify = TermA)
The MGC1 in the present example issues a MODIFY command. The Modify command contains a signal descriptor for the application of dial tone to the user. The digit map descriptor here is used to configure a digit map on the termination. The digit map name used in the example is Dmap1 and the dial pattern is 2XXX. The event descriptor lists digit map completion event of the DTMF detection package and onhook of the analog line supervision package. The request id specified in the event descriptor is 1112.

Step 5
MGC1 to MG1:

MEGACO/1 [216.33.33.61]: 27000
Transaction = 1235 {
  Context = - {
    Modify = TermA {
      Signals {cg/dt},
      DigitMap = Dmap1 { (2XXX) }
      Events = 1112 {
        al/on, dd/ce {DigitMap = Dmap1 }},
      }
    }
  }
}

MG after receiving the Modify command after validation responds to the MGC1 and starts processing the descriptors listed.

Step 6
MG1 to MGC1:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1235 {
  Context = - {Modify = TermA}
}

The descriptors are processed in the order that is specified by the MGC1. In this example the order of descriptor is signal descriptor, digit map descriptor followed by Events descriptor. The MG first processes the signal descriptor. The dial tone is applied to the Termination specified. The Digit map is updated in the Database of the termination. The Digit map said to be ACTIVE on the termination as the digit map completion event is listed in the events descriptor with the digit map name. A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, and that event descriptor contains a digit map completion event which itself contains a digit map parameter. UserA after receiving the dial tone starts dialing digits. In this example we will not dwell into the different possible cases of digit dialing by the user. It’s assumed that the user dials digits that match the pattern specified in the digit map. Let’s assume that the user has
dialed 2992. MG detects the digits dialed and reports the same as parameter to the digit map completion event. A notify command is generated from MG1 to MGC1. The MG again used the same request identifier as specified by the MGC1.

Step 7
MG1 to MGC1:
MEGACO/1 [209.110.59.34]: 25000
Transaction = 2001 {
  Context = - {
    Notify = TermA {ObservedEvents =1112 {
      20010202T10010000:dd/ce {ds="2992", Meth=FM})
    }
  }
}

MGC1 after receiving the Notify command responds back with the Notify response.

Step 8
MGC1 to MG1:
  MEGACO/1 [216.33.33.61]: 27000
  Reply = 2001 {
    Context = - {Notify = TermA}
  }

MGC after receiving the Notify command starts analyzing the dialed digits. In this example it is assumed that the called subscriber is connected to the RGW2, which is controlled by MGC2. The MGC1 communicates to MGC2 that there is a call initiation for UserB.

The MGC1 generates a transaction with two commands clubbed into the same Action. The first command is to create a new context and add the physical termination TermA into it. The second command is generated to create an ephemeral termination and add the created termination in the same context that was created because of the earlier command. Here we assumed a single set of SDP information indicating that Reserve group is not used. The Reserve Value feature is also not used. The MGC1 thus initiates service-level call establishment by sending the following control message to the RGW1.

Step 9
MGC to MG1:
  MEGACO/1 [216.33.33.61]: 27000
  Transaction = 1236 {
    Context = $ {
      Add = TermA {
        Signals {cg/rt}
      }
    }
    Add = $ {
      Media {
        ...
In this example the connection field specifies only the connection type (ATM). The other fields are unspecified as they are not needed. The RGW1 in its response indicates its NSAP address. In this example the RGW1 creates a context with contextID 1. The physical termination TermA is added to context 1. The mode of the physical termination was earlier set to Receiveonly and in this message the ephemeral termination is requested to create with Receiveonly mode. The ephemeral termination created in this example is EphA. The RGW1 in its Local SDP information specifies the "eecid" that needs to be used by the other RGW for establishing an ATM SVC between RGW1 and RGW2.

Step 10
RGW1 to MGC1:

MEGACO/1 [209.110.59.34]: 25000
Reply = 1236 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          v=0
          o=- 2873397496 0 ATM NSAP 47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
          s=-
          c=ATM NSAP 47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
          t=0 0
          m=audio - AAL2/ITU 8
          a=eecid:b3d58e32
        }
      }
    }
  }
}
MG2 after receiving the SDP information from MG1 generates an Add command towards the RG2. The ContextID specified in the action is $. The first command adds the physical termination TermB to the newly created context. The Signal descriptor for this termination lists the ring signal of the analog line supervision package. This alerting signal is applied to the termination of the TermB. The Event descriptor specifies offhook event of the analog line supervision package. The second Add is meant to create an ephemeral termination. MG2 has the local information for the ephemeral termination EphA in the RG1. This information is passed as remote information to the RG2. The new ephemeral termination that will be created will take these parameters as the remote SDP information.

Step 11
MG2 to RG2:

```plaintext
MEGACO/1 [216.33.33.62]:27000
Transaction = 1237 {
  Context = $ {
    Add = TermB { Media {
      LocalControl {Mode = Receiveonly} },
      Signals {al/ri} EVENTS =1234{al/of {events = 1235 {al/on}}},
    },
    Add = $ {Media {
      LocalControl {
        Mode = sendrecv,
      },
      Local {
        v=0
        o=- 2873397497 0 ATM - -
        s=-
        c=ATM - -
        t=0 0
        m=audio $ - -
        a= bearerType:SVC on
      },
      Remote {
        v=0
        o=- 2873397496 0 ATM
        NSAP 47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
        s=-
        c=ATM NSAP
        47.0091.8100.0000.0060.3E64.FD01.0060.3E64.FD01.00
        t=0 0 m=audio - AAL2/ITU 8
        a=eedid:b3d58e32
      } }
  }
}
```

Madhubabu, et al.
RGW2 after receiving the new transaction from MGC2 starts processing it. It creates a new context with contextID 2. It adds the physical termination TermB to that context and start processing the descriptor specified in the command. The signal descriptor lists "ring" signal to be applied on the termination. The event descriptor lists the off hook event. The RGW2 creates an ephemeral termination with TerminationId EphB. The local information is under-specified from the MGC2. The RGW2 allocates the necessary resources for processing the media descriptor for the ephemeral termination. The RGW2 creates an SVC connection using the "eecid" specified in the remote descriptor. The RGW2 as indicated by the MGC that an SVC is to be used as the <localInitiation> flag is on in the Local SDP information. The RGW2 responds to MGC2 with the following transaction reply.

Step 12
RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000
Reply = 1237 {
  Context = 2 {
    Add = TermB,
    Add = EphB {
      Media {
        Local { v=0
          o=- 2873397714 0 ATM
          NSAP 47.0091.8100.0000.0040.2A74.EB03.0020.4421.2A04.00
          s=- ATM NSAP 47.0091.8100.0000.0040.2A74.EB03.0020.4421.2A04.00
          t=0 0
          m=audio - AAL2/ITU 8
        }
      }
    }
  }
}

The MGC2 waits for the UserB to go offhook. Once the UserB goes offhook, RGW2 reports the notification of the offhook event to the MGC2.

Step 13
RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000
Transaction = 3000 {
  Context = 2 {
    Notify = TermB {ObservedEvents =1234 {
      20000202T10020000:al/of}}
  }
}
The MGC2 responds to the RGW2 with the Notify response.

Step 14
MGC2 to RGW2:

MEGACO/1 [216.33.33.60]: 27000
Reply = 3000 {
  Context = 2 {Notify = TermB}
}

The MGC2 generates a transaction towards RGW2 with two commands in one action. It changes the mode of both the terminations to sendrecv. The Signal descriptor of the Modify command for the first termination, stops the ring signal already applied on the termination and the event descriptor lists the onhook event.

Step 15:
MGC to MG2:

MEGACO/1 [216.33.33.60]: 27000
Transaction = 1238 {
  Context = 2 {
    Modify = TermB {
      Signals { }; to turn off ringing
      Events = 1235 {al/on},
      Media {
        LocalControl {
          Mode = SendRecv,
        }
      }
    }
    Modify = EphB {
      Media {
        LocalControl {
          Mode = SendRecv,
        }
      }
    }
  }
}

The MG2 responds to the request from MGC.

Step 16
RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000
Reply = 1238 {
  Context = 2 {Modify = TermB, Modify = EphB}
}

The MGC2 generates message to the MGC1 to indicate using the "eecid" sent earlier the SVC is created by the RGW2. The MGC1 generates a Modify message towards the RGW1 to stop the ringback tone and to report
the remote SDP information for the ephemeral termination EphA. The mode of the two terminations TermA and EphA is set to send receive.

Step 17
MGC1 to RGW1:

```
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1239 {
  Context = 1 {
    Modify = TermA {
      Media {
        LocalControl {
          Mode = sendrecv
        }
      }
    }
    Signals { }
  },
  Modify = EphA {
    Media {
      LocalControl {
        Mode = sendrecv
      }
    }
  }
}
```
The empty signal descriptor in the Modify command for termination TermA, specifies to stop the ringback tone at the calling end. The mode is changed to send receive. RGW1 responds to the MGC1 with the response for the Modify commands.

Step 18
MG1 to MGC:

```
MEGACO/1 [209.110.59.34]: 25000
Reply = 1239 {
  Context = 1 {Modify = TermA, Modify = EphA}
}
```
The two users can exchange the media. The call can be termination either by the calling user or the called user. In this example it is assumed that the calling party has gone on-hook The UserA after the conversation goes onhook indicating the tearing down of the call. The same is reported in the Notify command from RGW1 to MGC1.

Step 19
RGW1 to MGC1:

```
MEGACO/1 [209.110.59.34]:25000
Transaction = 2002 {
  Context = 1 {
    Notify = TermA {ObservedEvents =1112 {
      20010202T10030000:al/on}
  }
}
```

Madhubabu, et al.                                             [Page 266]
The MGC1 responds to the MG1s Notify message.

Step 20
MGC1 to RGW1:

MEGACO/1 [216.33.33.61]:27000
Reply = 2002 {
  Context = 1 {
    Notify = TermA
  }
}

The MGC2 after receiving the information from MGC1, generates a Modify command towards the RGW2 for applying busy tone to the called subscriber. The mode of both terminations is set to receive only.

Step 21
MGC2 to RGW2:

MEGACO/1 [216.33.33.60]: 27000
Transaction = 1240 {
  Context = 2 {
    Modify = TermB {
      Signals {cg/bt}
      Media {
        LocalControl {
          Mode = recvonly}
      }
    },
    Modify = EphB {
      Media {
        LocalControl {
          Mode = recvonly}
      }
    }
  }
}

The RGW2 responds to this modify request.

Step 22
RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000
Reply = 1240 {
  Context = 2 {
    Modify= TermB, Modify = EphB
  }
}

The MGC1 generates transactions with two subtracts commands one for physical and other for ephemeral terminations.

Step 23:
MGC1 to RGW1

MEGACO/1 [216.33.33.61]: 27000

Transaction = 1241 {
  Context = 1 {
    Subtract = TermA {Audit{ }},
    Subtract = EphA {Audit(Statistics)}
  }
}

The RGW1 subtracts the two terminations from the context. The context itself is deleted with the subtract of the last termination from it. The RGW1 responds to this transaction from MGC1 with statistics on ephemeral termination.

Step 24

RGW1 to MGC1:

MEGACO/1 [209.110.59.34]: 25000

Reply = 1241 {
  Context = 1 {
    Subtract = TermA
    Subtract = EphA {
      Statistics {
        rtp/s=1234; packets sent
        nt/os=56789; octets sent
        rtp/pr=987; packets received
        nt/or=65432; octets received
        rtp/pl=10; % packets lost
        rtp/jit=30,
        rtp/delay=30; average latency
      }
    }
  }
}

The User B after going onhook, the RGW2 generates Notify command towards the MGC2.

Step 25

RGW2 to MGC2:

MEGACO/1 [207.176.47.90]: 26000

Transaction = 3001 {
  Context = 2 {
    Notify = TermB {ObservedEvents = 1235 {
      20000202T10070000:al/on}}
  }
}

The MGC2 responds to the RGW2 with the Notify response.

Step 26

MGC2 to RGW2:

MEGACO/1 [216.33.33.60]: 27000

Reply = 3001 {
Context = 2 {Notify = TermB}

The MGC2 generates subtract command towards RGW2 for removing TermB from valid context.

Step 26
MGC2 to RGW2:
MEGACO/1 [216.33.33.61]: 27000
Transaction = 1242 {
  Context = 2 {
    Subtract = TermB {Audit{ }},
    Subtract = EphB {Audit{Statistics}}
  }
}
The RGW2 responds to the subtract commands generated by MGC2.

Step 27
RGW2 to MGC2:
MEGACO/1 [207.176.47.89]: 26000
Reply = 1242 {
  Context = 2 {
    Subtract = TermB
    Subtract = EphB {
      Statistics {
        rtp/ps=987, ; packets sent
        nt/os=65432, ; octets sent
        rtp/pr=1234, ; packets received
        nt/or=56789, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=30,
        rtp/delay=30 ; average latency
      }
    }
  }
}
The two users UserA and UserB are ready for initiating/receiving further calls.

Acknowledgements:
The authors wish to thank several colleagues at Kenetec and in the industry who have contributed towards the development of these call flow document and who have reviewed and sent valuable technical comments. Special thanks to Mathi Packiam for providing timely comments. Basic call flows have been provided by the authors of the MGCP call flow document. Technical ideas/comments have been provided by Thillaivilagam Anand, Sarika Gupta, Mehul Shah, Peter Michielsen, Terry L Anderson and Sasitharan R.