Megaco and SIP Internetworking

Salman Abdul Baset and Zahid Anwar

Table of Contents

1. ABSTRACT 3

2. ABBREVIATIONS 3

3. INTRODUCTION 4

4. PREVIOUS WORK 4

5. SIP FOR ESTABLISHING CALLS BETWEEN MGS ON DIFFERENT DOMAINS 5
- 5.1 Detailed Call Flow 6
- 5.2 Call-Flow Description 7
  - 5.2.1 Registration 7
  - 5.2.2 Call 7
  - 5.2.3 Negotiation of SDP 7
  - 5.2.4 Media Flow 7
  - 5.2.5 Call Tear Down 7
- 5.3 Conclusions and Observations 7

6. SIP FOR MGC HAND OFF PROCEDURE 9
- 6.1 MGC Intensive: 9
- 6.2 Client Intensive: 10

7. SIP FOR MEGACO STATE TRANSFER 11

8. APPENDEX A 12

9. DISCLAIMER 20

10. REFERENCES 21

1. Abstract

Megaco[1] is a master slave protocol and is used to establish, and terminate calls across terminations (end points) connected to Media Gateways (MGs). A Media Gateway Controller (MGC) controls the Media Gateway(s). MGC controlling subordinate MGs constitute a ‘calling domain’. Megaco due to its master slave nature does not describe the establishment of calls across domains or across MGCs. SIP[2] has been envisioned as a messaging protocol across MGCs. This document describes Megaco-SIP call flows, and illustrates certain call-procedures mapping.

2. Abbreviations Used

This recommendation defines the following terms.

GW	Gateway
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
MG	Media Gateway
MGC	Media Gateway Controller
NM	Network Management
SIP	Session Initiation Protocol

3. Introduction

Megaco[1] (Media Gateway Control Protocol) exploded gatekeeper model of H.323 and put signaling control in a Media Gateway Controller (MGC) thereby unbundling call intelligence from media. Megaco instructs Media Gateways (MG) to connect streams coming from outside a packet network on to a packet stream such as RTP. Master-MGC issues commands to send and receive media from addresses, generate tones, and to modify configuration.

SIP[2] (Session Initiation Protocol) is a peer-to-peer protocol and is used for establishing multimedia sessions between different clients. SIP uses Proxy Servers to help route requests to the user's current location. SIP provides a registration function that allows users to upload their current locations for use by proxy servers.

Megaco is used for communication downward, to the media gateways and does not constitute a complete system. The architecture requires a session initiation protocol (Peer to Peer) for communication between MGCs. SIP is envisioned as a suitable protocol for Soft Switch to Soft Switch communication.

In this document we describe establishment and termination of calls using SIP across MGCs. We then discuss MGC HandOff procedure and elaborate its details. Finally we discuss the transfer of MGC state using SIP.

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 and SIP. It is not the intent of the document to exhaustively represent the inter-working functionality between these protocols; however, an attempt is made to depict its usage in such a case for the purpose of completeness in the larger perspective.

4. Previous Work

SIP-T[3] describes internetworking between SIP and PSTN Networks. It discusses carrying ISUP messages across SIP Networks. "Megaco/H.248 Call flow examples" [4] describes establishment and termination of a call from MGC to SIP User Agent. There is no detailed published material on the establishment and tear down of a call, and handling of different call-procedures across MGCs that use MEGACO to control the Media Gateways.

5. SIP for Establishing Calls between MGs on different Domains

Media Gateway 1 (MG1) is registered with Media Gateway Controller 1 (MGC1), which is in the domain Controller1.com. Media Gateway 2 is registered in a different domain (Controller 2) to Media Gateway Controller2.com. A termination (end point) on MG2 with alias 8687000 and Termination Id 0 wants to call a termination on MG2 with alias 97378000 and Termination Id 0 (See Fig 1).

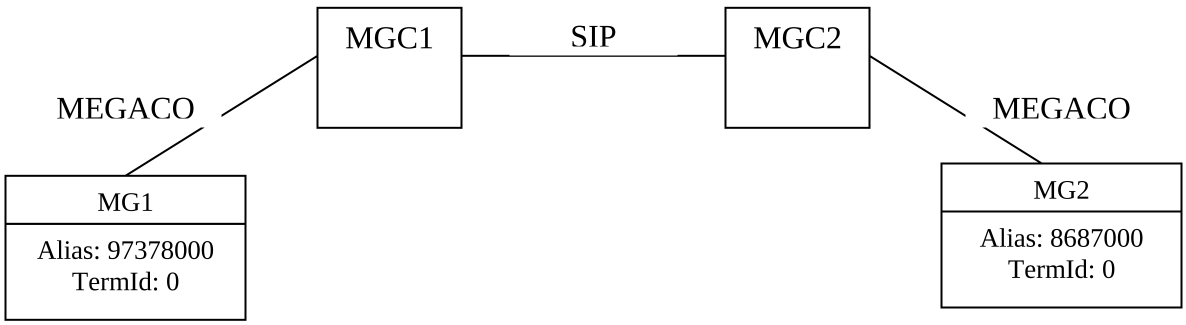
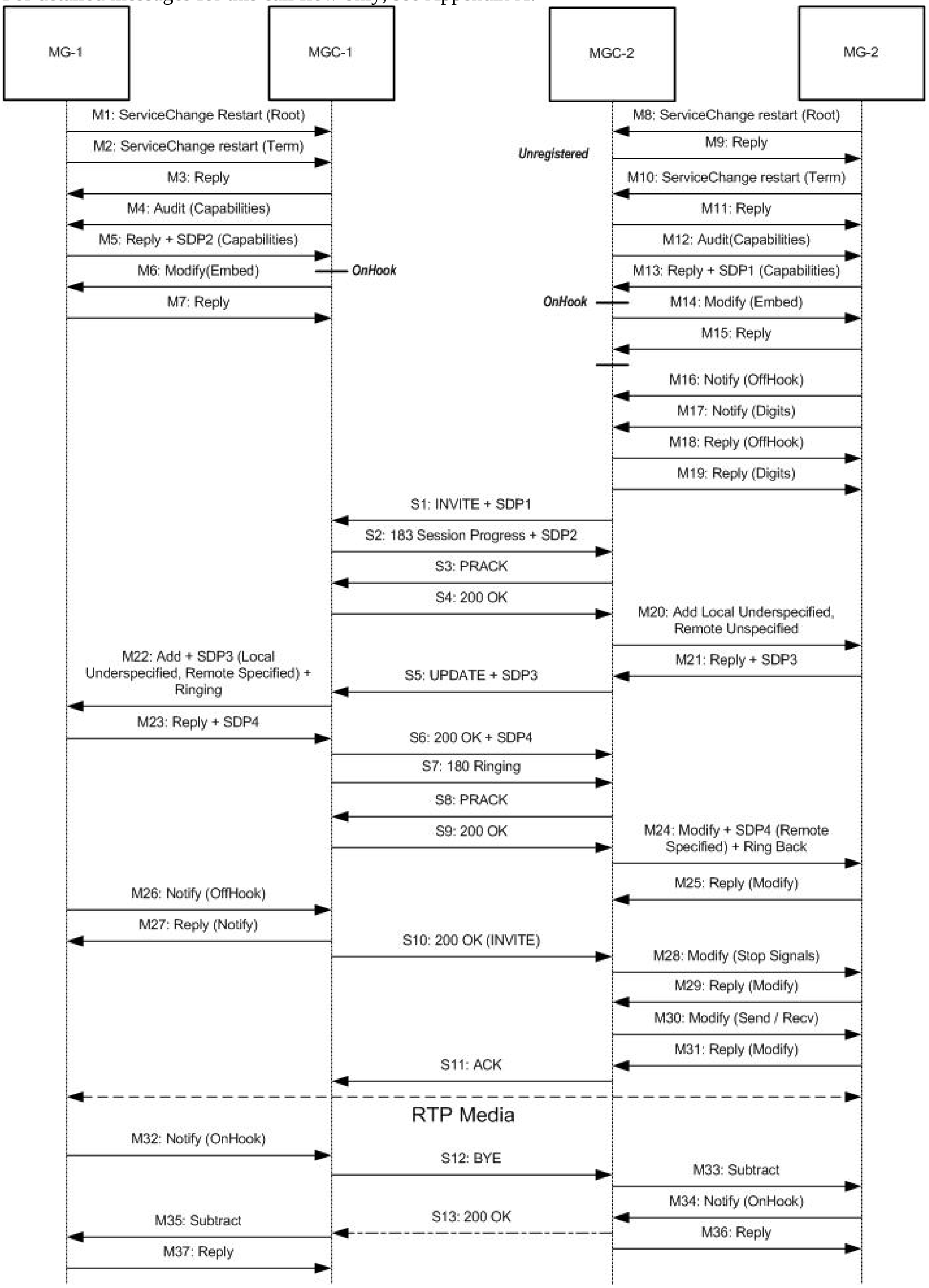


Figure 1

5.1 Detailed Call Flow

For detailed messages for this call flow only, see Appendix A.



5.2 Call-Flow Description

5.2.1 Registration

MG1 registers itself with MGC1 using the Service Change Restart Command. MGC1 accepts the Gateway and Audits its Capabilities. It then Modifies the terminations with an Embedded Event Descriptor and instructs it to detect an off hook event, play a dial tone on off-hook and then detect dial digits. The same procedure is followed between MG2 and MGC2. (See messages M1-M7 and M8-M15)

5.2.2 Call

Termination 0 on MG2 goes Off-Hook and calls Termination 0 on MG1. It notifies MGC2, which in turn determines (beyond the scope of this document) that the Gateway is not registered to it but to MGC1. It sends a SIP INVITE request to MGC1 along with the initially Audited Capabilities. It enforces in the INVITE that the provisional responses be delivered to it reliably[5]. MGC1 compares the capabilities received in INVITE with the audited capabilities of MG1 and computes a set of common capabilities, which it returns to MGC1 using SIP 183 Session Progress response. MGC2 acknowledges using PRACK.

5.2.3 Negotiation of SDP

MGC2 issues an Add command to MG2 under specifying the Local SDP sent by MGC1. MG2 creates an Ephemeral Termination, chooses an SDP (SDP3), and sends it in the Megaco Local Media Descriptor to inform the callee of its choosen capabilities as a reply to the Add command. MGC2 informs MGC1 of remote SDP through SIP UPDATE[6] request. MGC1 sends an Add Command to MG1 with the new SDP received. MG1 in turn chooses its Ephemeral Termination and Capabilities and returns these to MGC1. It also plays the Ringing tone on the Termination. MGC1 sends this SDP (SDP4) to MGC2 in a 200 OK response to UPDATE request. MGC1 also sends a 180 Ringing Response to MGC2. MGC2 after receiving remote SDP4 and 180 Ringing response modifies its MG2 and starts the Ring Back Tone.

5.2.4 Media Flow

When the User picks up the hand set in response to the Ringing tone, the callee is informed via the SIP 200 OK response to INVITE request. Note that no message body is present in the 200 OK response. MGC2 Modifies the Termination to stop the signals and changes the stream mode to Send and Receive. MGC1 is acknowledged via SIP ACK and RTP Media is started.

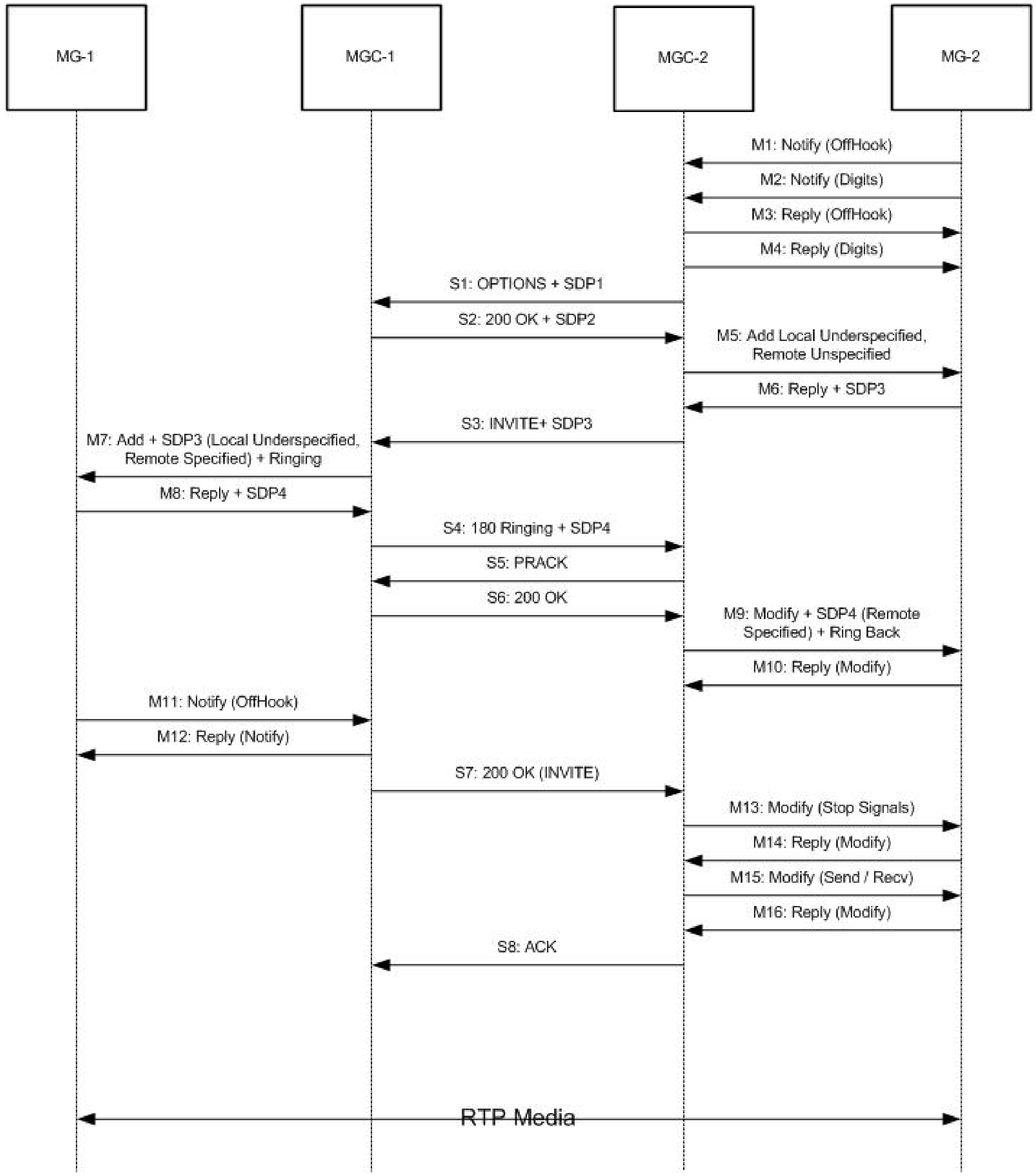
5.2.5 Call Tear Down

When either of the party decides to end the conversation the MGC is informed via the Notify OnHook command. The remote side is informed via SIP BYE request after which both MGCs proceed to delete the Ephemeral Terminations via the Subtract Command

5.3 Conclusions and Observations

The above explanation and call-flow successfully actualizes the envisioned usage of SIP as a messaging protocol between two MGCs.

Consider the Add Termination message (M20). This Megaco command can only be carried out when the presence of remote party is determined. Thus the main purpose of the first INVITE request is to determine the presence of remote party i.e. MGC1. This results in the exchange of 11 SIP messages between two MGCs. To reduce the message exchange, MGC1s presence can be determined by the OPTION request. This results in only 8 messages being exchanged across two MGCs. SIP message exchange can be further reduced if reliable delivery of provisional responses is not enforced.


MG-1
MGC-1
MGC-2
MG-2
M1: Notify (OffHook)
M2: Notify (Digits)
M3: Reply (OffHook)
M4: Reply (Digits)
S1: OPTIONS + SDP1
S2: 200 OK + SDP2
M5: Add Local Underspecified, Remote Unspecified
M6: Reply + SDP3
S3: INVITE+ SDP3
M7: Add + SDP3 (Local Underspecified, Remote Specified) + Ringing
M8: Reply + SDP4
S4: 180 Ringing + SDP4
S5: PRACK
S6: 200 OK
M9: Modify + SDP4 (Remote Specified) + Ring Back
M10: Reply (Modify)
M11: Notify (OffHook)
M12: Reply (Notify)
S7: 200 OK (INVITE)
M13: Modify (Stop Signals)
M14: Reply (Modify)
M15: Modify (Send / Recv)
M16: Reply (Modify)
S8: ACK
RTP Media

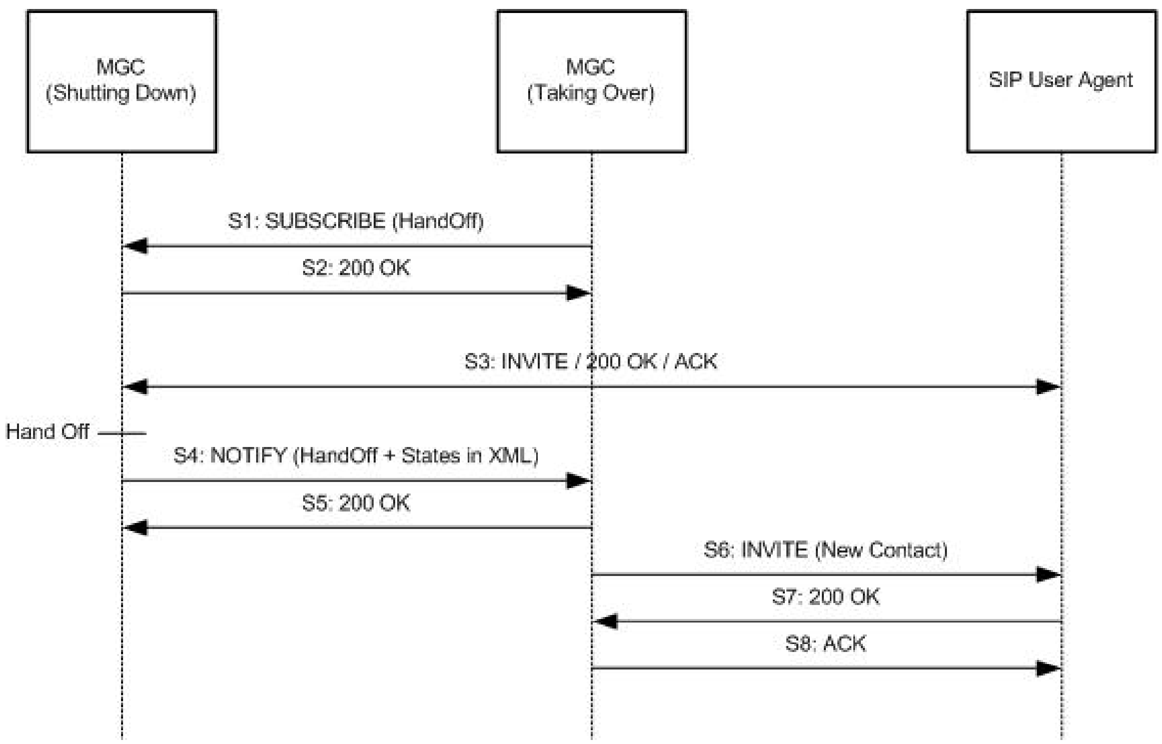
6. SIP for MGC Hand Off Procedure

The Megaco protocol provides a feature namely that an MGC that is shutting down because of failure or maintainance requests its subordinate MGs to register themselves with another MGC. To accomplish such a procedure the MGC which is shutting down informs its MGs through a Megaco Service Change Command with Handoff in the Reason Parameter. As a result of this message the MGs register themselves with the secondary MGC which in turn Audits the terminations to determine their states. This transfer of states is transparent and the terminations remain in their previous states (whether its in Offhook, Onhook, Ringing or In call). An interesting scenario occurs when an MG termination is in call with a SIP User Agent. A mechanism to inform the new MGC that a MG termination is in a SIP Call is needed. Two mechanisms are suggested by which the secondary MGC can acquire the SIP states and continue with the call.

6.1 MGC Intensive:

In this scenario, MGCs can subscribe to each other through a SIP SUBSCRIBE request for a HandOff Event. As soon as any MGC knows that is shutting down, it retrieves states of those terminations that are in a SIP Call, packs them in XML, and sends them to the MGC that will take its place using a SIP NOTIFY request. The MGC talking over sends its Contact and SDP in a Re-INVITE request to the SIP User Agent.

The advantage of this approach is that call-state machine is transparently transferred transparently without the involvement of remote peer.

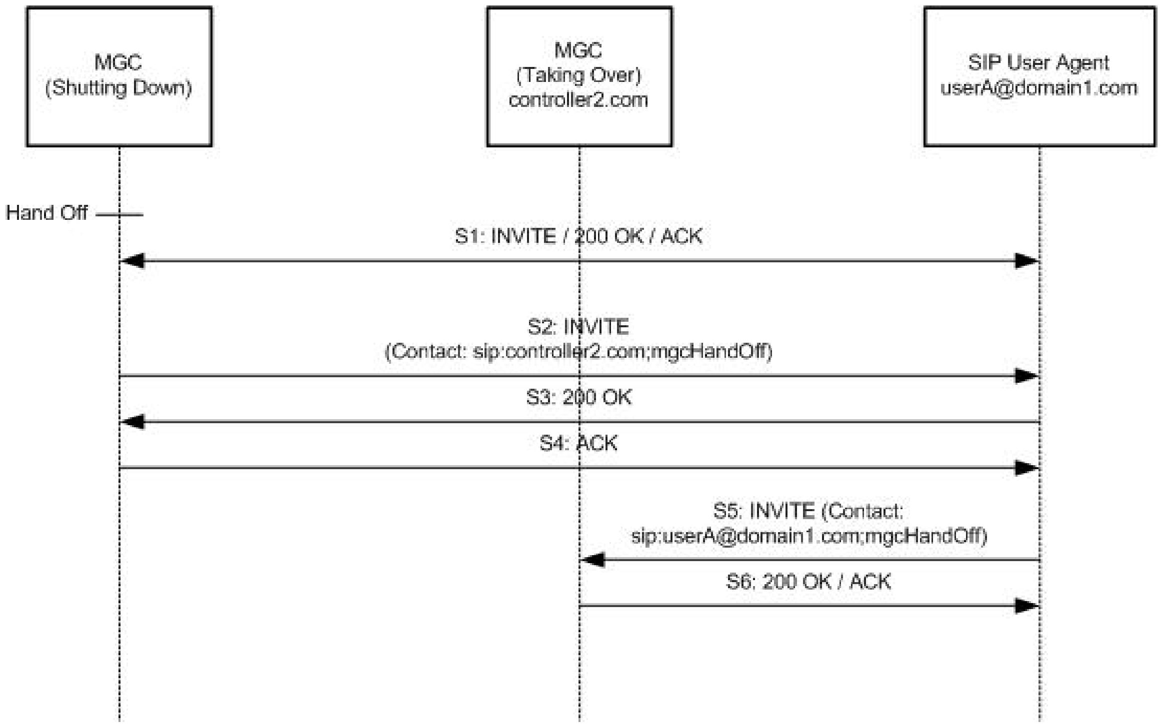


6.2 Client Intensive:

In this scenario the processing burden is shifted to the SIP User Agent. As soon as a MGC detects HandOff Event, it sends a re-INVITE to SIP User Agent. That re-INVITE contains the host name of new MGC in Contact field. It also contains a special parameter 'mgcHandOff' in the Contact Field that indicates the remote Peer to send a re-INVITE to the new MGC.

SIP User Agent then sends a re-INVITE to the MGC taking over that contains the 'mgcHandOff' parameter in the Contact Field. The MGC taking over sees this parameter and uses it to build the state of call.

This approach shifts the processing burden to clients. Of course User Agents must be able to understand the special parameter in the Contact Header which is also the biggest disadvantage of following this approach.



7. SIP for Megaco State Transfer

According to RFC 3015 Section 11.5 "No recommendation is made on how the MGCs involved in the Handoff maintain state information; this is considered to be out of scope of this recommendation". Consider the scenario that an MGC was controlling a large number of MGs and was being bombarded by UDP messages from hundreds of terminations. Suddenly the Administrator decides to shut down the MGC for maintenance purposes. The MGC stops reception of any further messaging from the MGs, sends messages already made for instruction of MGs in its transmission queues and sends a ServiceChange Handoff to all the MGs. Ideally speaking the MGs would then switch over to the new MGC, register themselves with it and carry on their media processing. However in the real world this would happen for a small percentage of the MGs only. Owing to the service delay of the UDP protocol and network losses many vital commands from the original MGC and possibly the ServiceChange Handoff command may be lost. Due to this packet loss many terminations are liable to go out of sync and some MGs might not even be aware that their controlling body has been changed.

To counter this problem we propose that for messages of vital importance and whose replies have not been successfully received by the failing MGC, SIP be used to transfer these messages to the secondary MGC by embedding the Megaco Messages as a SIP NOTIFY Message Body. The new MGC would then carry out the tasks that the failing MGC could not do successfully in its lifetime. The Content-Type of such a message will be application/megaco. Of course the MGCs will have subscribed other MGCs for receiving a HandOff Notification.

```
NOTIFY sip:controller2.com SIP/2.0
Via: SIP/2.0/UDP controller1.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:controller2.com>;tag=a6c85cf
From: <sip:controller1.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 2 NOTIFY
Content-Type: application/megaco
Content-Length: 380

MEGACO/1 [172.16.15.1]:1721 Transaction = 1007{Context = -
ServiceChange = ROOTMG1{Services{Method = Restart,Reason =
HandOff,ServiceChangeAddress = 1721,Profile = ResGw/0,Version = 1}}}}

MEGACO/1 [172.16.15.1]:1721 Transaction = 1007{Context = -
ServiceChange = ROOTMG2{Services{Method = Restart,Reason =
HandOff,ServiceChangeAddress = 1721,Profile = ResGw/0,Version = 1}}}}
```

8. Appendex A

M1: MG-1 registers with MGC1 using the ServiceChange command:

```
MEGACO/1 [172.16.15.1]:1721 Transaction = 1007{Context = -
{ServiceChange = ROOTMG1{Services{Method = Restart,Reason =
JustRestart,ServiceChangeAddress = 1721,Profile = ResGw/0,Version =
1}}}}
```

M2: MG1 registers a termination with MGC1:

MG1 to MGC1:
MEGACO/1 [172.16.15.1]:1721 Transaction = 1008{Context = -{ServiceChange =
0MG1{Services{Method = Restart,Reason = JustRestart,ServiceChangeAddress = 1721,Profile =
ResGw/0,Version = 1}}}}

M3: MGC1 sends a ServiceChange Reply

MGC1 to MG1:
MEGACO/1 [172.16.2.6]:2944 Reply = 1008{Context = -{ServiceChange =
0MG1{Services{ServiceChangeAddress = 2944,Profile = ResGw/0}}}}

M4: MGC1 Audits the Media Codecs on MG1

MGC1 to MG1
MEGACO/1 [172.16.2.6]:2944 Transaction = 10001{Context = -{AuditCapability =
0MG1{Audit{Media}}}}

M5: MG1 sends its SDPs in return

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Reply = 10001{Context = -{AuditCapability = 0MG1{Media{Stream =
0{Local{
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
}}}}}}

M6: The MG1 accepts the Modify with this reply:

MGC1 to MG1:
MEGACO/1 [172.16.2.6]:2944 Transaction = 10002{Context = -{Modify = 0MG1{Media{Stream =
1{LocalControl{Mode = SendReceive},Local{
}}},DigitMap = DialPlan0{9737xxxx},Events = 0{al/of{Embed{Signals{cg/dt},Events =
0{al/on,dd/ce}}}}}}}

M7: The MG1 replies to the Modify Command

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Reply = 10002{Context = -{Modify = 0MG1}}

M8: The MG2 registers it self with MGC2

MG2 to MGC2

MEGACO/1 [172.16.15.1]:1721 Transaction = 1003{Context = -{ServiceChange = ROOTMG2{Services{Method = Restart,Reason = JustRestart,ServiceChangeAddress = 1721,Profile = ResGw/0,Version = 1}}}}

M9: The MGC2 replies to MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Reply = 1003{Context = -{ServiceChange = ROOT{Services{ServiceChangeAddress = 2944,Profile = ResGw/0}}}}

M10: The MG2 registers its termination with MGC2

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Transaction = 1004{Context = -{ServiceChange = 0MG2{Services{Method = Restart,Reason = JustRestart,ServiceChangeAddress = 1721,Profile = ResGw/0,Version = 1}}}}

M11: The MGC2 replies to MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Reply = 1004{Context = -{ServiceChange = 0MG2{Services{ServiceChangeAddress = 2944,Profile = ResGw/0}}}}

M12: The MGC2 Audits the Capabilities of MG2

MEGACO/1 [172.16.2.6]:2944 Transaction = 10003{Context = -{AuditCapability = 0MG2{Audit{Media}}}}

M13: The MG2 returns its capabilities to MGC2

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Reply = 10003{Context = -{AuditCapability = 0MG2{Media{Stream = 0{Local{
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
}}}}}}

M14: The MGC2 sends a modify to MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Transaction = 10004{Context = -{Modify = 0MG2{Media{Stream = 1{LocalControl{Mode = SendReceive},Local{
}}},DigitMap = DialPlan0{9737xxxx},Events = 0{al/of{Embed{Signals{cg/dt},Events = 0{al/on,dd/ce}}}}}}}

M15: The MG2 sends a modify reply

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Reply = 10004{Context = -{Modify = 0MG2}}

M16: MG2 notifies MGC2 of an Off hook Event

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Transaction = 3001{Context = -{Notify = 0MG2{ObservedEvents = 0{20012702T12000000 : al/of}}}}

M17: MG2 notifies MGC2 of Dial Digits Event

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Transaction = 3002{Context = -{Notify = 0MG2{ObservedEvents = 0{20012702T12000000 : dd/ce{ds = "97378000",Meth = FM}}}}}

M18: MGC2 Replies to the Notify

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Reply = 3001{Context = -{Notify = 0MG2}}

M19: MGC2 Replies to the Notify

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Reply = 3002{Context = -{Notify = 0MG2}}

M20: MGC2 Sends an Add command to MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Transaction = 10006{Context = ${Add = 0MG2,Add = ${Media{Stream = 1{LocalControl{Mode = ReceiveOnly},Local{
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
}}}}}}

M21: MG2 Replies to the Add Command

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Reply = 10006{Context = 1{Add = 0MG2,Add = 1000MG2{Media{Stream = 1{Local{
v=0
c=IN IP4 172.16.6.6
m=audio 5010 RTP/AVP 0
}}}}}}

M22: MGC1 Sends an Add command to MG1

MGC1 to MG1
MEGACO/1 [172.16.2.6]:2944 Transaction = 10007{Context = ${Add = 0MG1{Signals{al/ri},Events = 1{al/of}},Add = ${Media{Stream = 1{LocalControl{Mode = SendReceive},Local{
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
},Remote{
v=0
c=IN IP4 172.16.6.6
m=audio 5010 RTP/AVP 0
}}}}}}

M23: MG1 Replies to MGC1

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Reply = 10007{Context = 1{Add = 0MG1,Add = 1000MG1{Media{Stream = 1{Local{
v=0

c=IN IP4 172.16.2.22
m=audio 5010 RTP/AVP 0
}}}}}}

M24: MGC2 sends a Modify Command to MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Transaction = 10008{Context = 1{Modify = 0MG2{Signals{cg/rt}},Modify = 1000MG2{Media{Stream = 1{Remote{
v=0
c=IN IP4 172.16.2.22
m=audio 5010 RTP/AVP 0
}}}}}}

M25: MG2 replies to MGC2

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Reply = 10008{Context = 1{Modify = 0MG2,Modify = 1000MG2}}

M26: MG1 Notifies MGC1 of Offhook Event

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Transaction = 3001{Context = 1{Notify = 0MG1{ObservedEvents = 0{20012702T12000000 : al/of}}}}

M27: MGC1 replies to MG1

MGC1 to MG1
MEGACO/1 [172.16.2.6]:2944 Reply = 3001{Context = 1{Notify = 0MG1}}

M28: MGC1 sends a Modify to MG1 stopping all Signals

MGC1 to MG1
MEGACO/1 [172.16.2.6]:2944 Transaction = 10009{Context = 1{Modify = 0MG1{Signals{},Events = 0{al/on,bcas/sz,bcas/cf,ctyp/dtone}}}}

M29: MG1 Replies to the Modify Command

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Reply = 10009{Context = 1{Modify = 0MG1}}

M30: MGC2 stops all signals on MG2

MGC2 to MG2
MEGACO/1 [172.16.2.6]:2944 Transaction = 10010{Context = 1{Modify = 0MG2{Signals{},Events = 0{bcas/sz,al/on,bcas/cf,ctyp/dtone}},Modify = 1000MG2{Media{Stream = 1{LocalControl{Mode = SendReceive}}}}}}

M31: MG2 replies to the MGC2

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Reply = 10010{Context = 1{Modify = 0MG2,Modify = 1000MG2}}

M32: MG1 Notifies MGC1 of OnHook Event

MG1 to MGC1

MEGACO/1 [172.16.15.1]:1721 Transaction = 3002{Context = 1{Notify = 0MG1{ObservedEvents = 0{20012702T12000000 : al/on}}}}

M33: MGC2 subtracts the Terminations in Context in MG2

MGC2 to MG2
MEGACO/1 [172.16.15.1]:1721 Reply = 10011{Context = 1{Subtract = 0MG2,Subtract = 1000MG2}}

M34: MG2 Notifies MGC2 of OnHook

MG2 to MGC2
MEGACO/1 [172.16.15.1]:1721 Transaction = 3003{Context = -{Notify = 0MG2{ObservedEvents = 0{20012702T12000000 : al/on}}}}

M35: MGC1 subtracts the Terminations in Context in MG1

MGC1 to MG1
MEGACO/1 [172.16.2.6]:2944 Transaction = 10012{Context = 1{Subtract = 0MG1,Subtract = 1000MG1}}

M36: MGC2 Replies to MG2

MG2 to MGC2
MEGACO/1 [172.16.2.6]:2944 Reply = 3003{Context = -{Notify = 0MG2}}

M37: MG1 Replies to Subtract Command

MG1 to MGC1
MEGACO/1 [172.16.15.1]:1721 Reply = 10012{Context = 1{Subtract = 0MG1,Subtract = 1000MG1}}

SIP Messages

S1: MGC2 to MGC1

```
INVITE sip:97378000@controller1.com SIP/2.0
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
Supported: 100rel
CSeq: 1 INVITE
Contact: <sip:controller2.com>
Content-Type: application/sdp
Content-Length: 38

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

S2: MGC1 to MGC2

```
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
Supported: 100rel
CSeq: 1 INVITE
RSeq: 1234
Contact: <sip:controller1.com>
Content-Type: application/sdp
Content-Length: 38

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

S3: MGC2 to MGC1

```
PRACK sip:97378000@controller1.com SIP/2.0
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 PRACK
RAck: 1234 1 INVITE
Content-Length: 0
```

S4: MGC1 to MGC2

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 PRACK
RAck: 1234 1 INVITE
Contact: <sip:controller1.com>
Content-Length: 0
```

S5: MGC1 Sends a 180 Ringing Response to MGC2

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
Supported: 100rel
CSeq: 1 INVITE
```

```
RSeq: 1235
Contact: <sip:controller1.com>
Content-Length: 0
```

S6: MGC2 to MGC1: Sends PRACK to acknowledge the receipt of 180 Ringing

```
PRACK sip:97378000@controller1.com SIP/2.0
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 PRACK
RAck: 1235 1 INVITE
Content-Length: 0
```

S7: MGC1 to MGC2: Sends a 200 OK to acknowledge PRACK

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 PRACK
RAck: 1235 1 INVITE
Contact: <sip:controller1.com>
Content-Length: 0
```

S8: MGC2 to MGC1: Sends UPDATE

```
UPDATE sip:97378000@controller1.com SIP/2.0
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bKasdfhjk
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 2 UPDATE
Contact: <sip:controller2.com>
Content-Type: application/sdp
Content-Length: 50

v=0
c=IN IP4 172.16.6.6
m=audio 5010 RTP/AVP 0
```

S9: MGC1 to MGC2: Sends 200 OK for UPDATE

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bKasdfhjk
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
```

```
Call-ID: a84b4c76e66710
CSeq: 2 UPDATE
Contact: <sip:controller1.com>
Content-Type: application/sdp
Content-Length: 51

v=0
c=IN IP4 172.16.2.22
m=audio 5010 RTP/AVP 0
```

S10: MGC1 to MGC2: Sends 200 OK for INVITE

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bKasdfhjk
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 INVITE
Contact: <sip:controller1.com>
Content-Length: 0
```

S11: MGC2 to MGC1: Sends ACK for INVITE

```
ACK sip:97378000@controller1.com SIP/2.0
Via: SIP/2.0/UDP controller2.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: <sip:97378000@controller1.com>;tag=a6c85cf
From: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 ACK
Content-Length: 0
```

S12: MGC1 to MGC2: Sends BYE on detecting OffHook

```
BYE sip:8687000@controller2.com SIP/2.0
Via: SIP/2.0/UDP controller1.com;branch= z9hG4bKqwertyu
Max-Forwards: 70
From: <sip:97378000@controller1.com>;tag=a6c85cf
To: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 BYE
Content-Length: 0
```

S13: MGC2 to MGC1: Sends 200 OK to BYE request

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP controller1.com;branch= z9hG4bKqwertyu
Max-Forwards: 70
From: <sip:97378000@controller1.com>;tag=a6c85cf
To: <sip:8687000@controller2.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 1 BYE
Content-Length: 0
```

9. Disclaimer

This document should be treated as a mean to illustrate the usage of SIP across MGCs, not as a guide for implementing Media Gateway, Media Gateway Controller, SIP User Agent or SIP Proxy. The calls flows mentioned in this document are only informative. All the messages are encoded in the ABNF syntax for simplicity. The same calls flows are valid for binary Megaco messages. 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 and between MGCs. 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.

10. References

[1] F. Cuervo, N. Greene, C. Huitema, A. Rayhan, B. Rosen, J. Segers, “Media Gateway Control Protocol Version 1.0”, RFC 3015, November 2000.

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[3] Vemuri, A., Peterson, J., “Session Initiation Protocol for Telephones (SIP-T): Context and Architectures”, RFC 3372, September 2002.

[4] Madhubabu Brahmanapally, Prerepa Viswanadham, krishna Gundamaraju, Kenetec Inc “Megaco/H.248 Call flow examples”, July 2001

[5] Rosenberg, J., and Schulzrinne, H., and E. Schooler, "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[6] Rosenberg, J, " The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, September 2002.