Session 1352
VoIP and Circuit-to-Packet

OPNET Implementation of the Megaco/H.248 Protocol: Multi-Call and Multi-Connection Scenarios

Edlic Yiu, Edwood Yiu, and Ljiljana Trajković
{enyiu, eyiu, ljilja}@cs.sfu.ca

Presenter: Kun (Karen) Wu

Communication Networks Laboratory
Simon Fraser University
Vancouver, British Columbia, Canada

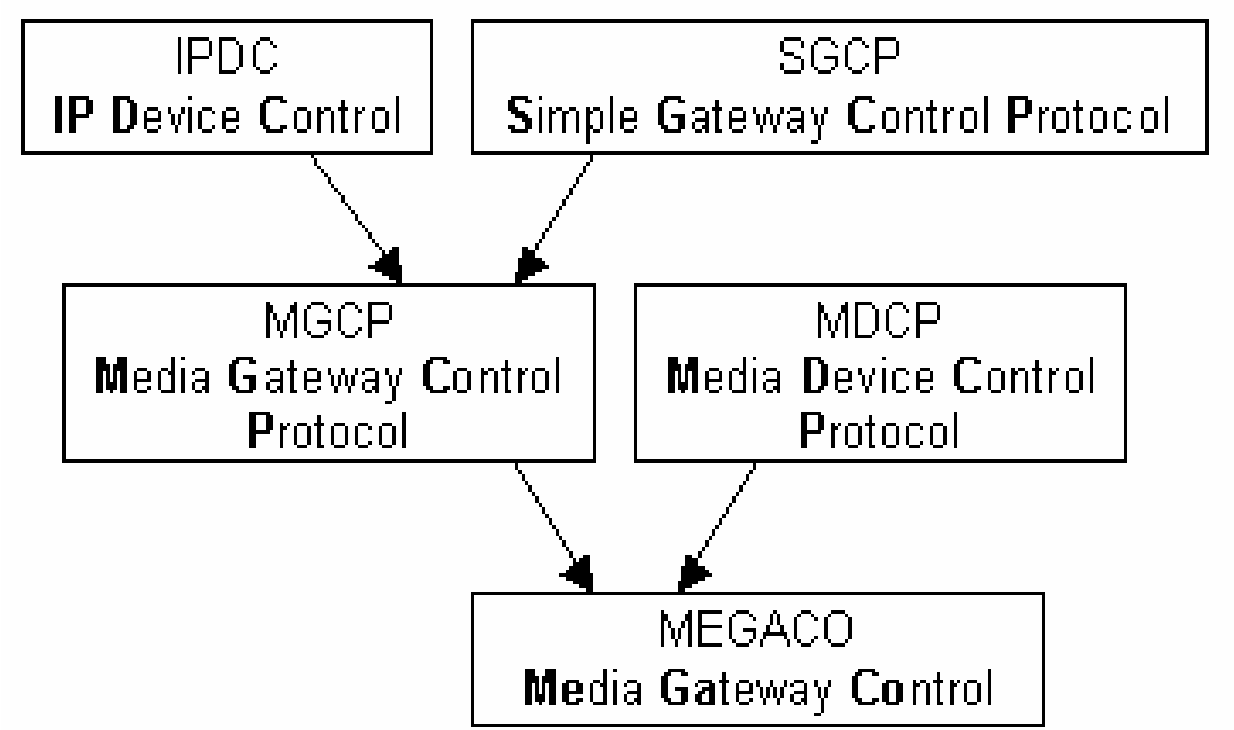
Roadmap

- Introduction
- Megaco/H.248 and VoIP
- Design architecture
- Design considerations
- OPNET implementation
- Call flow scenarios
- Simulation results
- Conclusion

Introduction

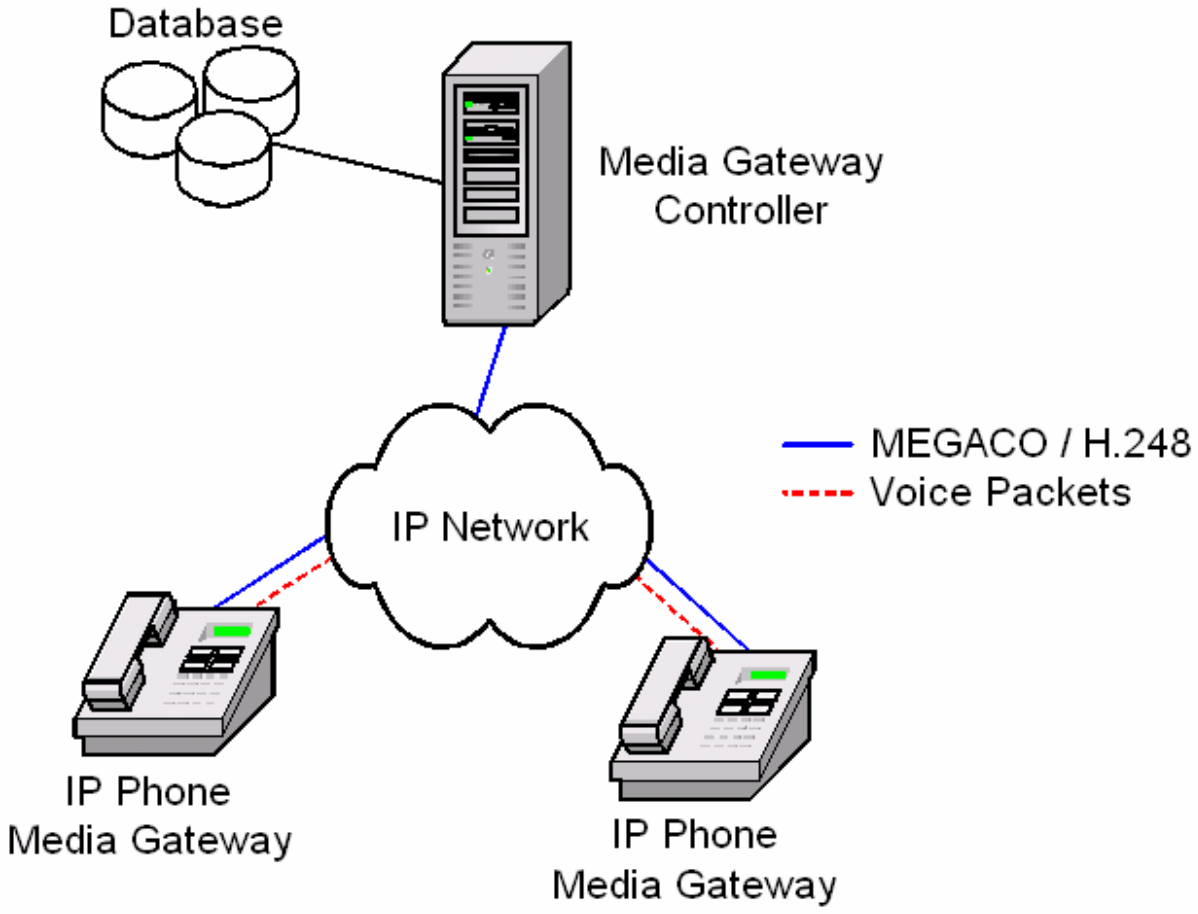
- Voice over IP (VoIP) is getting popular in both commercial and residential markets.
- It enables a telecommunication company to cut costs by allowing a single network to transmit both data and voice traffic.
- Offers inexpensive rate for long distance calls.
- Voice quality resulting from packets transmitted over the IP network is comparable to the voice quality in Public Switched Telephone Network (PSTN).
- To control and manage the voice traffic, Megaco/H.248 signaling protocol was introduced by Internet Engineering Task Force (IETF) and International Telecommunication Union (ITU).

Megaco/H.248 protocol: history



Gateway architecture

- Employs the master/slave architecture



Gateway architecture

- Media Gateway Controller (MGC):
  - central point of intelligence for call signaling
  - maintains the state of each MG and responds appropriately to any event notification
- Media Gateway (MG):
  - a dumb terminal
  - waits for the command from the MGC for its next action
  - streams voice packets over the IP network
  - de/compresses RTP packets

Megaco/H.248 command set

[MGC ↔ MG]

- ServiceChange — Notify the responder of the new service state

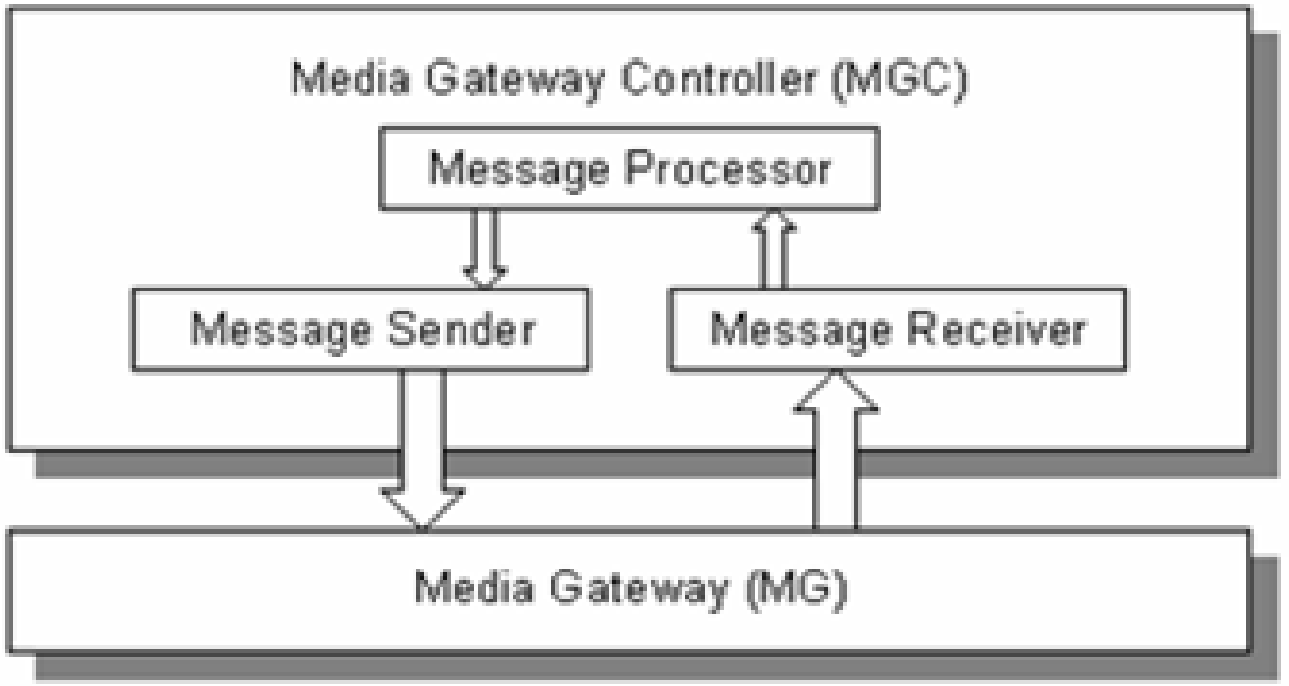
[MGC → MG]

- AuditValue — Determine the characteristics of an endpoint
- AuditCapabilities — Determine the capabilities of an endpoint
- Add — Add a connection
- Modify — Change a connection characteristic
- Subtract — Tear down a connection
- Move — Move an endpoint from one connection to another connection (call-waiting)

[MG → MGC]

- Notify — Notify the responder of an event (on-hook)

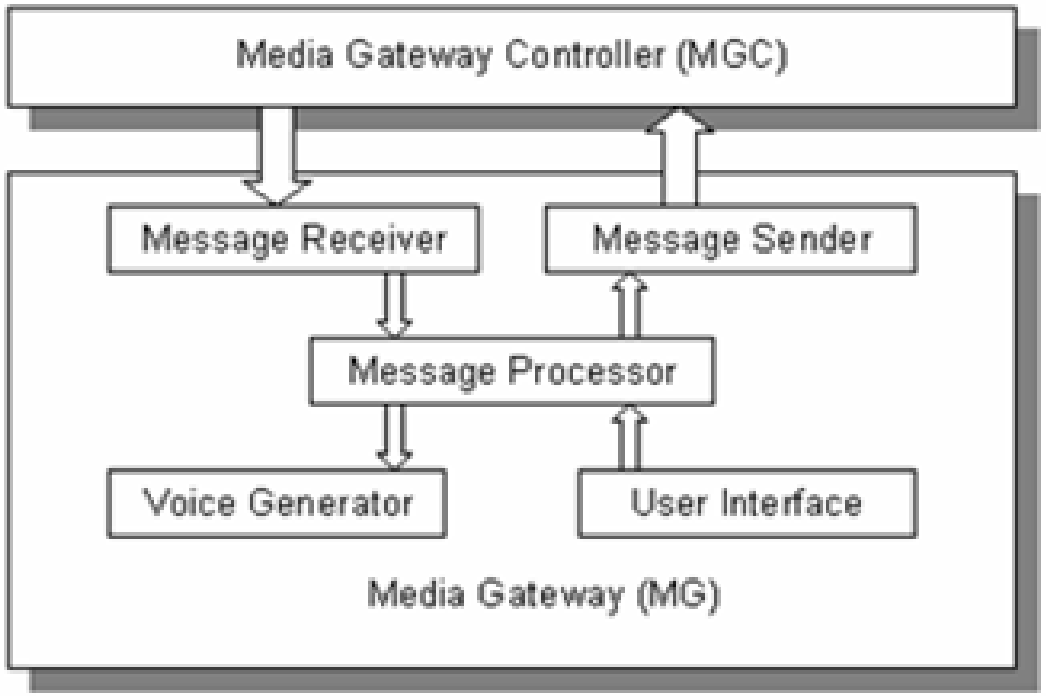
Design architecture: MGC



MGC component responsibilities

Component	Responsibility
Message Receiver	▪ Receive MEGACO messages from the MGs ▪ Extract parameters from MEGACO messages ▪ Redirect message parameters to MP
Message Processor	▪ Receive message parameters from MR ▪ Read statuses of the related MGs ▪ Determine actions for the related MGs ▪ Request MS to compose response messages if necessary
Message Sender	▪ Receive requests from MP ▪ Compose MEGACO messages ▪ Send MEGACO messages to the MGs

Design architecture: MG



MG component responsibilities

Component	Responsibility
Message Receiver	▪ Receive MEGACO messages from the MGC ▪ Extract parameters from MEGACO messages ▪ Redirect message parameters to the MP
Message Processor	▪ Receive message parameters from the MR ▪ Respond according to the command set ▪ Request the MS to compose response messages if necessary
Message Sender	▪ Receive requests from the MP ▪ Compose MEGACO message ▪ Send MEGACO messages to the MGC
User Interface	▪ Receive user request via the object attribute ▪ Request the MS to compose transaction message
Voice Generator	▪ Generate voice packets ▪ Encapsulate voice packet into RTP payloads ▪ Send RTP messages to other MGs

Design considerations

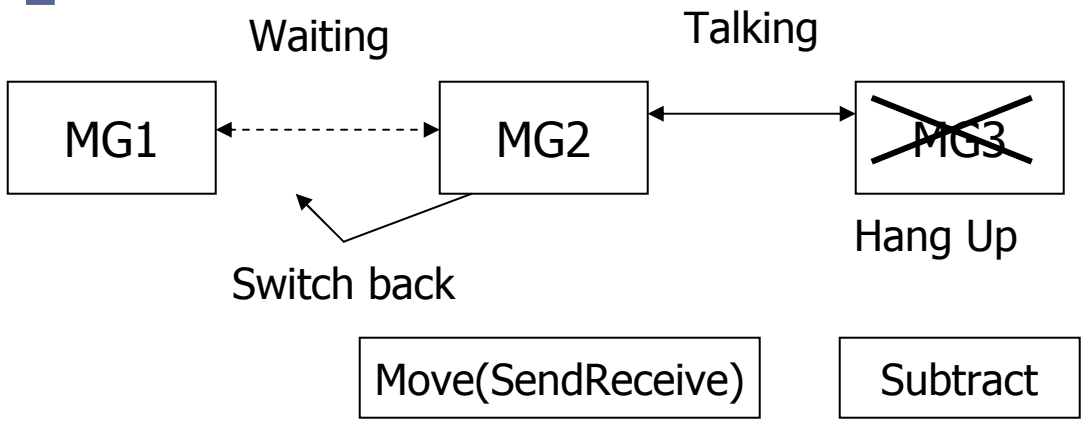
- Unlimited number of MGs
  - In order to support the multi-call and multi-connection scenario, MG architecture needs to support an unlimited number of MGs.
- Control intelligence in MG
  - We consider three cases for the Subtract command to illustrate the complexity in the multi-call and multi-connection scenarios.

Control intelligence in MGC

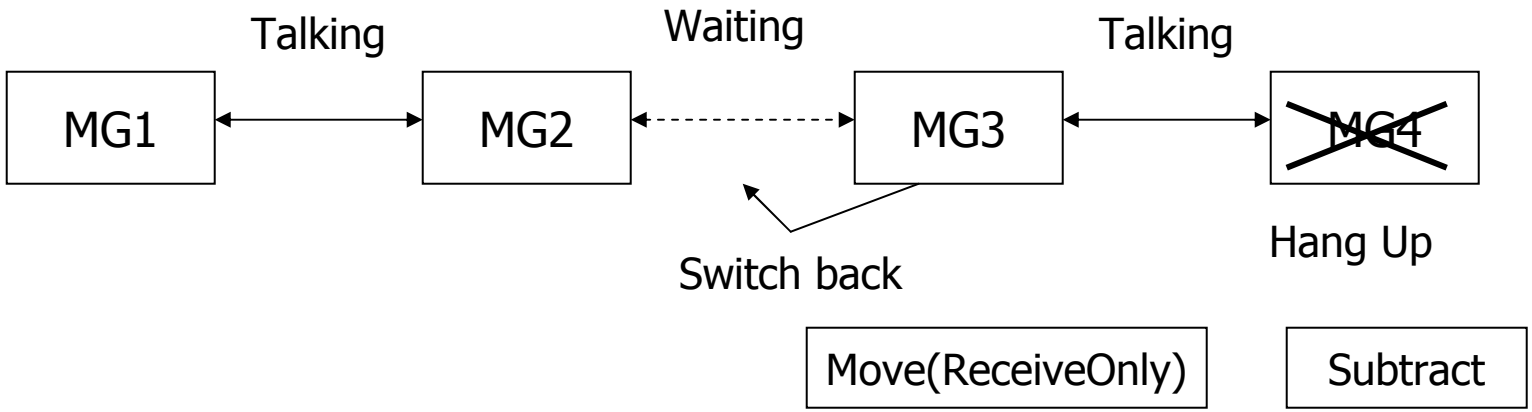
- Three scenarios are used to validate the control intelligence in MGC.
- Scenario 1

Control intelligence in MGC

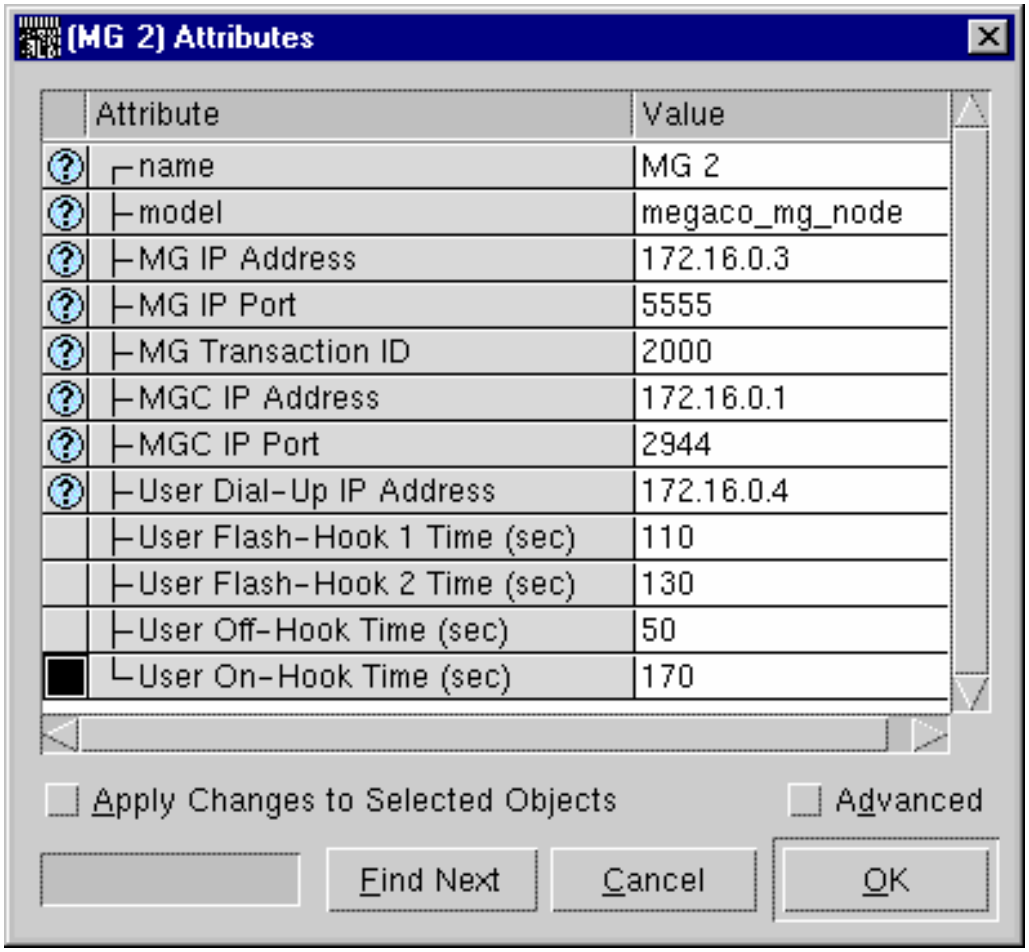
- Scenario 2



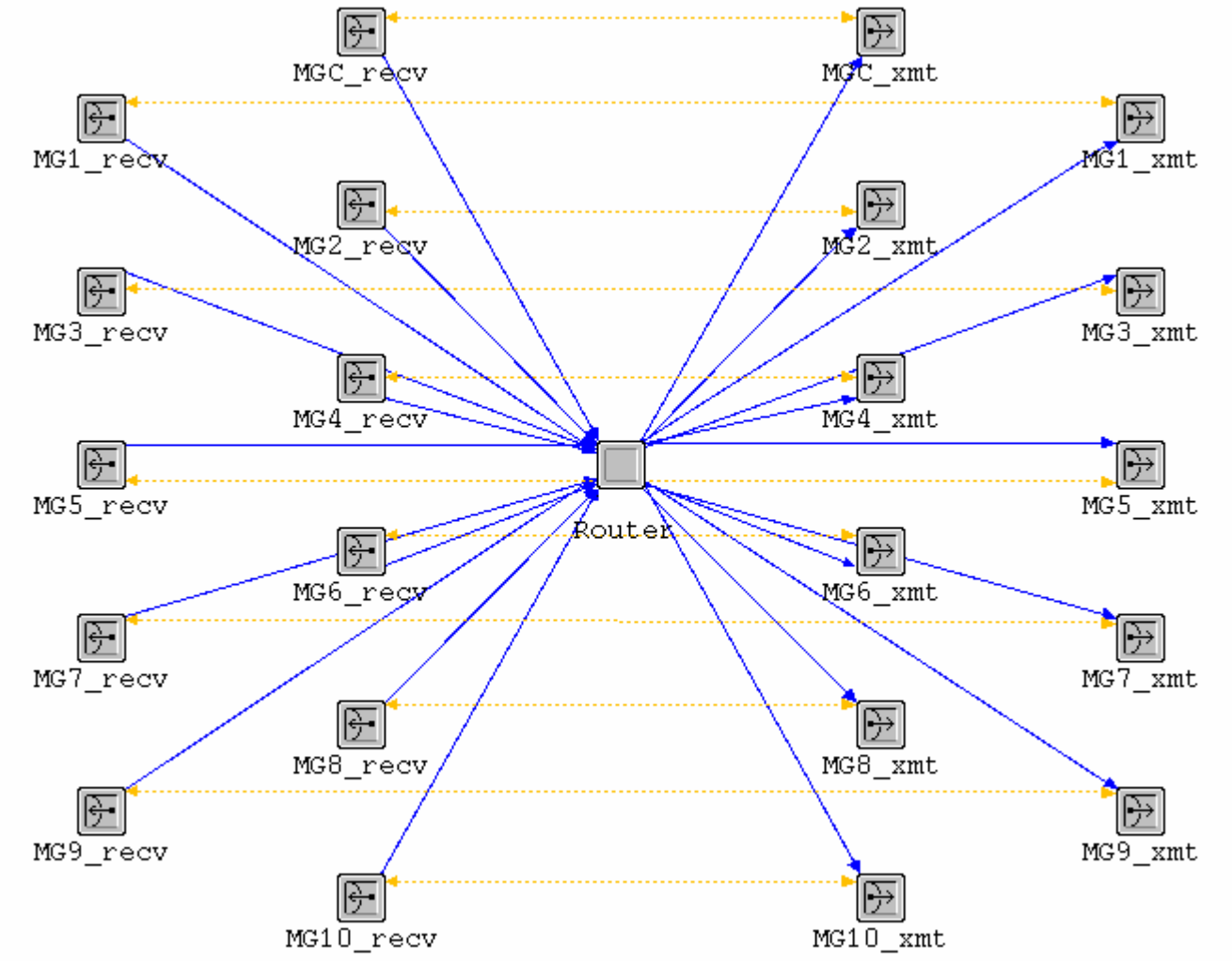
- Scenario 3



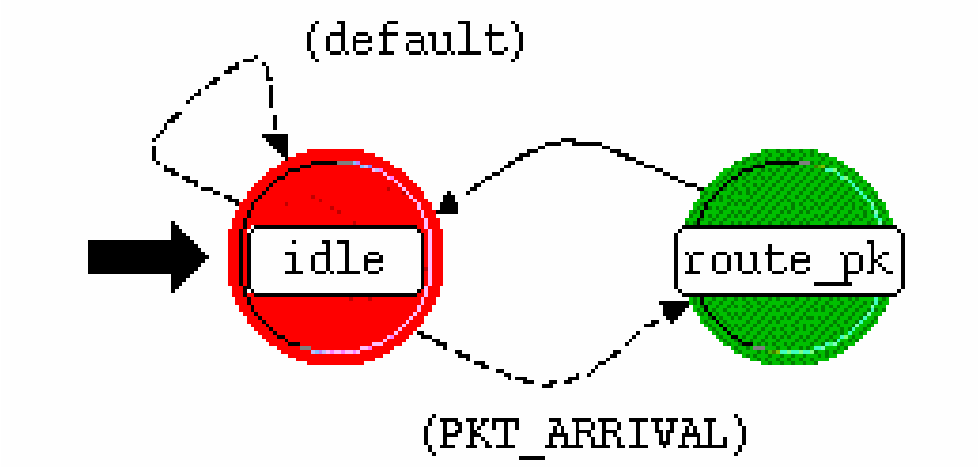
MG object attributes



OPNET implementation: router node model

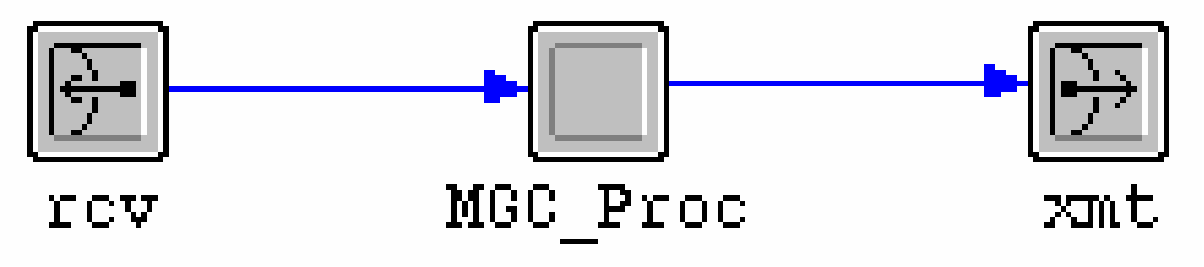


OPNET implementation: router process model



OPNET implementation: Media Gateway Controller

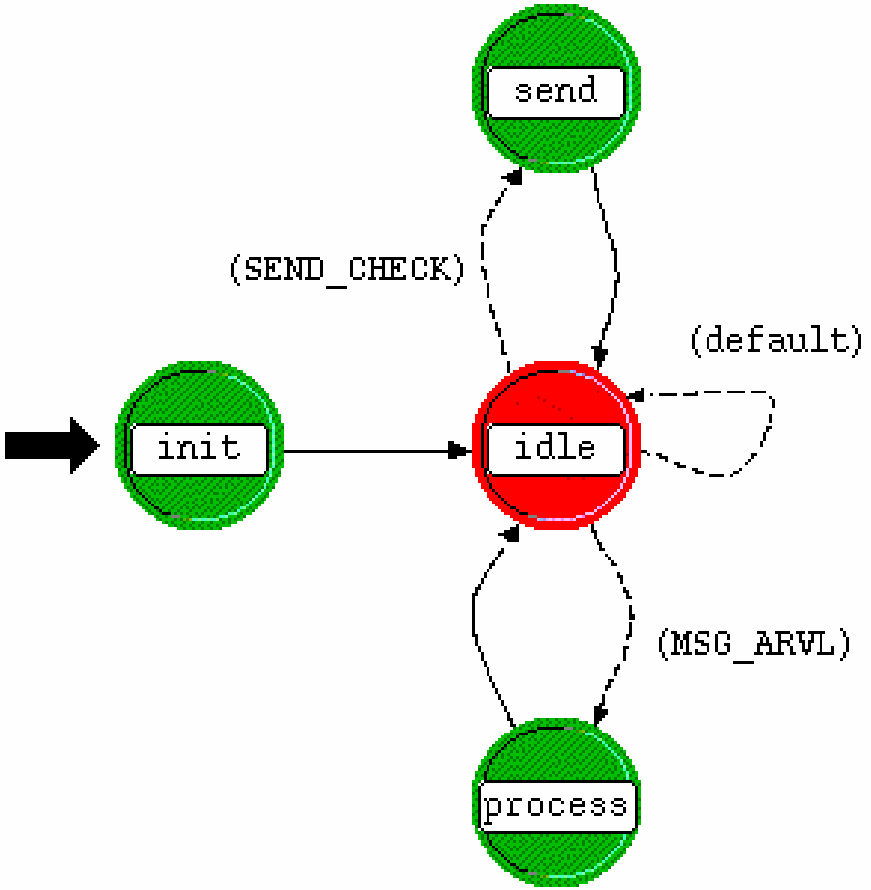
- MGC Node Model



- The MGC processor is responsible for:
  - parsing MEGACO/H.248 messages
  - determining necessary actions for the MGs
  - composing the MEGACO/H.248 messages.

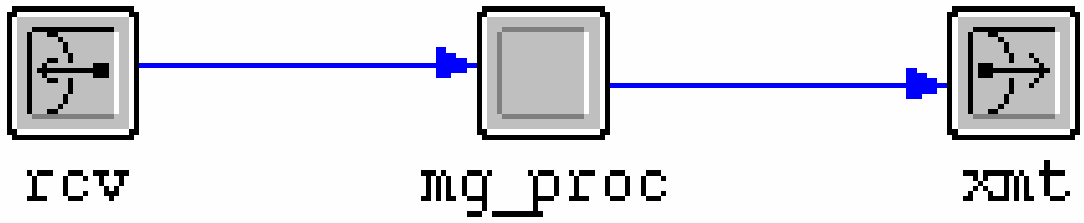
OPNET implementation: Media Gateway Controller

- MGC Process Model



OPNET implementation: Media Gateway

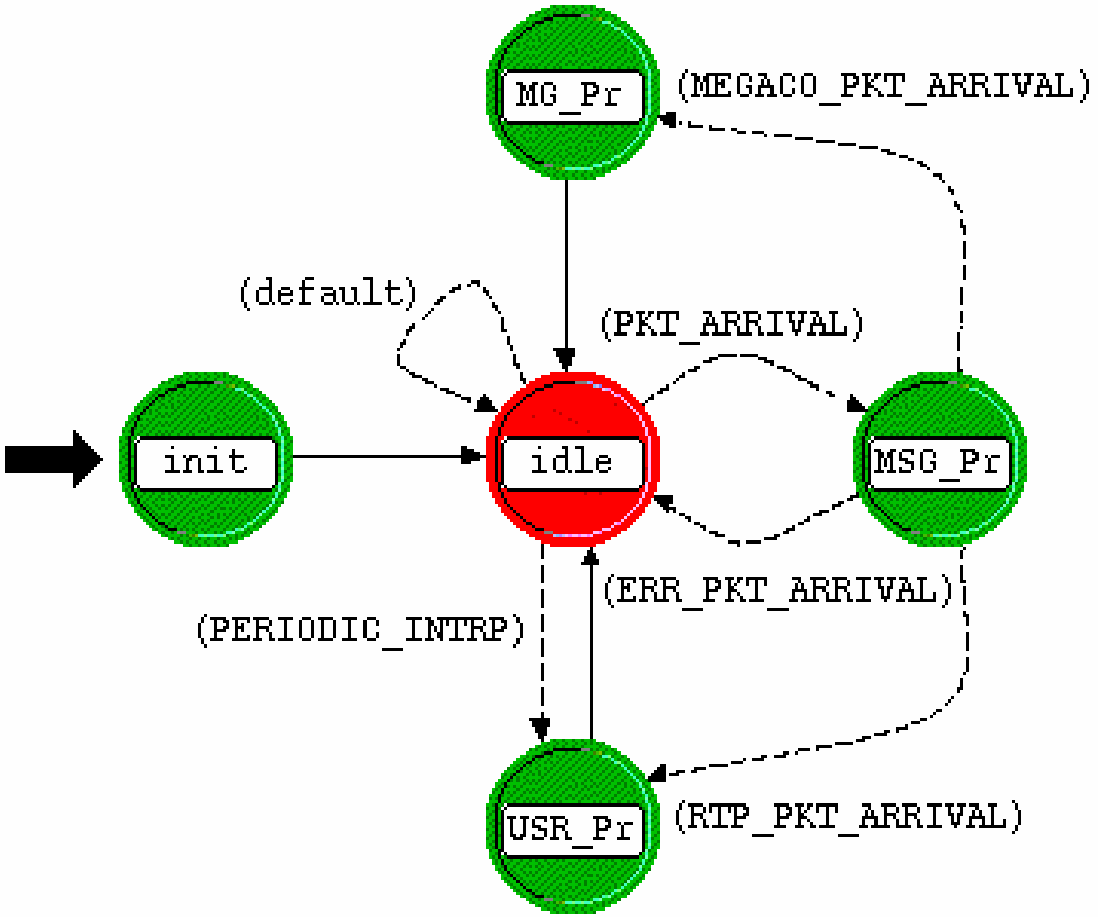
- MG Node Model



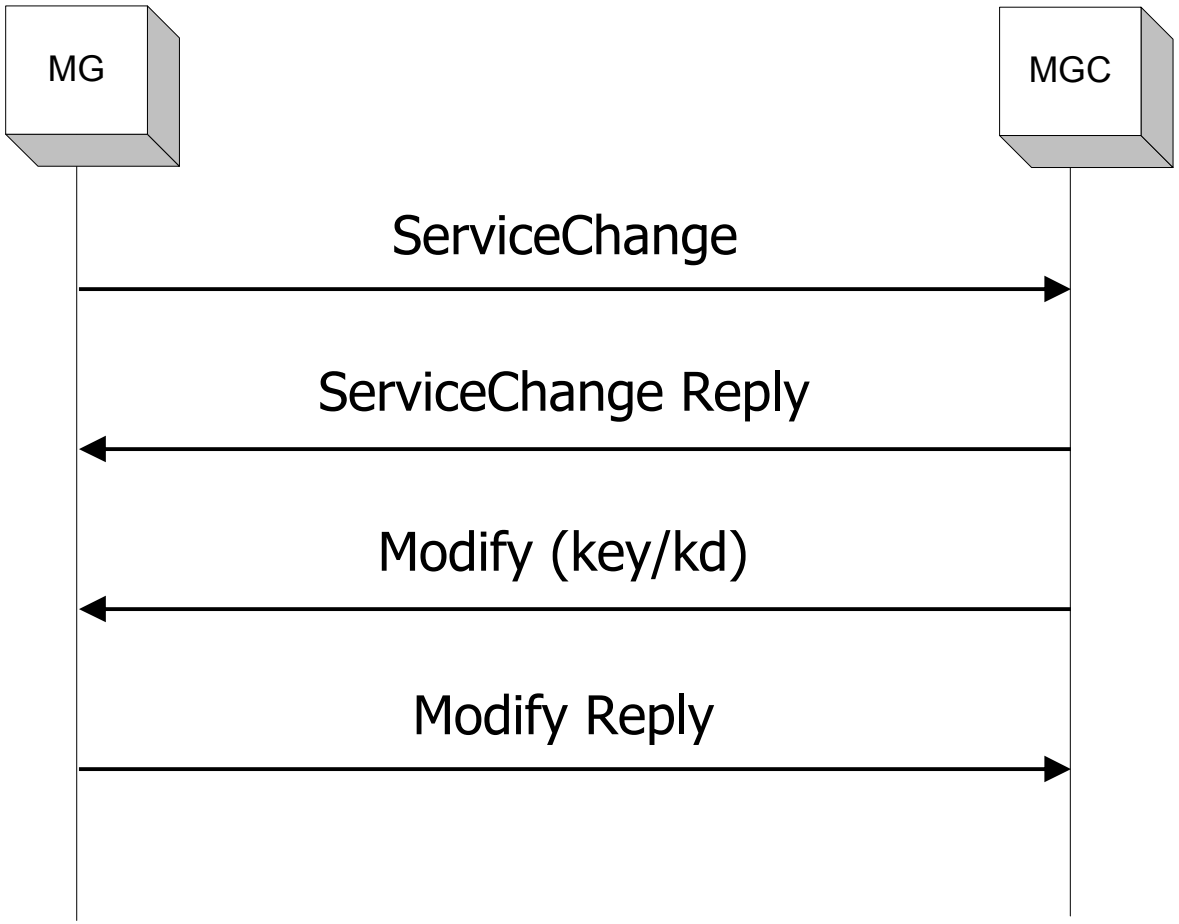
- The MG processor is responsible for:
  - handling MEGACO/H.248 commands sent from the MGC
  - detecting events initiated by the user
  - generating Real-Time Transport Protocol (RTP) packets for voice transmission.

OPNET implementation: Media Gateway

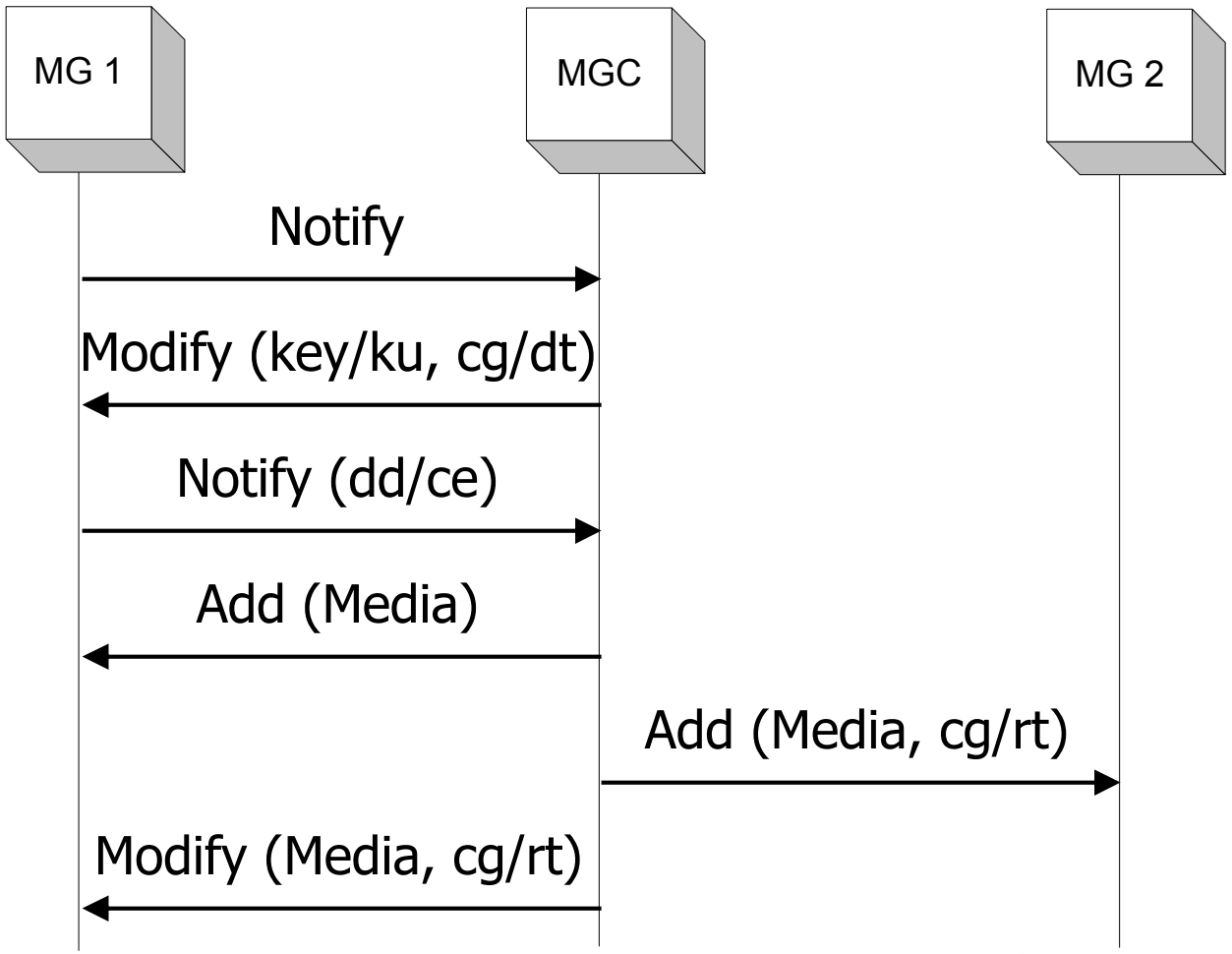
- MG Process Model



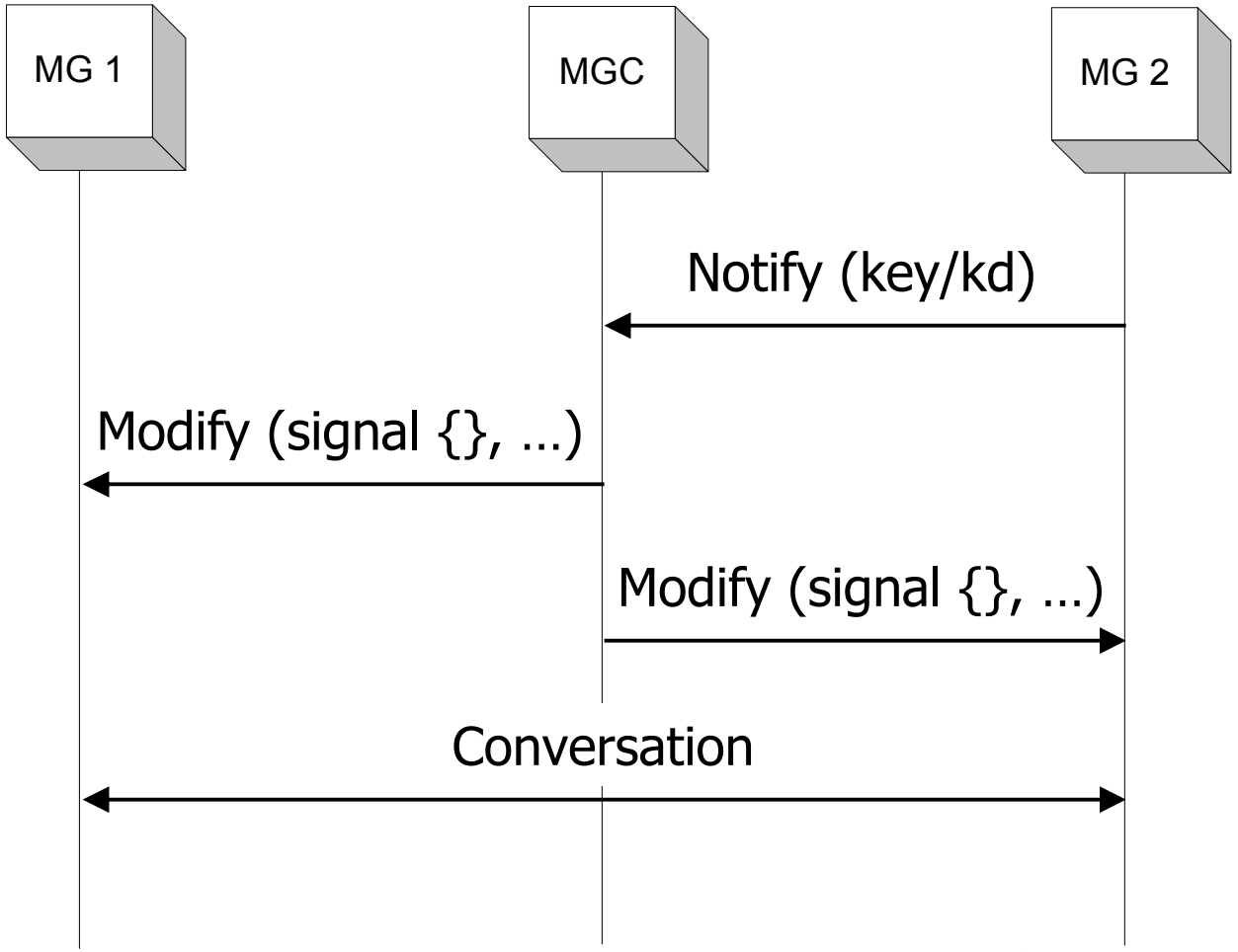
Call flow scenario: MG registration procedure



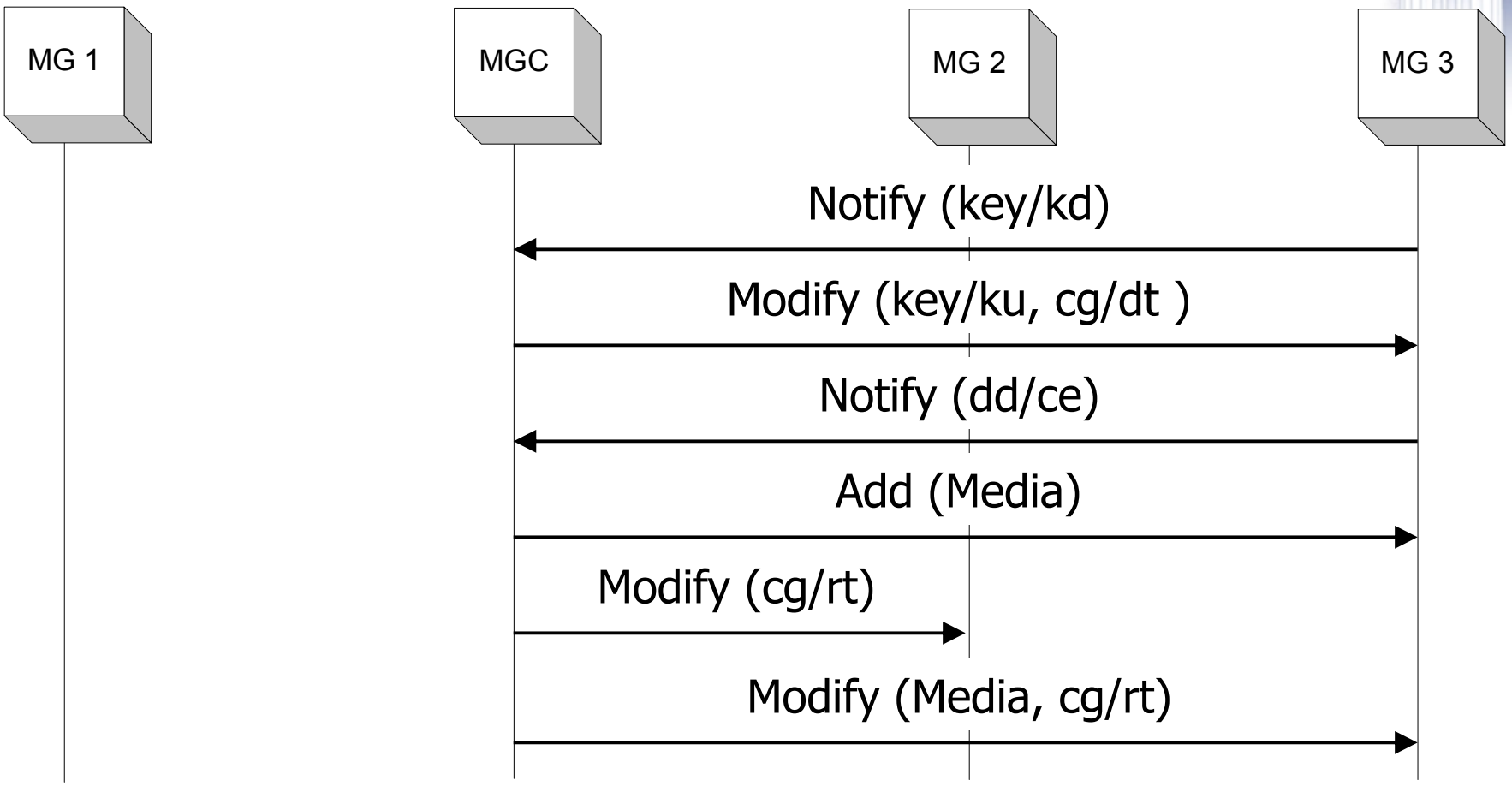
Call flow scenario: call setup procedure



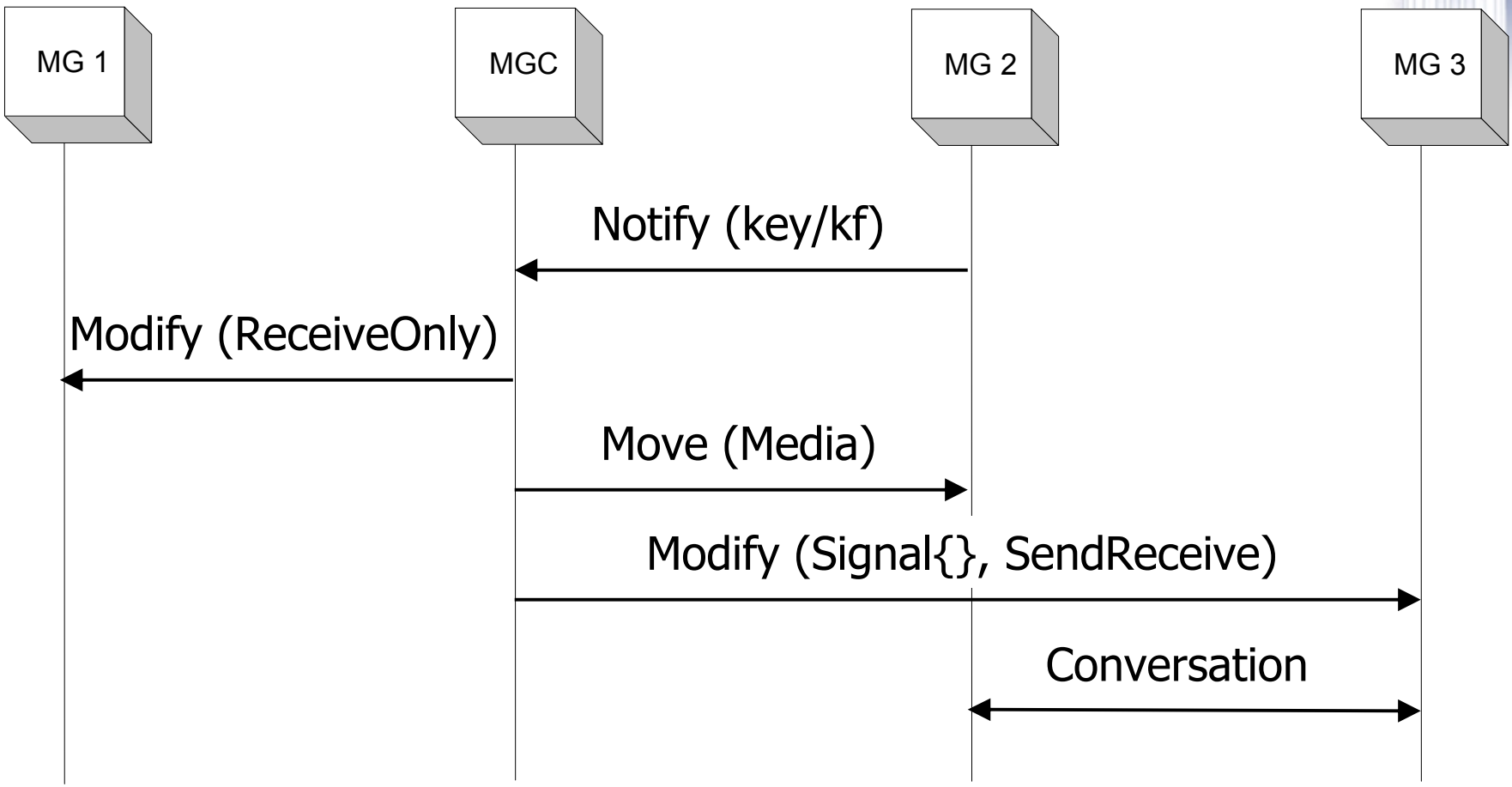
Call flow scenario: call setup procedure (cont.)



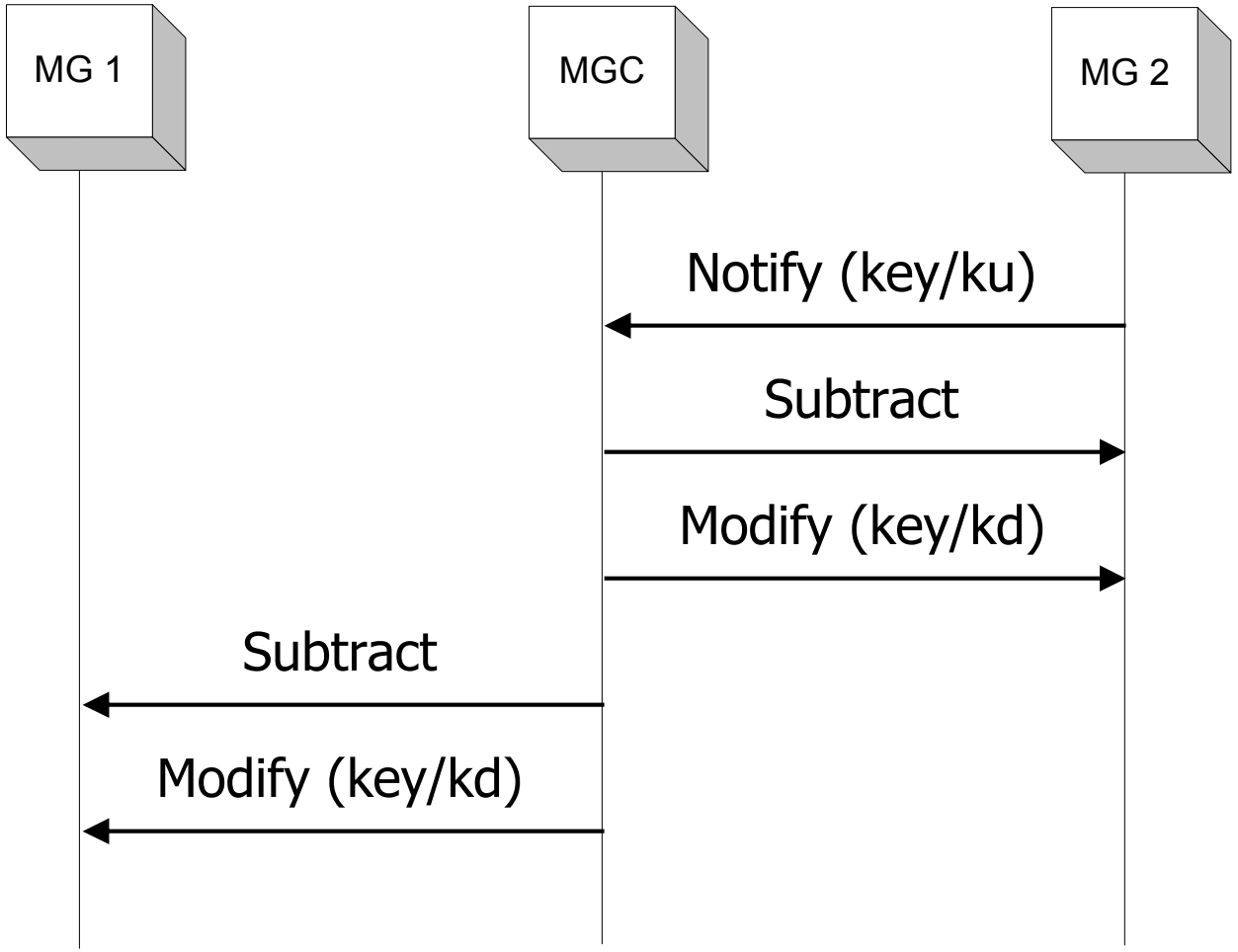
Call flow scenario: call waiting procedure



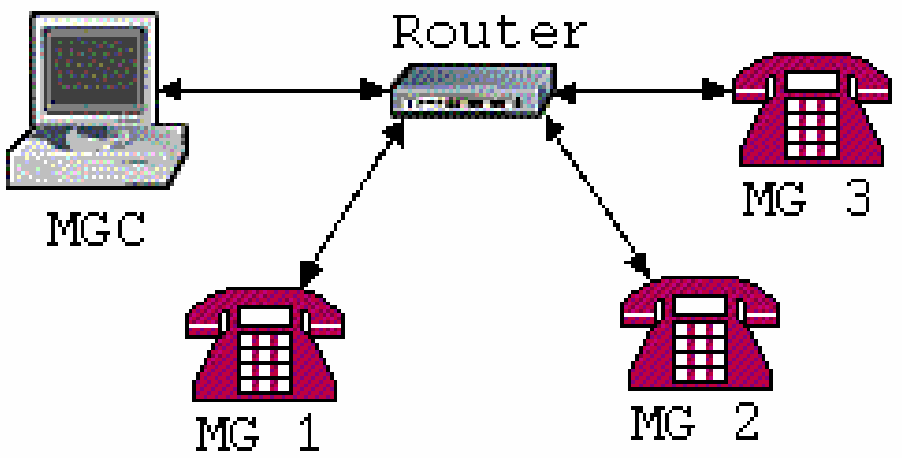
Call flow scenario: call waiting procedure (cont.)



Call flow scenario: call release procedure



Simulation scenario: simple call-waiting



1. MG1, MG2, and MG3 register with MGC
2. MG1 connects to MG2
3. MG3 calls MG2
4. MG2 switches to MG3, while MG1 is on hold
5. MG3 hangs up, and MG2 switches to MG1
6. MG2 hangs up

Subset of call-waiting scenario: call release

[1] MGC received the following message:

```
MEGACO/1 [172.16.0.3]:5555
Transaction = 2003 {
Context = - {
Notify = ui {
ObservedEvents = 5 {
20030414T145870:key/ku}}}}
```

[2] MGC just sent message to MG:

```
MEGACO/1 [172.16.0.1]:2944
Reply = 2003 {
Context = - {
Notify = ui}}
```

[3] MGC just sent message to MG:

```
MEGACO/1 [172.16.0.1]:2944
Transaction = 25 {
Context = 1 {
Subtract = at/hf,
Subtract = tr}}
```

[4] MGC received the following message:

```
MEGACO/1 [172.16.0.2]:5555
Reply = 25 {
Context = 1 {
Subtract = at/hf,
Subtract = tr {
Statistics {
rtp/ps=38,
rtp/pr=34,
rtp/pl=3,
rtp/jit=0}}}}
```

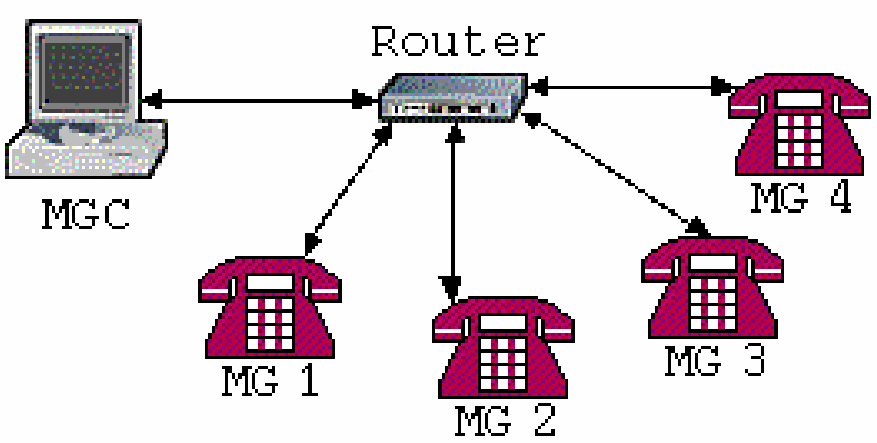
[5] MGC just sent message to MG:

```
MEGACO/1 [172.16.0.1]:2944
Transaction = 26 {
Context = - {
Modify = ui {
Events = 11 {key/kd}
},
Modify = at/hf {
Signal = {}}}}
```

[6] MGC received the following message:

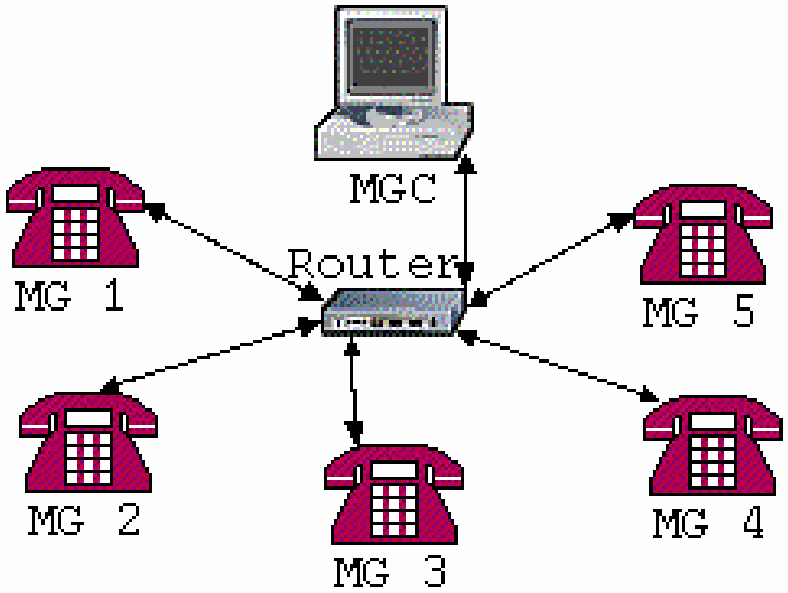
```
MEGACO/1 [172.16.0.2]:5555
Reply = 26 {
Context = - {
Modify = ui,
Modify = at/hf}}
```

Simulation scenario: complex call-waiting



1. All MGs register with MGC
2. MG2 connects to MG3
3. MG1 calls MG2, and MG4 calls MG3
4. MG2 switches to MG1, while MG3 switches to MG4
5. MG2 and MG3 switches back, while MG1 and MG4 are put on hold
6. MG3 hangs up
7. MG4 gets removed from the inactive connection
8. MG2 switches back to MG1
9. MG2 hangs up

Simulation scenario: multi-call and multi-connection



1. All MGs register with MGC
2. MG1 connects to MG2, while MG4 connects to MG5
3. MG3 calls MG2
4. MG2 switches to MG3, while MG1 is put on hold
5. MG3 hangs up, and MG2 switches back to MG1
6. MG2 and MG4 hang up.

Conclusion and future work

- We described the OPNET implementation of Megaco/H.248 signaling protocol.
- The OPNET implementation of Megaco/H.248 protocol supports an unlimited number of MG interconnections.
- Several call flow scenarios between the MGC and MGs were simulated to verify the implementation.
- Future work:
  - implementation of the Megaco/H.248 protocol over the IP network.

References

[1] S. Wu, M. Riyadh, and R. Mannan, and Lj. Trajković, "OPNET implementation of the Megaco/H.248 protocol," OPNETWORK 2002, Washington, DC, Aug. 2002.

[2] T. Taylor, "Megaco/H.248: a new standard for media gateway control," *IEEE Communications Magazine*, pp. 124-132, October 2000.

[3] "Media Gateway Control (Megaco)," Alcatel Executive Briefing, Alcatel Internetworking, December 2001.

[4] N. Greene, M. Ramalho, and B. Rosen, "Media Gateway Control Protocol architecture and requirements," RFC 2805, April 1999: http://www.ietf.org/rfc/rfc2805.txt (accessed in February 2003).

[5] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers, "Megaco protocol version 1.0," RFC 3015, November 2000: http://www.ietf.org/rfc/rfc3015.txt (accessed in February 2003).

[6] P. Blatherwick, R. Bell, and P. Holland, "Megaco IP phone media gateway application profile," RFC 3054, January 2001: http://www.ietf.org/rfc/rfc3054.txt (accessed in February 2003).

[7] M. Brahmanapally, P. Viswanadham, and K. Gundamaraj, "Megaco/H.248 call flow examples," October 2002: http://www.ietf.org/internet-drafts/draft-ietf-megaco-callflows-01.txt (accessed in February 2003).

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," RFC 1889, January 1996: http://www.ietf.org/rfc/rfc1889.txt (accessed in August 2002).