Abstract
This article explains how the MGCP and MEGACO protocols work. A brief introduction about how MGCP was born is given and then the various messages in the MGCP protocols are explained. Couple of scenarios are then presented where we see how the protocol actually works. This is followed by brief look at the other variant of this Master Slave protocol called Megaco. Conclusion of the paper is then presented. Appendix A contains the glossary of terms used in this article, while Appendix B contains the notations used to explain MGCP Messages. Appendix C contains some interesting comments made at the VON conference.

1 Introduction
In 1998 some R&D departments started to realize that H.323v1 was not satisfying some very important requirements from the carriers. Lack of mature products, lack of some features in H.323v1, lack of marketing efforts in favor of H.323v2 and time to market issues pushed the incumbent vendors to react against H.323 and propose alternative protocols to address the needs of large-scale phone-to-phone deployments. In mid 1998, the important RFI (Request for Information) and RFP (Request for Proposal) for building large VoIP networks were sent to vendors. The first proposal came from Bellcore (now Telcordia) and Cisco by the name of SGCP (Simple Gateway control protocol). The second proposal came from ITU-T SG16, ETSI TIPHON and IETF by the name of IPDC (Internet Protocol Device Control).

It was not long before the forces behind these two protocols realized that by unifying their efforts they could get bigger consensus and foster the adoption of their position. Bellcore and Level3 played a key role in merging these protocols into one, the MGCP (Media Gateway control protocol).

Some time later another protocol by the name of MEGACO was introduced. Megaco is now a coordinated standard between IETF (MEGACO) and the ITU (H.248).
In both MGCP and MEGACO/H.248 the main two components are Media Gateway and Media Gateway Controller.

Media Gateways are low intelligence distributed devices, which terminates lines/trunks and provide translation of POTS voice/fax signals for IP transport. Media Gateway Controller provides centralized intelligence for

a) total control over Media Gateways
b) Call admission and billing
c) Signaling interface to PSTN
d) Translation for other protocols. E.g. SIP and H.323.

2 MGCP
MGCP is designed to interface a media gateway controller and media gateway. MGCP is a text-based protocol and supports centralized call model. MGCP is a master slave protocol. MGCP assumes a call control architecture where the call control "intelligence" is outside the gateways and handled by external call control elements.

In its principle MGCP is very close to the proprietary protocols of the switch manufacturers that convey information back and forth between call control points and service switching points. This principle in the context of MGCP clearly places the intelligence on the physically separate entity, the media gateway controller and not on the hardware endpoint, the media converter. But unlike the switch architecture as specified in IN documents where the call control remains close to the actual hardware endpoints, in the MGCP architecture the call control functionality is no longer attached to the media part.

The MGCP assumes that these call control elements, or Call Agents, will synchronize with each other to send coherent commands to the gateways under their control. MGCP does not define a mechanism for synchronizing Call Agents. MGCP is, in essence, a master/slave protocol, where the gateways are expected to execute commands sent by the Call Agents.

MGCP allows combination of commands to be sent in one PDU, this combination reduces the number of messages necessary to establish a call. However, MGCP still requires at least 11 round trips to establish a phone to phone call.

MGCP has seamless PSTN Integration. Many existing Internet Telephony solutions require two stage dialing where a gateway number must be dialed prior to dialing the actual destination number. This is cumbersome for the end-user. However, if gateways are made dumb then they will be inexpensive enough for the end-users to buy
and place in their home. This avoids the need for two-stage dialing since the users telephone will already be connected to the gateway!

MGCP assumes a connection model where the basic constructs are endpoints and connections. Endpoints are sources or sinks of data and could be physical or virtual. Example of physical endpoints is an interface on a gateway that terminates a trunk connected to a PSTN switch. Example of a virtual endpoint is an audio source in an audio-content server.

Connections may be either point to point or multipoint. A point to point connection is an association between two endpoints with the purpose of transmitting data between these endpoints. Once this association is established for both endpoints, data transfer between these endpoints can take place. A multipoint connection is established by connecting the endpoint to a multipoint session.

Connections can be established over several types of bearer networks:
- Transmission of audio packets using RTP and UDP over a TCP/IP network.
- Transmission of audio packets using AAL2, or another adaptation layer, over an ATM networks.
- Transmission of packets over an internal connection, for example the TDM backplane or the interconnection bus of a gateway.

2.1 Telephony Gateway

A telephony gateway is a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Examples of gateways are:
- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits
- Voice over ATM gateways, which operate much the same way as voice over IP trunking gateways, except that they interface to an ATM network.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broadband wireless devices
- Access gateways, that provide a traditional analog (RJ11) or digital PBX interface to a Voice over IP network. Examples of access gateways include small-scale voice over IP gateways.
- Business gateways, that provide a traditional digital PBX interface or an integrated "soft PBX" interface to a Voice over IP network.
- Network Access Servers that can attach a "modem" to a telephone circuit and provide data access to the Internet. It is expected, in the future, the same gateways will combine Voice over IP services and Network Access services.
- Circuit switches, or packet switches, which can offer a control interface to an external call control element.

Note: The examples of gateways give above are just functional classification of gateway. It is possible that two or more gateways explained above are present in the same physical gateway.

2.2 Calls and Connections

Connections are created on the call agent on each endpoint that will be involved in the "call." Each connection will be designated locally by a connection identifier, and will be characterized by connection attributes.

When the two endpoints are located on gateways that are managed by the same call agent, the creation is done via the three following steps:

1. The call agent asks the first gateway to "create a connection" on the first endpoint. Denoted by Step 1 in Figure 1. The gateway allocates resources to that connection, and respond to the command by providing a "session description." (step 2) The session description contains the information necessary for a third party to send packets towards the newly created connection, such as for example IP address, UDP port, and packetization parameters.

2. The call agent then asks the second gateway to "create a connection" on the second endpoint. (Step 3) The command carries the "session description" provided by the first gateway. The gateway allocates resources to that connection, and respond to the command by providing its own "session description." (Step 4).

3. The call agent uses a "modify connection" command to provide this second "session description" to the first endpoint.(Step 5) Once this is done, communication can proceed in both directions.
When the two endpoints are located on gateways that are managed by the different call agents, these two call agents shall exchange information through a call-agent to call-agent signaling protocol, in order to synchronize the creation of the connection on the two endpoints. Once established, the connection parameters can be modified at any time by a "modify connection" Command. The call agent may for example instruct the gateway to change the compression algorithm used on a connection, or to modify the IP address and UDP port to which data should be sent, if a connection is "redirected."

The call agent removes a connection by sending to the gateway a "delete connection" command. The gateway may also, under some circumstances, inform a gateway that a connection could not be sustained

### 2.3 Usage of SDP

The Call Agent uses the MGCP to provision the gateways with the description of connection parameters such as IP addresses, UDP port and RTP profiles. These descriptions will follow the conventions delineated in the Session Description Protocol which is now an IETF proposed standard, documented in RFC 2327. SDP allows for description of multimedia conferences. This version limits SDP usage to the setting of audio circuits and data access circuits. The initial session descriptions contain the description of exactly one media, of type "audio" for audio connections, "nas" for data access.

### 2.4 High Availability and Load Balancing in MGCP

Call Agents are identified by their domain name, not their network addresses, and several addresses can be associated with a domain name. In a typical configuration, the MG sends Notifications to the CA. After trying to contact the CA for some configurable number of times and not getting any response back, it starts contacting the other (back-up) MGC within the same domain name.

If a CA is overloaded, it can inform the MG about the same, by changing the Notified Entity with the MG to a new CA. Therefore, when the MG has to deliver the next Notification, it does so to the new CA.

### 2.5 MGCP Commands

The table below lists the various MGCP Commands. CA denotes the Call Agent and GW denotes the Gateway. CA --> GW would mean that the command is send from CA to GW.

<table>
<thead>
<tr>
<th>Sr no.</th>
<th>Commands</th>
<th>Command flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateConnection</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>2</td>
<td>ModifyConnection</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>3</td>
<td>DeleteConnection</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>4</td>
<td>NotificationRequest</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>5</td>
<td>Notify</td>
<td>CA &lt;-- GW</td>
</tr>
<tr>
<td>6</td>
<td>AuditEndpoint</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>7</td>
<td>AuditConnection</td>
<td>CA --&gt; GW</td>
</tr>
<tr>
<td>8</td>
<td>RestartInProgress</td>
<td>CA &lt;-- GW</td>
</tr>
<tr>
<td>9</td>
<td>EndpointConfiguration</td>
<td>CA --&gt; GW</td>
</tr>
</tbody>
</table>

We shall now look into the individual MGCP Commands. Every command is represented by a few parameters, details on what those parameters can be found in Appendix B. For more information on how command is represented, check the RFC 2705.

#### Endpoint Configuration

The EndpointConfiguration commands are used to specify the encoding of the signals that will be received by the endpoint. For example, in certain international telephony configurations, some calls will carry mu-law encoded audio signals, while other will use A-law. The Call Agent will use the EndpointConfiguration command to pass this information to the gateway.

Command is represented by:

```
ReturnCode
EndpointConfiguration( EndpointId, BearerInformation)
```

#### Notification Request

The Notification Request commands are used to request the gateway to send notifications upon the occurrence of specified events in an endpoint. For example, a notification may be requested for when a gateway detects that an endpoint is receiving tones associated with fax communication. One of the nice features of this command is the association of actions with each of the events. Using this facility, the communication and processing of information between the two entities can be optimized. To each event is associated an action, which can be:

- Notify the event immediately, together with the accumulated list of observed events,
- Accumulate the event in an event buffer, but don't notify yet.
- Accumulate according to Digit Map.

Command is represented by:

```
ReturnCode
NotificationRequest( EndpointId, [NotifiedEntity,] [RequestedEvents,] RequestIdentifier)
```
Create Connection
This command is used to create a connection between two endpoints. In addition to the necessary parameters that enable a media gateway to create a connection, the localConnectionOptions parameter provides features for quality of service, security, and network related QOS.

Command is represented by:
ReturnCode,
ConnectionId,
[SpecificEndPointId],
[LocalConnectionDescriptor],
[SecondEndPointId],
[SecondConnectionId]

CreateConnection(CallId,
EndpointId,
[NotifiedEntity],
LocalConnectionOptions,
Mode,
[RemoteConnectionDescriptor | SecondEndPointId],
[Encapsulated NotificationRequest],
[Encapsulated EndpointConfiguration])

Modify Connection
This command is used to modify the characteristics of a gateway's "view" of a connection. This "view" of the call includes both the local connection descriptors as well as the remote connection descriptor.

Command is represented by:
ReturnCode,
[LocalConnectionDescriptor]
ModifyConnection(CallId,
EndpointId,
ConnectionId,
[NotifiedEntity],
LocalConnectionOptions,
Mode,
[RemoteConnectionDescriptor | SecondEndPointId],
[Encapsulated NotificationRequest],
[Encapsulated EndpointConfiguration])

Delete Connection
This command is used to terminate a connection. As a side effect, it collects statistics on the execution of the connection. If there are more than one gateway involved, the call agent will send the Delete Connection command to each of the media gateways. It is also possible for the Call Agent to delete multiple connections at the same time, using for example wild card options.

Command is represented by:
ReturnCode,
Connection-parameters
DeleteConnection(CallId,
EndpointId,
ConnectionId,
[Encapsulated NotificationRequest],
[Encapsulated EndpointConfiguration])

In some circumstances, a gateway may have to clear a connection, for example because it has lost the resource associated with the connection, or because it has detected that the endpoint no longer is capable or willing to send or receive voice. The gateway terminates the connection by using a variant of the DeleteConnection command.

Audit Endpoint
The Audit EndPoint command can be used by the Call Agent to find out the status of a given endpoint. This feature has been inherited from the switch environment.

Command is represented by:
ReturnCode,
EndpointIdList[
[RequestedEvents],
[DigitMap],
[SignalRequests],
[RequestIdentifier],
[NotifiedEntity],
[ConnectionIdentifiers],
[DetectEvents],
[ObservedEvents],
[EventStates],
[BearerInformation],
[RestartReason],
[RestartDelay],
[ReasonCode],
[Capabilities]]
AuditEndPoint(EndpointId,
[RequestedInfo])

Audit Connection
The Audit Connection command can be used by the Call Agent to retrieve the parameters attached to a connection.

Command is represented by:
ReturnCode,
[CallId],
[NotifiedEntity],
[LocalConnectionOptions],

Restart in Progress
The RestartInProgress command is used by the gateway to signal that an endpoint, or a group of endpoint, is taken in or out of service.

Command is represented by:
ReturnCode,
[NotifiedEntity]
RestartInProgress (EndPointId,
RestartMethod,
[RestartDelay,]
[Reason-code])

3 Protocol At Work
We shall now see how MGCP works with the help of two examples.

3.1 MGCP in all IP Network
Let us now see how MGCP works in the case of all IP network. In the figure RGW = Residential Gateway, CA = Call Agent and EP = Endpoint.
For the sake of discussion, it is assumed that the two EPs, which want to talk with each other, are under the control of the same CA.
In the figure below the solid lines denote the signalling path and the dashed line denotes the media flow. The RGW, CA and database are all part of the IP Network.

3.2 MGCP in PSTN - IP Network
This is the case where the User A is in the PSTN network and he wants to call to an IP phone.
As before the solid lines denote the signaling path and the dotted lines denote the media path.
Point to be noted is that SG and TGW are on the edge of the IP cloud. They interface with both the IP world and SS7 and PSTN world respectively.
Figure 3: MGCP in PSTN-IP Network

1. EP1, which is in the PSTN world, dials the number of EP2.
2. This number reaches SSP through EP1’s local exchange.
3. SSP issues IAM (IAM is the ISUP Initial Address Message) to the CA, which is in the IP world. This IAM reaches SG via STP. SG is connected to IP world on one side and the SS7 world on the other. SG converts the ISUP on SS7 to ISUP on IP and sends the message to CA.
4. CA finds the IP address that serves the dialed number for EP2 from the database.
5. CA now sends the Create Connection command to the TGW to connect to the incoming trunk using CIC. TGW returns the SDP of the connection.
6. CA seizes the incoming trunk (asks the RGW to create call context) and reserves the outgoing trunk by sending the Create Connection to the RGW passing the SDP of TGW.
7. CA now sends Modify Connection to the TGW.
8. CA requests the RGW to ring the called line by sending Notification Request to the RGW.
9. When the CA receives the Ack from the RGW, it issues ACM to the SG.
10. The SG forwards the ACM (ACM is the ISUP Address Complete Message) to the SSP.
11. EP2 goes off-hook, the RGW notifies the CA by sending the Notification Request.
12. Now the voice channel has to be turned into the full duplex mode, the CA does this by sending the Modify Connection command to the TGW.
13. CA then sends the answer message to the SG, the STP forwards this message to the SSP.
14. The EP1 and EP2 are now talking!

4 MEGACO
MEGACO is used between elements of a physically decomposed multimedia gateway, i.e. a Media Gateway and a Media Gateway Controller. Megaco meets the requirements for a MGCP as defined in RFC 2705.

4.1 Connection Model
The connection model for the protocol describes the logical entities, or objects, within the Media Gateway that can be controlled by the Media Gateway Controller. The main abstractions used in the connection model are Terminations and Contexts.
A Termination sources and/or sinks one or more streams. In a multimedia conference, a Termination can be multimedia and sources or sinks multiple media streams. The media stream parameters, as well as modem, and bearer parameters are encapsulated within the Termination.
A Context is an association between a collection of Terminations. There is a special type of Context, the null Context, which contains all Terminations that are not associated to any other Termination. For instance, in a decomposed access gateway, all idle lines are represented by Terminations in the null Context.
The following figure is a graphical depiction of these concepts.
The empty dashed box in each context represents the logical association of Terminations implied by the Context.
4.2 MEGACO Commands

The protocol provides commands for manipulating the logical entities of the protocol connection model, Contexts and Terminations. Commands provide control at the finest level of granularity supported by the protocol. For example, Commands exist to add Terminations to a Context, modify Terminations, subtract Terminations from a Context, and audit properties of Contexts or Terminations. Commands provide for complete control of the properties of Contexts and Terminations. This includes specifying which events a Termination is to report, which signals/actions are to be applied to a Termination and specifying the topology of a Context (who hears/sees whom). Most commands are for the specific use of the Media Gateway Controller as command initiator in controlling Media Gateways as command responders. The exceptions are the Notify and ServiceChange commands.

Following are the various Megaco Commands.

- **Add.** The Add command adds a termination to a context. The Add command on the first Termination in a Context is used to create a Context.
- **Modify.** The Modify command modifies the properties, events and signals of a termination.
- **Subtract.** The Subtract command disconnects a Termination from its Context and returns statistics on the Termination's participation in the Context. The Subtract command on the last Termination in a Context deletes the Context.
- **Move.** The Move command atomically moves a Termination to another context.
- **AuditValue.** The AuditValue command returns the current state of properties, events, signals and statistics of Terminations.
- **AuditCapabilities.** The AuditCapabilities command returns all the possible values for Termination properties, events and signals allowed by the Media Gateway.
- **Notify.** The Notify command allows the Media Gateway to inform the Media Gateway Controller of the occurrence of events in the Media Gateway.
- **ServiceChange.** The ServiceChange Command allows the Media Gateway to notify the Media Gateway Controller that a Termination or group of Terminations is about to be taken out of service or has just been returned to service. ServiceChange is also used by the MG to announce its availability to an MGC (registration), and to notify the MGC of impending or completed restart of the MG. The MGC may announce a handover to the MG by sending it a ServiceChange command. The MGC may also use ServiceChange to instruct the MG to take a Termination or group of Terminations in or out of service.

5 Comparison between MGCP and MEGACO

Now that we have had a brief look at the two protocols, let us make a comparison between them.

<table>
<thead>
<tr>
<th>Features</th>
<th>MEGACO</th>
<th>MGCP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td>Media Gateway Controller</td>
<td>Call Agent</td>
</tr>
<tr>
<td><strong>Call</strong></td>
<td>Terminations within a call context</td>
<td>Endpoints with Connections</td>
</tr>
<tr>
<td><strong>Representative</strong></td>
<td>Any combination of multimedia and conferencing</td>
<td>Point to point and multipoint audio</td>
</tr>
<tr>
<td><strong>Call Types</strong></td>
<td>Text or binary</td>
<td>Text</td>
</tr>
<tr>
<td><strong>Coding</strong></td>
<td>TCP or UDP</td>
<td>UDP</td>
</tr>
<tr>
<td><strong>Internet Protocol</strong></td>
<td>Formal extension process defined within the IETF and the ITU</td>
<td>Less structured process, managed by industry consortia</td>
</tr>
</tbody>
</table>

6 Conclusion

With Megaco you can do everything that you could have done with MGCP and more. Megaco would be primarily used for the Media Gateway Control in the future. MGCP is being tested in many networks today and should soon be operational commercially, but the popularity is Megaco is fast rising. Since MGCP would be soon deployed, so it is likely to stay for some time. However the networks that will appear maybe a year from now will likely use Megaco for Media Gateway Control. So I see that MGCP and Megaco will co-exist for some years, before we mainly have Megaco for Media Gateway Control.

Appendix A

Glossary

<table>
<thead>
<tr>
<th>Terms</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>STP</td>
<td>Signaling Transfer Points</td>
</tr>
<tr>
<td>SP</td>
<td>Signaling Point</td>
</tr>
<tr>
<td>ISUP</td>
<td>ISDN User Part</td>
</tr>
</tbody>
</table>
Appendix B

- **ReturnCode**: ReturnCode is a parameter returned by the gateway. It indicates the outcome of the command and consists of an integer number optionally followed by commentary.

- **EndpointId**: EndpointId is the name for the endpoint in the gateway where command executes.

- **BearerInformation**: BearerInformation is a parameter defining the coding of the data received from the line side.

- **NotifiedEntity**: NotifiedEntity is specifies where the notifications should be sent. When this parameter is absent, the notifications should be sent to the originator of the NotificationRequest.

- **RequestedEvents**: RequestedEvents is a list of events that the gateway is requested to detect and report. Such events include, for example, fax tones, continuity tones, or on-hook transition. To each event is associated an action

- **RequestIdentifier**: RequestIdentifier is used to correlate the request with the notifications that it triggers.

- **DigitMap**: DigitMap allows the Call Agent to provision the gateways with a digit map according to which digits will be accumulated. If this parameter is absent, the previously defined value is retained.

- **SignalRequests**: SignalRequests is a parameter that contains the set of signals that the gateway is asked to apply to the endpoint, such as, for example ringing, or continuity tones. Signals are identified by their name, which is an event name, and may be qualified by parameters.

- **QuarantineHandling**: The QuarantineHandling parameter specifies the handling of “quarantine” events, i.e. events that have been detected by the gateway before the arrival of the NotificationRequest command, but have not yet been notified to the Call Agent.

- **DetectEvents**: DetectEvents specifies a list of events that the gateway is requested to detect during the quarantine period.

- **ConnectionId**: ConnectionId uniquely identifies the connection within one endpoint.

- **SpecificEndpointId**: SpecificEndpointId parameter identifies the responding endpoint when returned from a CreateConnection command.

- **LocalConnectionDescriptor**: LocalConnectionDescriptor is a session description that contains information about addresses and RTP ports, as defined in SDP.

- **SecondEndpointId**: When a SecondEndpointId is returned from a CreateConnection command, the command really creates two connections that can be manipulated separately through ModifyConnection and DeleteConnection commands.

- **SecondConnectionId**: When this is returned from a CreateConnection, it identifies the second connection.

- **LocalConnectionsOptions**: LocalConnectionsOptions is a parameter used by the Call Agent to direct the handling of the connection by the gateway. Some of the fields contained in LocalConnectionsOptions are: Encoding Method, Packetization period, Bandwidth, Type of Service, Usage of echo cancellation and so on.

- **Mode**: Mode indicates the mode of operation for this side of the connection. The mode are "send", "receive", "send/receive", "conference", "data", "inactive", "loopback", "continuity test", "network loop back" or "network continuity test."

- **DetectEvents**: DetectEvents, the list of events that are currently detected in quarantine mode.

- **RestartMethod**: The RestartMethod parameter specified the type of restart of the endpoint. The methods include "graceful" and "forced".

- **RestartDelay**: The parameter is expressed as a number of seconds. If the number is absent, the delay value should be considered null.

- **Capabilities**: The capabilities for the endpoint are similar to the LocalConnectionsOptions parameter and including event packages and connection modes.
Appendix C
Mentioned below are some interesting comments from Speakers during the VON conference.

- In 1998 there were more than 1 trillion minutes of POTS usage.
- The US market for Telephony services is about $250 billions and the global telecom service market is about $800 billion.
- The cross-over for the wide-area data traffic exceeding voice traffic is happening about now, but voice revenues are much greater than the data revenues.
- By 2004, 5% to 20% of long distance calls will be VoIP.
- Circuit switching will be dead by 2005.
- Voice will be only 1% of the total global network traffic by 2008.
- The worldwide market for IP Telephony will grow from $480 million in 1999 to $19 billion in 2004.

References
[2] JAIN MGCP API, version 0.9.