GCP/H.248 Protocol

EFORT
http://www.efort.com

GCP/H.248 Gateway Control Protocol is a (master/slave) protocol for control of gateway functions at the edge of the packet network. Examples of this is IP-PSTN trunking gateways and analog line gateways. The main function of GCP is to allow gateway decomposition into a call agent (call control) part (known as Media Gateway Controller, MGC) - master, and an gateway interface part (known as Media Gateway, MG) - slave. MGC controls call establishment and teardown within MGs The MG has no call control knowledge and only handle making the connections and simple configurations. The GCP protocol involves a series of transactions between MGCs and MGs.
The GCP specification is written such that it provides both a text encoding and a binary encoding of the protocol.
This tutorial introduces the GCP/H.248 protocol with its connection model made of terminations and contexts, as well as its commands, actions and transactions.

1 GCP/H.248 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.

1.1 Termination

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 bearer parameters are encapsulated within the Termination.

Terminations are abstractions of the protocol that represent ports connected to the gateway. Each termination is identified by a termination ID, which is assigned to the termination at the time of termination creation.

Terminations are by nature of two kinds :

- Terminations representing a physical connection, like for example a TDM line, are by nature semi-permanent terminations and they exist as long as they are physically present.
- Ephemeral terminations exist only for the duration of their use. These are for example RTP streams. Ephemeral terminations are created and destroyed as need be.

Terminations can generate signals, tones and announcements for example. Terminations can also be programmed to detect events, such as tones, and trigger notification messages to the MGC.
The MGW also produces statistics from terminations, and the MGC can recall these upon request.

Thus, a termination is described by means of the following information :

- Properties corresponding to attributes of the termination such as mode (receive-only, send-and-receive), state (out-of-service, in-service, in-test).

- Events that the termination may detect (e.g., on-hook, off-hook, flash-hook).
- Signals that the termination may generate (e.g., dial-tone, call-waiting tone)
- Statistics that the termination may return (e.g., number of packets/cells sent, number of packets/cells received, etc.).

According to its type (RTP type, ATM type, Analog line type, TDM type, etc), a termination supports the set or a subset of the information presented above.

In the following example, we consider an Integrated Access Device (IAD) at the customer location emulating a Residential Gateway. This Gateway has on one side an RJ11 port with a permanent analog termination and on the other side RTP termination to deliver voice over IP over DSL line.

The analog termination is a permanent termination which has the following characteristics :

- Properties
  - Mode = send/receive
- Events
  - al/oh, al/on (off hook and on hook events of an analog line)
- Signals
  - al/ri (Ringing tone signal of an analog line when an incoming call is received)
- No Statistics

The RTP Termination is an ephemeral termination which has the following characteristics :

- Properties
  - Mode = send/receive
- Events
  - rtp/pt (payload transition (any change of payload type in an RTP packet may be notified)
- No Signals
- Statistics
  - rtp/ps (number of packets sent since the creation of that termination)
  - rtp/pr (number of packets sent since the creation of that termination)
  - rtp/pl (RTP packet loss ratio)
  - rtp/jit (RTP packet jitter)
  - rtp/delay (RTP packet delay)

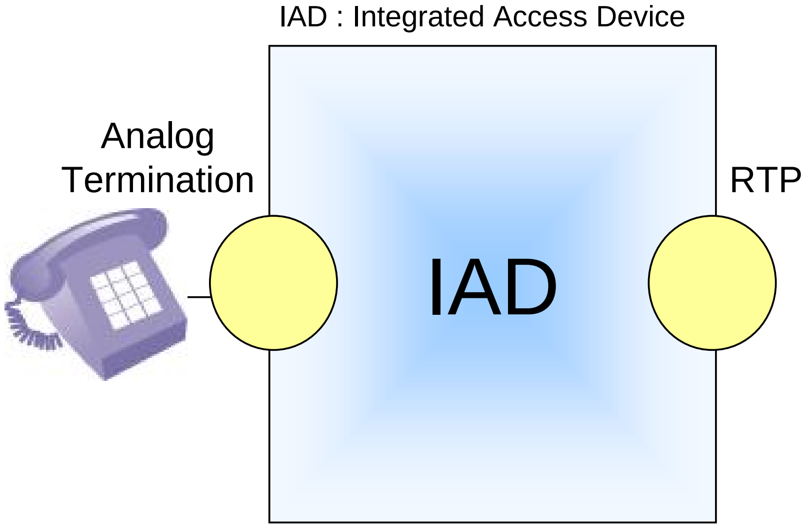

Figure 1 : GCP/H.248 Terminations

GCP/H.248 is designed to be used with a very wide range of gateways. Each type of gateway has different requirements, specifically on the characteristics of the terminations it implements. To cater for this diversity, Megaco has an extension mechanism called a Package. A Package is described in a document, which may come from a standards organization, a consortium or other trade group, a vendor or even an operator.

Packages include definitions of Properties, Events, Signals, and Statistics that are related.

An MG implements sets of terminations. Each termination realizes a set of Packages that specialize that termination to the function it performs. For example, there is an analogue line package that would be implemented on terminations that represent POTS lines.

Several basic packages are described in the MEGACO specification. Among these are the Tone Detection Package, the DTMF Generator Package, the Analog Line Supervision Package and the RTP package, to name just a few. The MEGACO spec also provides guidance for the specification of new packages, which should be registered with the IANA.

1.2 Context

A context is an association between a number of terminations for the purposes of sharing media between those terminations. Terminations may be added to contexts, removed from contexts, or moved from one context to another. A termination can exist in only one context at any time, and terminations in a given gateway can exchange media only if they are in the same context.

A context is an association between a collection of terminations. These terminations will be in contact with each other according to the settings of the call. A typical example of call setup is to have two terminations in the same context.

The Streams of each of the terminations in the Context are connected together, by default in a ‘star’ connection in such a way that whatever is received by any Termination is transmitted by all other Terminations in the Context.

Contexts are multimedia, and the mixing of Streams in the Context is by StreamId (all streams with StreamId=1 are mixed, but are separate from all Streams with StreamId=2).

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 permanent terminations in the null Context. Null context is introduced as a convention where persistent terminations are held when they are not in a real context. Setting up calls means moving terminations from the null context to a new context. When permanent terminations are returned to the null context, they take on their configured default property values.

ROOT termination represents the MG itsell. It is useful for specifying properties of the MG as a device.

The existence of terminations in the same context does not necessarily mean that they can all send data to each other and receive data from each other at any given time, however. The context itself has certain attributes, including the topology, which indicates the flow of media between terminations (which terminations may send media to others or receive media from others). Also, the priority attribute indicates the precedence applied to a context when an MGC must handle many contexts simultaneously, and an emergency attribute is used to give preferential handling to emergency calls.

A context is identified by a ContextID, which is assigned by the MG and is unique within a single MG. As is the case for terminations, MECAGO enables wildcarding when referring to

contexts, such that the all (*) and choose ($) wildcards may be used. The all wildcard may be used by an MGC to refer to every context on a gateway. The choose wildcard ($) is used when an MGC requires the MG to create a new context.
The maximum number of Terminations in a Context is a MG property. Media gateways that offer only point-to-point connectivity might allow at most two Terminations per Context. Media gateways that support multipoint conferences might allow three or more terminations per Context.

Following is a graphical depiction of these concepts. The asterisk box in each of the Contexts represents the logical association of Terminations implied by the Context.

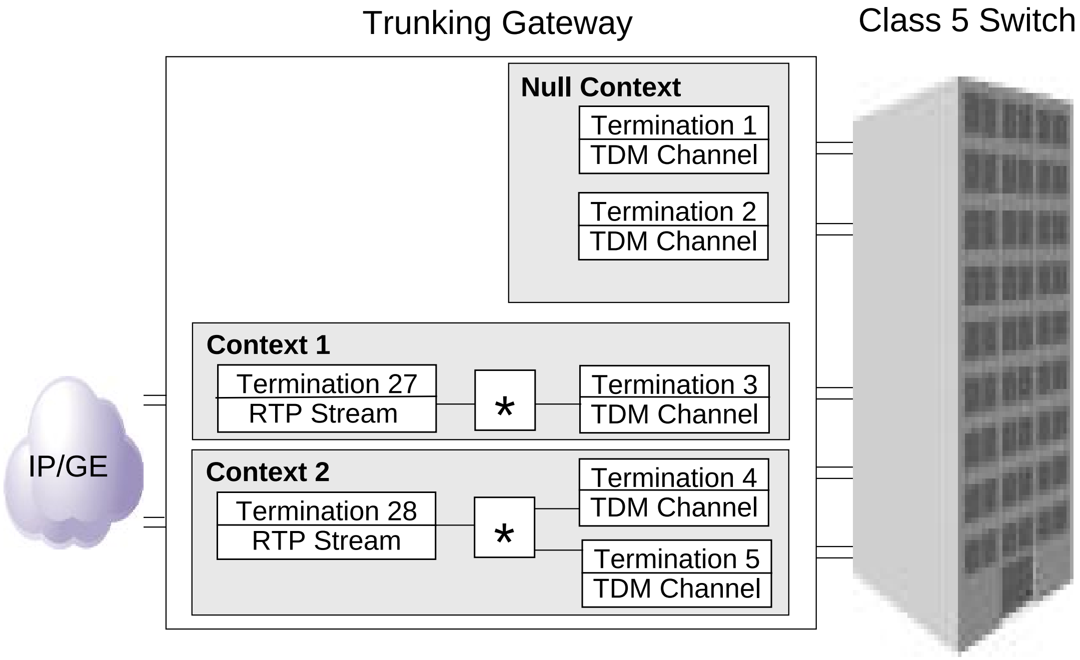

Figure 2 : GCP/H.248 Contexts

2 GCP/H.248 Commands

The GCP/H.248 protocol messages are expressed as a series of commands that operate upon the Terminations and Contexts. Each command includes the name of the command to be executed, the TerminationId(s) on which the command is to be executed, and a series of parameters called ‘Descriptors’. Commands are grouped into ‘actions’ each one of which specifies the Context(s) that will be affected by the commands in the action.
The GCP/H.248 Commands are :

- Add: Add is used to add a termination to a Context. The MGC may specify that a new Context be created to hold the first Termination Added to the Context. The Add command may specify new values for the properties of the termination. It may also specify events to be detected on it, signals to be applied to it, DigitMaps to be used on it, etc. Each of these specifications is described in a Descriptor
- Modify: changes properties, events, signals of a termination or context
- Subtract: A Termination is removed from a Context with a Subtract command. Ephemeral Terminations are destroyed upon being Subtracted. Permanent terminations return to the Null Context. If the last Termination is Subtracted, the Context is destroyed. When a Termination is Subtracted, by default a series of statistics are sent by the MG for the Termination.

- Move: A Termination may be moved from one Context to another atomically with the Move command. Move may also change properties, events, signals, etc, as in an Add command.
- AuditValue : The current values of all properties, list of events, current signals, current statistics and other state of terminations as well as the TerminationIds and the ContextIds to which they belong may be read by the MGC using the AuditValue command.
- AuditCapabilities : The AuditCapabilities command is used by an MGC to retrieve the possible values of properties, signals, and events associated with one or more terminations. At first glance, this command may appear very similar to the AuditValue command. The difference between them is that the AuditValue command is used to determine the current status of a termination, whereas the AuditCapabilities command is used to determine the possible statuses that a termination might assume. For example, AuditValue would indicate any signals that are currently being applied by a termination, where AuditCapabilities could indicate all the possible signals that the termination could apply if required.
- Notify: The MG may detect asynchronous conditions the MGC may wish to know about. The MGC may specify a list of such Events, detection of any one of which will cause the MG to notify the MGC of its occurrence. Off-hook on an analogue line is an example of such an Event; detection of a DTMF digit is another common Event, each of the DTMF digits being a separate Event. Events are generated by a termination and detected by the MGW, which can then forward them to the MGW controller. An MGW controller will request to be informed of certain events using the modify message. The MGW will inform the controller of the occurrence of an event using the notify message.
- ServiceChange : The MG and the MGC report major state change to each other with the ServiceChange command. The MG uses ServiceChange to create a control association between itself and its controlling MGC. Either the MG or the MGC can put terminations out-of-service with this command.

VERB	DIRECTION
Add	MGC →MG
Modify	MGC →MG
Subtract	MGC →MG
Move	MGC →MG
AuditValue	MGC →MG
AuditCapability	MGC →MG
Notify	MG→MGC
ServiceChange	MG →MGC ; MGC →MG

MG : Media Gateway
MGC : Media Gateway Controller

Figure 3 : GCP/H.248 Commands

3 GCP/H.248 Transactions

Commands are grouped into 'Actions' that operate on a Context or set of Contexts. Within an Action, Commands are executed sequentially. Each Command operates on a Termination or set of Terminations. Actions are grouped into Transactions that are sent as a unit to the corresponding entity. Within a Transaction, Actions are executed in order.

The sender assigns a TransactionId to each Transaction command with the fully specified TerminationIds.
For each command, the response also may include Descriptors. The Descriptors that are returned depend on what the command was, and how the parameters in the Descriptors were specified.
The recipient acknowledges transactions when they are executed. The reply includes the TransactionId specified in the request, and for each action, and each command within the action, a reply is normally generated.

At the first failing Command in a Transaction, processing of the remaining Commands in that Transaction stops. If a command contains a wildcarded TerminationID, the command is attempted with each of the actual TerminationIDs matching the wildcard. A response within the TransactionReply is included for each matching TerminationID, even if one or more instances generated an error. If any TerminationID matching a wildcard results in an error when executed, any commands following the wildcarded command are not attempted. Commands may be marked as "Optional" which can override this behaviour - if a command marked as Optional results in an error, subsequent commands in the Transaction will be executed. If a command fails, the MG shall as far as possible restore the state that existed prior to the attempted execution of the command before continuing with command processing.

A TransactionReply includes the results for all of the Commands in the corresponding TransactionRequest. The TransactionReply includes the return values for the Commands that were executed successfully, and the Command and error descriptor for any Command that failed.

TransactionPending is used to periodically notify the receiver that a Transaction has not completed yet, but is actively being processed. Applications should implement an application level timer per transaction. Expiration of the timer should cause a retransmission of the request. Receipt of a Reply should cancel the timer. Receipt of Pending should restart the timer.

TransactionRequest (TransactionID {
ContextID$_1$ { Command$_1$, Command$_2$, .., Command$_N$ }, *Action*
ContextID$_2$ { Command$_1$, O-Command$_2$, .., Command$_N$ },
ContextID$_M$ { Command$_1$, Command$_2$, .., Command$_N$ }   })

TransactionReply (TransactionID {
ContextID$_1$ { Response$_1$, Response$_2$, .., Response$_N$ },
ContextID$_2$ { Response$_1$, Response$_2$, .., Response$_N$ },
ContextID$_M$ { Response$_1$, Response$_2$, .., Response$_N$ }   })

TransactionPending (TransactionID { })

Figure 4 : GCP/H.248 Transactions

When a MG reports an error to a MGC, it does so in an error descriptor. The error descriptor is returned in a transaction reply when the requested command could not be executed. The error descriptor consists of an error code, optionally accompanied by a textual description of the error. The error codes are registered at the IANA. The list below contains the various

MEGACO error codes and related descriptions. New error codes can be specified, in which case they must be registered with the IANA.

The identified error codes are:
400 : Bad Request
401 : Protocol Error
402 : Unauthorized
403 : Syntax Error in Transaction
406 : Version Not Supported
410 : Incorrect identifier
411 : The transaction refers to an unknown ContextId
412 : No ContextIDs available
421 : Unknown action or illegal combination of actions
422 : Syntax Error in Action
430 : Unknown TerminationID
431 : No TerminationID matched a wildcard
432 : Out of TerminationIDs or No TerminationID available
433 : TerminationID is already in a Context
440 : Unsupported or unknown Package
441 : Missing Remote Descriptor
442 : Syntax Error in Command
443 : Unsupported or Unknown Command
444 : Unsupported or Unknown Descriptor
445 : Unsupported or Unknown Property
446 : Unsupported or Unknown Parameter
447 : Descriptor not legal in this command
448 : Descriptor appears twice in a command
450 : No such property in this package
451 : No such event in this package
452 : No such signal in this package
453 : No such statistic in this package
454 : No such parameter value in this package
455 : Parameter illegal in this Descriptor
456 : Parameter or Property appears twice in this Descriptor
471 : Implied Add for Multiplex failure
500 : Internal Gateway Error
501 : Not Implemented
502 : Not ready
503 : Service Unavailable
504 : Command Received from unauthorized entity
505 : Command Received before Restart Response
510 : Insufficient resources
512 : Media Gateway unequipped to detect requested Event
513 : Media Gateway unequipped to generate requested Signals
514 : Media Gateway cannot send the specified announcement
515 : Unsupported Media Type
517 : Unsupported or invalid mode
518 : Event buffer full
519 : Out of space to store digit map
520 : Media Gateway does not have a digit map
521 : Termination is "ServiceChanging”
526 : Insufficient bandwidth
529 : Internal hardware failure
530 : Temporary Network failure
531 : Permanent Network failure

581 : Does Not Exist

References

IEFT RFC 3525. C. Groves, M. Pantaleo, LM. Anderson, T. Taylor, “Gateway Control Protocol Version 1”, June 2003.