

Comparing versions of MGCP
 Author: Nancy Greene
 Date: March 10/99

- Updated with comments from Paul Sijben and Christian Huitema
- Updated with requirements mapped to each row in the table

The table below compares the protocols MDCP[1] (author Paul Sijben et al), MGCP [3,4] (authors Christian Huitema et al), and MGCP-modified [2] (authors Fernando Cuervo et al). It also attempts to map requirements from [5] to specific functionality in the protocols that satisfies these requirements.

The goal of the table is to highlight similarities and differences of the approaches, and make it easier to decide what is important and what definitions and messages should be used to build the megaco MGCP protocol.

References

[1] P. Sijben et al, MDCP, <http://www.ietf.org/internet-drafts/draft-sijben-megaco-mdcp-01.txt>
 [2] F. Cuervo et al, 2-ended connection model for MGCP, <http://www.ietf.org/internet-drafts/draft-cuervo-megaco-mgcp-00.txt>
 [3] C. Huitema et al, Proposed MGCP, <http://www.ietf.org/internet-drafts/draft-huitema-megaco-mgcp-v0r1-0x.txt>
 [4] C. Huitema et al, <http://www.ietf.org/internet-drafts/draft-huitema-megaco-mgcp-firewall-00.txt>
 [5] N. Greene et al, Megaco Requirements, <http://www.ietf.org/internet-drafts/draft-ietf-megaco-reqs-01.txt> (or <ftp://standards.nortelnetworks.com/megaco/docs/minn99/draft-ietf-megaco-reqs-01.txt>)
 [6] M. Holdrege, MSF requirements input to MEGACO, <http://www.ietf.org/internet-drafts/draft-ietf-megaco-msf-reqs-00.txt>

NOTE: since we didn't label paragraphs and bullets in [5], it is hard to refer to them. What I have done is label the paragraphs sequentially per section as section# p1, section# p2 etc. for paragraphs, and the bullets per section as section# *1 for the first bullet, section# *2 for the second, etc. For example, the 5th paragraph in section 6 is labeled as section 6 p5, and the 4th bullet in section 3 is labeled as section 3 *4 .

NOTE also that this is NOT a complete table of all requirements and how the protocol versions compare. This still needs to be done.

Issue/Requirement addressed	MDCP-Paul Sijben [1]	MGCP - Fernando Cuervo [2]	MGCP -Christian Huitema [3]
Issue1. Definitions ([5] section 3)	[1]1.Resource: - Properties, e.g. resourceId, package for media edgepoint, script for signalling edgepoint - Actions it supports - Events it can generate - Is allocated/reserved before it is used - Does not exist outside a call	[2]1.Edgepoint: - properties are set using MediaDescriptor, SignallingDescriptor - actions are described in the SignallingDescriptor - events are described in the SignallingDescriptor - denotes an interface - can exist outside a call - particular properties for a particular call are associated with an externalConnectionId	[3]1.Endpoint: - - - actions are in a script supported events and actions are defined by the endpoint type (DS0, analog, conference) and by the set of packages that it supports. - events activated in NotificationRequest - particular media-

		for that call	processing properties of an endpoint for a particular call are associated with a particular connectionId - can exist outside a call
Issue2. Types of endpoint/edge points, and associating parameters with them. ([5] section 3 *1, section 6.3, section 8.1, [6] Connectivity Reqs)	[1]2.Media edgepoint resource - SCN Bearer channel (e.g.DS0) - SCN analog - ATM VC - RTP Stream - UDP port	[2]2.Edgepoint with a MediaDescriptor (set using MediaAdaptationConfig) - when second edgepoint is RTP port, it is listed as an alias for the first endpoint - when second edgepoint is something else, e.g. ATM, then it can carry its own MediaDescriptor, SignallingDescriptor if necessary	[3]2.Endpoint with L: params - no separate endpoint when second endpoint is an RTP port - separate endpoint possible for hairpin, for ATM
Issue3. Handling of internal resources ([5] section 5.1.2)	[1]3.Internal media resource - created using a connect command - e.g. conference bridge - connect command can have more than 2 edgepoints, internal resources in it, so only need one connect command to set up a call from one edgepoint to another through a bridge.	[2]3.Internal edgepoint that the MGC knows about - if an MG wants to expose an internal resource that it has, then the MGC can use it to ensure that the particular exposed resource is included in a call - to do this would take 2 CreateMGConnections - one from one edgepoint to the resource, and one from the resource to the other edgepoint - if MG did not want to expose this internal resource, it could still use it, but MGC would not know about it.	[3]3. - - assumes that each endpoint has a set of "resources" that it can support. This is defined by the type of the endpoint (DS0, analog, conference...) and by the packages that it supports (e.g. announcement, script). - gateways may include different types of endpoints, e.g. an explicit IVR server, that can be explicitly used by an MGC.
Issue4. Handling of signalling ([5] section 4 *a, section 5.5.3, section 8.1)	[1]4.Signalling edgepoint resource - MDCP signalling edgepoint management signalling edgepoint - H.245 signalling edgepoint - Signalling backhaul - FAS edgepoint (just relays msgs)	[2]4.Edgepoint with a SignallingDescriptor (set using SignallingConfig)	[3]4. - Per packet signalling such as Q.931 is explicitly out of scope, relies on "sigtran" for signalling backhaul. - Analog and CAS signalling handled through event packages.
Issue5. Support of scripts ([5], section 5.2,	[1]5.Script = property of an edgepoint - you can Download it, Delete, start it, stop it - if it is downloaded to a media or	[2]5.Script is assigned to an edgepoint using SignallingConfig	[3]5. - Script is assigned to an endpoint using NotificationRequest

5.6)	edgepoint, the scope of the script is that edgepoint, if the commands apply to the management edgepoint, the scope is the entire MG		- Notification Request syntax allows for basic scripting facility providing digit accumulation (digit map) and reflex action (one level of embedding).
Issue6. Handling dial plans ([5], section 5.2)	[1]6.Dial plan is a type of script	[2]6.Same as MDCP	[3]6.Digit map D: is a regular expression. (Explicit refusal to use a "dial plan" notation, digit map is just a regular expression.)
Issue7. Conference concept	[1]7.Conference = one or more calls	[2]7. Need to define how a conference relates to a call	[3]7. – A conference is obtained by terminating several connections on the same endpoint. (Explicit refusal to use an explicit concept in the MG, so as not to create feature interaction.)
Issue8. Call concept	[1]8.Call - identifies resources & connections in use for a point to point connection, may involve more than one connection	[2]8.A CallId groups together all connections and media adaptations used to build the end-to-end call-	[3]8. A "call-id" is associated to connections for accounting and logging purposes. This is essentially an arbitrary tag, provided by the MGC. It is possible to delete all connections that belong to a given call-id.
Issue9. Connections internal to an MG	[1]9.Connection - unidirectional, each connection is known by a pair of resource ids	[2]9.MG-connection - is known by the two edgepoints it connects - is internal to a MG	[3]9. Internal connections can be established, e.g. for hairpin purposes. Connections can be ½ duplex or full duplex (mode parameter)
Issue10. Connections external to MG	[1]10. The external connection: - denoted by the pair (myAddress, remoteAddress) in the appropriate edgepoints. - QoS, encryption parameters pertain to the world outside the MG	[2]10.External Connection - ExtConnectionId - runs from one MG edgepoint out towards a network - an MG assigned id that represents a modification of the media configuration of an endpoint	[3]10.Connection - External connections are really ½ connections, running from an endpoint out towards a network. each MG defines its receiving parameters, and is informed of the other MG parameters. - Parameters defined in SDP: address, ports,

			<p>algorithms, encryption.</p> <ul style="list-style-type: none"> - Connections are locally identified by a binary id.
Issue11.Packages ([5], section 5.6, section 8)	[1]11.Packages: Announcement Player, SCN Voice, Call Progress Tones, DTMF detector, DTMF generator, Echo Cancellor, Media Crypt, RTCP, Scripting	[2]11.Same, although echo canceller is an edgepoint property as with [3]	[3]11.Same, except for "echo canceller" which is an endpoint property.
Issue12. Registration of MG with an MGC ([5] section 6.4)	[1]12.Registration happens between an MGC and an MG	[2]12.Not addressed, but would support registration between and MGC and an MG - to exchange configuration info, ...	[3]12. Registration is performed through the "security" layer, results in security association. MG can announce its presence through "Restart in progress" command.
Issue13. Transport ([5] section 5.7)	<p>[1]13.</p> <ul style="list-style-type: none"> - explicitly ack all msgs per conference, using send/rcv seq #s - so, disallows out of sequence delivery of msgs for a single conference 	<p>[2]13.At megaco application level:</p> <ul style="list-style-type: none"> - application RoundTripTime (RTT) - sequencing of commands for an edgepoint - embedding of commands - session - be able to bring down a call thread and start over - application level windowing - request/response matching <p>At Transport Interface</p> <ul style="list-style-type: none"> - open, close, send, flush <p>Transport Protocol</p> <ul style="list-style-type: none"> - Aggressive retransmit policy, interval initially very close to - Very limited retransmit attempts (say order of 3 - 5 tries). - Retransmits exponentially back off to avoid undue congestion. - Re-ordering internal to the reliable UDP protocol handler, over a small window. - No adaptive behaviour. 	<p>[3]13</p> <p>Transaction-based transport performed as part of the application:</p> <ul style="list-style-type: none"> - explicit acknowledgement of all commands, - handling of RTT and aggressive repetition, with randomized exponential backoff. - request response matching through transaction-id - 3-ways handshake for management of copies. <p>Ordering provided within the application:</p> <ul style="list-style-type: none"> - precedence rule between connection commands, - explicit sequencing of notification request and notification through notification id.
Issue14. Control association ([5] section 6.4)	[1]14.Connection control session	[2]14.Not currently defined, but assumes it is needed	[3]14 Control association is mapped onto security association (IPSEC). "Null" audit commands can be used for testing liveness of MG.
Issue15.	[1]15.	[2]15.	[3]15. -

<p>Resource Management ([5] section 6.5)</p>	<ul style="list-style-type: none"> - Audits, etc provided with Query message - Resource management control session - MG can use this to authenticate MGC and vice versa 	<ul style="list-style-type: none"> - Audits etc not defined - would use same methods as in [1], [3] - could use ExternalConnectionIds for a resource management control session 	<ul style="list-style-type: none"> - Connection control allows flexible resource management, from handoff policy (resource managed by the MG) to detailed inventory (resource explicitly allocated by MGC). - Allocation is always reported to MGC. Audit of endpoint and connections can be used to track allocated resource. - Error codes can be returned when the gateway is out of resource.
<p>Issue16. Failover ([5] section 5.7, 6.2)</p>	<p>[1]16.Failover procedures</p> <ul style="list-style-type: none"> - timeout on heartbeat - secondary discovers - voluntary handover 	<p>[2]16.Not currently defined, but assumes it is needed</p>	<p>[3]16. Association between MGC & MG is defined at security level. Any MGC can take over if it has the proper security credential and "distributed systems" capability (stable memory). An MGC that takes over without proper memory can use a combination of Audit and DeleteConnection commands to obtain a "clean slate."</p>
<p>Issue17. All or nothing execution of a series of commands (transport reqt)</p>	<p>[1]17.Multiple commands are allowed within one message. If a command fails, the rest in that message will not be considered.</p> <ul style="list-style-type: none"> - same thing for connect command: if any connection setup fail, then the whole connect command fails. 	<p>[2]17.Embedding commands to ensure all or nothing execution of them</p> <ul style="list-style-type: none"> - explicit embedding of the command so that parsing will find start and stop of embedded command 	<p>[3]17.Embedding commands to ensure all or nothing execution of them</p> <ul style="list-style-type: none"> - explicit embedding of the command so that parsing will find start and stop of embedded command has been proposed (see [4])
<p>Issue18. Messages (covered in 19-28)</p>			
<p>Issue19. configuring of endpoints/edge points for media adaptation</p>	<p>[1]19. Done using properties associated with an edgepoint -</p>	<p>[2]19.MediaAdaptationConfig</p> <ul style="list-style-type: none"> - to tell the MG that media flows entering or leaving the MG need media adaptation - can include one or more aliases for the edgepoint being configured 	<p>[3]19. Uses Local&RemoteConnection Session Descriptions to associate attributes with the connection associated with an endpoint - these</p>

		<ul style="list-style-type: none"> - Each alias is an IP or transport address for the edgepoint being configured - Used on its own when MG is adapting to IP 	<p>are params in Create/ModifyConnection. Uses LocalConnectionOption to provide directives to the MG.</p> <p>also EndpointConfiguration currently used to set u-law or a-law</p>
Issue21. configuring of endpoints/edgepoints for signalling	[1]21. Done using properties associated with an edgepoint, when edgepoint is created - can also be done later using Modify command	<p>[2]20. SignallingConfig for an edgepoint</p> <ul style="list-style-type: none"> - to tell MG to use CAS or FAS if an edgepoint is to be handled by a CallAgentId (CallAgentId=NotifiedEntity in [3]) - SignallingDescriptor describes the signalling - e.g., digit map, pointer to a script - Embedded ApplySignal 	<p>[3]20.NotificationRequest for an endpoint</p> <ul style="list-style-type: none"> - uses D: for digit map - specifies list of events that should be detected and associated actions.
Issue21. Modifying properties of endpoints/edgepoints	[1]21. Modify	<p>[2]21. ModifyExtConnMode</p> <ul style="list-style-type: none"> - used to modify mode of external connection when no MG-connection is set up - this is the case when adapting to IP. 	[3]21. ModifyConnection
Issue22. asking MG to apply a signal	[1]22. Request, RequestConfirm, RequestReject - request an action	<p>[2]22.ApplySignal</p> <ul style="list-style-type: none"> - the signal descriptor in this command is constrained by the signallingDescriptor 	<p>[3]22.S:</p> <ul style="list-style-type: none"> - signals listed with the S: line are constrained by the package in use on the MG - generation of signals is tightly coupled with detection of events.
Issue23. indication of event ([5] section 5.3)	[1]23.Indication - indication of an event	<p>[2]23. NotifyEvent</p> <ul style="list-style-type: none"> - once an edgepoint is provisioned with a SignallingDescriptor, it can generate events (e.g. off-hook) 	[3]23. Notify
Issue24. Connections	[1]24. Reserve Connect Disconnect Modify Delete	<p>[2]24. CreateMGConnection</p> <ul style="list-style-type: none"> - used to set direct connections between edgepoints that provide external connections embedded - MediaAdaptationConfig, SignallingConfig for each edgepoint - NOT used when MG is adapting to IP <p>ModifyMGConnection</p> <ul style="list-style-type: none"> - for changing mode of an edgepoint - embedded 	<p>[3]24. CreateConnection ModifyConnection DeleteConnection</p>

		<p>ModifyExtConnMode DeleteMGConnection - embedded EdgepointReset, up to two, one for each edgepoint</p>	
<p>Issue25. Returning edgepoints/endpoints to a default config ([5], section 5.8, last paragraph)</p>	<p>[1]25. When a call goes away, edgepoints associated with it go away. - resources and their default properties are defined, but they do not explicitly exist as edgepoints until they are reserved or made part of a connection.</p>	<p>[2]25.EdgepointReset - to release external connection ids for a call</p>	<p>[3]5. DeleteConnection - frees all resources associated with the connection. Deleting all connections associated to an endpoint restore media processing attributes to their default values. NotificationRequest is used to reset the signals and requested events to a defined value.</p>
<p>Issue26. Resource auditing ([5] section 5.7 p5, p6, section 6.1)</p>	<p>[1]26.Query</p>	<p>[2]26.AuditEndpoint</p>	<p>[3]26. AuditEndpoint AuditConnection</p>
<p>Issue27. Control association between MGC and MG ([5] section 6.4)</p>	<p>[1]27. Registration - start a control session RegistrationResponse, RegistrationReject Termination - terminate control session</p>	<p>[2]27. not yet defined</p>	<p>[3]27. Delegated to the security association layer.</p>
<p>Issue28.indication of reboot ([5] section 5.7, p7)</p>	<p>[3]28. Termination</p>	<p>[3]28. not yet defined</p>	<p>[3]28. RestartInProgress</p>