# Owners and List of Contacts

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zahid Anwar</td>
<td><a href="mailto:anwar@uiuc.edu">anwar@uiuc.edu</a></td>
<td>2826160-298</td>
<td>PhD Candidate</td>
</tr>
</tbody>
</table>

Software Research Group
Table of Contents

Owners and List of Contacts ................................................................. 2
Signoffs ............................................................................................... 5
Revision History ................................................................................. Error! Bookmark not defined.
Table of Contents ............................................................................. 3
Introduction ......................................................................................... 5
Overview ................................................................................................ 5
Overall Conceptual Model ................................................................... 5
Gateway Handler .................................................................................. 6
Call Handler ............................................................................................ 6
Decoding Process .................................................................................. 7
GW Handler Queue ................................................................................ 7
CH Task Queue ..................................................................................... 7
Manager ................................................................................................ 7
Gateway Handler Detail ......................................................................... 8
Gateway Controller ............................................................................... 9
Decoder .................................................................................................. 9
Encoder .................................................................................................. 9
Encoding Responses ............................................................................. 10
Piggy-Backing ....................................................................................... 10
Encoder Task Queue ............................................................................. 10
Sent Cmpd Queue ................................................................................ 10
Rcvd Cmpd Queue ................................................................................ 10
Rsp History Queue ............................................................................... 10
GW Controller Task Queue ............................................................... 10
Rcvd Rsp Queue .................................................................................. 10
Transaction Id Map .............................................................................. 11
Termination Object ............................................................................... 11
TermState .............................................................................................. 11
EncoderTask .......................................................................................... 11
Activity Structure .................................................................................. 11
Components of Call Handler ............................................................... 12
CallState ............................................................................................... 13
Use Cases, Sequence Diagrams And Contracts ................................... 14
Use Case 1 ............................................................................................. 14
Sequence Diagram 1 ............................................................................ 14
Contracts ............................................................................................... 15
Use Case 2 ............................................................................................. 15
Sequence Diagram 2a .......................................................................... 16
Sequence Diagram 2b .......................................................................... 17
Contracts ............................................................................................... 17
Interaction (Collaboration Diagrams) .................................................... 20
Startup ................................................................................................. 23
Decoding ............................................................................................... 24
Processing ............................................................................................. 25

Software Research Group
Soft Switch Design Document

Encoding ................................................................. 26

Procedures and Termination State Machine .......................... 24
State Unregistered .................................................. 24
If Get Capability Task .............................................. 25
State Onhook ........................................................ 25
State Dialing .......................................................... 27
State Digits Collected .............................................. 28
State Ring Back ....................................................... 30
State Active ............................................................ 34
State Active And Onhook ......................................... 36
State Waiting For Onhook ......................................... 38
State Dialing .......................................................... 39

Procedures and Call Handler State Machine ......................... 43
Start ................................................................. 43
Wait for Setup Complete .......................................... 44
Wait For Ring Complete ......................................... 44
Wait for Called Term Off Hook ................................. 45
Active ................................................................. 45

Application Programming Interface ................................. 45
Call Agent API .................................................... 45
Queue API .......................................................... 47
GatewayController API ........................................ 47
Termination API .................................................... 48
GatewayHandler API ........................................... 48
Manager API ....................................................... 49

Pending Issues ........................................................ 50
**Introduction**

**Overview**

This document discusses the H.248 Media Gateway Controller Components at a high level of abstraction. It deals with both the architectural and the functional details. Note that the terms Megaco and H.248, and similarly, the terms Media Gateway Controller (MGC) and Call Agent will be used interchangeably in this document.

So far, the MGC conforms Level 3 of IETF specified compliance criteria excluding conference calling and call-waiting.

**Overall Conceptual Model**

The Major components in the Call Agent are:

- Gateway Handler
- Call Handler
- Decoding Process
- Gateway Handler Queue
- Call Handler Task Queue
- Manager
Figure: Overall Class Hierarchy

Note: The class diagram only shows relationships between the major classes. Some Queues and other subsidiary classes that are not shown are explained below

Gateway Handler

Gateway Handler is responsible for controlling gateways. The Gateway Handler deals with all the messages except those that are specific to connections. A separate instance of Gateway Handler is used for managing, control, and state maintenance of each Media Gateway.

Call Handler

Call Handler is responsible for establishing and deleting connections. Call Handler receives messages from gateway controllers through CH Task Queue. Depending upon the type of
message the call handler activates the state machine of the concerned call object. Call handler can pass messages to gateway controllers through GW Handler Queue.

**Decoding Process**

Header Decoder Process is a centralized decoding resource. On receiving a message Header Decoder maps the Alias to get Gateway Handler Id, and passes it to the appropriate Gateway Handler. The Header Decoder is also responsible for the authentication and parsing of the message, which is either in ASN or EBNF form.

**GW Handler Queue**

GW Handler Queue is associated with every Gateway Handler. All messages destined for the Gateway Handler are pushed in this queue. Decoder retrieves these messages, It passes Request to RCVD Cmd Queue, Responses to RCVD Rsp Queue and Tasks to Call Handler Task Queue.

**CH Task Queue**

CH Task Queue contains the Messages relating to the Connections to be handled. This is a central queue shared by all the gateway controllers to send tasks to the Call Handler.

**Manager**

Manager acts as a data store. It keeps data in the form of maps. Each map contains a key value pair. The rest of the components reference the manager when they want to a access a pointer to any other component.
Figure: Overall Call Agent Architectural View

Gateway Handler Detail

The components in Gateway Handler are:

- Gateway Controller
- Decoder
- Encoder
- Encoder Task Queue
• Sent Cmd queue
• RCVD Cmd Queue
• RCVD Rsp Queue
• Rsp History Queue
• Gateway Controller Task Queue
• Transaction Id Map
• Termination Objects

**Gateway Controller**

Gateway Controller is responsible for controlling a provisioned gateway. For each termination in the gateway, gateway controller has a state machine to control the termination. This state machine can be activated when

a. A message from the gateway is received, or

b. A task from call handler is received

Gateway Controller is responsible for receiving gateway messages through **RCVD Command Queue** and **RCVD Responses Queue** and call handler messages through Gateway Controller Task Queue, and activating the state machine for the termination.

Gateway Controller is the entity that receives the messages from

- RCVD COMD QUEUE
- RCVD RSP QUEUE
- GW CONTROLLER TASK QUEUE

It interacts with Termination objects which are responsible for the management of the gateway endpoints. The Termination objects have state machines used to manage message specific to the terminations.

**Decoder**

The Decoder deals with Commands, Responses and Tasks. It retrieves messages from the Gateway Handler Queue, and passes Commands to RCVD Cmd Queue, Responses to RCVD Rsp Queue and Tasks to Call Handler Task Queue.

**Encoder**

The Encoder deals with both Commands and Responses. Encoding commands
Send Cmd Queue contains the commands to be sent, Encoder gets the Command, using the information in Encoder Task Queue encodes it and sends it through the UDP API.

Encoding Responses

Rsp History Queue of each gateway controller contains the responses to be sent, Encoder gets the Response, using the information in the Encoder Task Queue encodes it and sends it through the UDP API.

Piggy-Backing

The Megaco MGC does not support piggy backing.

Encoder Task Queue

Encoder Task Queue contains indications (a structure containing Transactions-IDs to be piggy-backed) and Queue Information.

Gateway controller when wishes to send several messages in the same UDP Packet, it notifies by passing an indication and Queue Information to the Encoder Task Queue. Encoder retrieves Messages corresponding to these Transaction-Ids and Queue information, piggybacks them into a single UDP packet.

Sent Cmd Queue

Sent Cmd Queue contains the commands to be sent, Encoder gets the Command, sends it through the UDP API. The ossencode ASN function is used at the UDP API. It is intended that this encoding process should be moved back into the Encoder.

Rcvd Cmd Queue

Rcvd Cmd Queue contains Commands to be handled. The Commands contain identifiers in the form of termination@gateway; Decoder receives the message from Gateway Handler Queue, decodes it as Command or Response, and pushes the command in the respective Rcvd Cmd Queue of the Gateway Handler accordingly.

Rsp History Queue

Rsp History Queue contains responses generated by the Gateway Controller. Encoder gets the Command, sends it through the UDP API. The ossencode ASN function is used at the UDP API. It is intended that this encoding process should be moved back into the Encoder.

GW Controller Task Queue

GW Controller Task Queue contains Tasks sent by the Call Handler. Decoder retrieves messages from the GW Controller Task Queue; gateway controller retrieves them, processes them accordingly, appends the Transaction-ID to it, and pushes commands to the Sent Cmd Queue.
**Rcvd Rsp Queue**

Rcvd Rsp Queue contains the Responses to be handled.

**Transaction Id Map**

Transaction Id map is a map of termination Id and transaction ids of the messages sent from that termination whose responses have not been received.

**Termination Object**

A termination object is the representation of a termination of the gateway. It handles all the activities related to the termination.

It contains the following data

1. Termination Identifier
2. Gateway Controller Identifier
3. Termination state

Follow enumerated data types are used by termination object

**TermState**

Defines the state of termination

Following are the structures used by the termination object

**EncoderTask**

Encoder task is a list of Activity structures. All the activity structures are piggy-backed in one message.

**Activity Structure**

QueueType

**Identifies the queue from which the message is to taken. Possible choices are**

- Sent Message Queue.
- Response History Queue
Components of Call Handler

The components of the Call Handler are:

- Service Call Handler Task Queue
- Call ID Generator
- Call Object (1…. N)

Call ID Generator generates a Call ID. Call ID or Call Identifier is a hexadecimal string, with a maximum of 32 characters.

Call Object are instantiated by Service Call Handler Task Queue. The state machine corresponding to a Call is running in a Call Object. It receives messages from the Service Call Handler Task Queue, and activates State Machine accordingly.
Following information is stored in Call Object

- CallId
- CallState
- Manager Info

Following enumerated data type are used by the Call Object

**CallState**

Defines the state of the Call Object.

---

**Figure: Call Handler Architecture**
USE CASES, SEQUENCE DIAGRAMS AND CONTRACTS

USE CASE 1

<table>
<thead>
<tr>
<th>Name:</th>
<th>Initialize Media Gateway Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Remote User</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary and Essential</td>
</tr>
<tr>
<td>Description:</td>
<td>A remote user through an SNMP Agent activates the MGC to start up and initialize all its Objects in order to start processing messages from different Gateways</td>
</tr>
</tbody>
</table>

**ACTOR**
1. The use case begins when the remote user wishes to initialize the call agent
2. The client creates the Text Encoder
3. The system displays a menu to the user giving the user additional option of adding gateways and terminations etc to the callagent
4. The user makes any necessary changes to the configuration according to his requirements

**SYSTEM**
2. The callagent starts up reading its configuration from a configuration file
3. The system displays a menu to the user giving the user additional option of adding gateways and terminations etc to the callagent
5. The system updates itself according to the users changes

SEQUENCE DIAGRAM 1

The above use case indicates the following list of the events, the Megaco Text Encoding system must respond and operate on

---

Remote User

Must

Create ( Instantiate )

Optional

Change Configuration

System MGC
## CONTRACTS:

1. **Name**: Create  
   **Responsibilities**: Initializes the system  
   **Type**: System  
   **References**: Interaction Diagram : System Startup  
   Use Case : Initialize Media Gateway Controller  
   **Exception**: If system could not be initialized, then stop further and indicate error  
   **Notes**: NIL  
   **Post_Conditions**:  
   • The Call Agent is started  
   **Pre_Conditions**: NIL

2. **Name**: Change Configuration  
   **Responsibilities**: Used to Add, Subtract, Start, Stop Gateway Handlers, Terminations, report calls in progress read configuration files and change digitmaps  
   **Type**: System  
   **References**: Interaction Diagram : System Startup  
   Use Case : Initialize Media Gateway Controller  
   **Exception**: An invalid command or a command in incorrect sequence will be reported as error  
   **Notes**: This is an optional operation.  
   **Post_Conditions**:  
   • The system configuration is updated  
   **Pre_Conditions**:  
   • The configuration should be updated in correct sequence

## USE CASE 2

<table>
<thead>
<tr>
<th>Name</th>
<th>Process Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Remote Client (Media Gateway)</td>
</tr>
<tr>
<td>Type:</td>
<td>Optional</td>
</tr>
<tr>
<td>Description:</td>
<td>A message is received on the network by the MGC. The MGC has to read it, authenticate it, process it, perform any error-checking if necessary and reply to the sending body</td>
</tr>
</tbody>
</table>

| ACTOR | SYSTEM |
1. The use case begins when the MGC detects a message destined for it on the network
2. The call agent reads it
3. Determines its validity
4. Determines its encoding
5. Determines its authenticity
6. Decodes it
7. Processes it
8. Replies

The above use case indicates the following list of the events, the Megaco Text Encoding system must respond and operate on:

**SEQUENCE DIAGRAM 2A**

```
Remote Client  System MGC

Must

Read Message

Optional

Determine validity,

Determine encoding,

Establish Authenticity

Optional

Process Message

Must

Reply (Encode)
```
SEQUENCE DIAGRAM 2B

CONTRACTS

1.

Software Research Group
### Name: Read Message and determine validity

**Responsibilities:** Reads data from the network

**Type:** System

**References:** Interaction Diagram: Decoding Process  
Use Case: Process Message

**Exception:** Garbled message read or failure in reading message this will be handled by the Root gateway which will send error message to Gateway

**Notes:** NIL

**Post Conditions:**
- The message is stored for decoding

**Pre Conditions:**
- Message should be valid

### Name: Decode Message

**Responsibilities:** Decode Text and Binary message according to port on which message is received using either ASN or ABNF

**Type:** System

**References:** Interaction Diagram: Decoding  
Use Case: Process Message

**Exception:** The decoding is not successful due to a protocol error or incomplete message is reported

**Notes:** NIL

**Post Conditions:**
- The message is converted to MGC internal TermCommandData structures

**Pre Conditions:**
- The message must be according to grammar rules

### Name: Determine message authenticity

**Responsibilities:** Check the authenticity of the Gateway by checking the IP and alias

**Type:** System

**References:** Interaction Diagram: Decoding  
Use Case: Process Message

**Exception:** The gateway is not authenticated in which case an error message is made and sent

**Notes:** NIL

**Post Conditions:**
- The message is sent for processing

**Pre Conditions:**
- Message must be from an authenticated source

---

*Software Research Group*
<table>
<thead>
<tr>
<th>Name</th>
<th>Process Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>To process the message according to Gateway/Termination state machine and construct a TermCommandData structure for Encoding</td>
</tr>
<tr>
<td>Type</td>
<td>System</td>
</tr>
<tr>
<td>References</td>
<td>Interaction Diagram : Process Message</td>
</tr>
<tr>
<td></td>
<td>Use Case : Process Message</td>
</tr>
<tr>
<td>Exception</td>
<td>The message might not be valid for the current state of the Gateway/Termination in which case an error message is constructed.</td>
</tr>
<tr>
<td>Notes</td>
<td>NIL</td>
</tr>
<tr>
<td>Post_Conditions</td>
<td>• A TermCommandData is made for Encoding</td>
</tr>
<tr>
<td>Pre_Conditions</td>
<td>• The message must be in conformance with the current state of the termination</td>
</tr>
</tbody>
</table>

5.

<table>
<thead>
<tr>
<th>Name</th>
<th>Encode Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>To encode message in either text or binary format depending on which is supported by the gateway</td>
</tr>
<tr>
<td>Type</td>
<td>System</td>
</tr>
<tr>
<td>References</td>
<td>Interaction Diagram : Encoding</td>
</tr>
<tr>
<td></td>
<td>Use Case : Process Message</td>
</tr>
<tr>
<td>Exception</td>
<td>The TermCommandData has not been properly constructed for proper encoding in which case an error is returned</td>
</tr>
<tr>
<td>Notes</td>
<td>NIL</td>
</tr>
<tr>
<td>Post_Conditions</td>
<td>• The encoded message is sent to the service encoder for writing to the network</td>
</tr>
<tr>
<td>Pre_Conditions</td>
<td>• The TermCommandData must be valid for encoding process</td>
</tr>
</tbody>
</table>
INTERACTION (COLLABORATION DIAGRAMS)

Manager
- GetGatewayHandlerID()
- SetGatewayHandlerID()
- SetGatewayHandlerQueue()
- SetGatewayHandlerQueueofTermination()
- AddGatewayHandlerStats()
- SetGatewayHandlerIP()
- SetTerminationID()

Manager acts as a data store. It keeps data in the form of maps. Each map contains a key-value pair.
The components in Manager are:
- Gateway Map — Key: Gateway alias, Value: Gateway ID
- Gateway Queue Map — Key: Gateway ID, Value: GWHandQuePtr
- TVG Map — Key: Term ID, Value: GWHandQuePtr
- Term Id Map — Key: Term ID, Value: GWHandId
- Term Alias Map — Key: Term Alias, Value: Term ID
- Phone No Map — Key: Phone Number, Value: Term Alias

TermIdVSp_Term
TermIdVSGWHId
GWAliasVSGWhId
TermIdVSp_GWhQueue
PhonevoVSTermId
TermIdVSp_Term

CallHandler
- ConfigureCallHandler()
- StartCallHandler()

CallHandler
- ConfigureCallHandler()
- StartCallHandler()

Encoder
- StartEncoder()
- ConfigureEncoder()

GatewayHandler
- SetGatewayHandlerID()
- GatewayHandler()
- StartGatewayHandler()

TimerController
- TermIDVspTerm
- GatewayControllerProcess()
- ConfigureGatewayController()
- AddTermination()
- GatewayController()

ReceivedResponseQueue
- ResponseHistoryQueue
- GatewayHandlerQueue()
- EncoderTaskQueue()
- SetGWWhDSProcessor()
- SetGWWhDSPtrGWWh

StartService()
0 Message Received

1 Select()
2 ReadData()

3.0 TextWrapper()
3.1 SetBuffer()
3.2 ParseMessage()
3.3 GetTransactionType()
3.4 IsErrorMessage()
3.5 IsReplyError()
3.6 GetVecTermData()

4 GatewayControllerMessage

5 PushIn()

6 Set_Event()

The Header Decoder deals with messages, i.e. both Commands and Responses

The Header Decoder maps the Gateway alias in the Gateway Map to retrieve Gateway ID. This Gateway ID is then mapped in the Gateway Queue Map to retrieve pointer to the Gateway Handler Queue for this gateway.
CallObject receives messages from the Service Call Handler Task Queue, and activates State Machine accordingly.
- CallId, CallState, Manager Info
- CallState (Defines the state of the Call Object.)

If new task in Call Handler Task Queue:
1. Pop task from the Queue
2. Get Call Id from the task
3. If Call Id present in Call Map:
   - Pass task to that Call Object
4. Else:
   - If task type is Setup:
     1. Make a CallObject
     2. Generate Call Id from Call Id generator
     3. Assign Call Id to Call Object
     4. Update Call Map
     5. Pass task to that Call Object
   - If task type is Terminate:
     1. Pass task to that CallObject
     2. After processing, remove call object
5. Else:
   - Return Error

A termination object is the representation of a termination of the gateway. It handles all the activities related to the termination. It contains the following data:
1. Termination Identifier
2. Gateway Controller Identifier
3. Termination state

Follow enumerated data types are used by termination object:
- SendAuditEndPoint
- SendSetupTask
- SendSetupCompleteTask
- SendRingCompleteTask
- SendRingbackCompleteTask
- SendCompleteCallTask
- SendOpenCompleteTask
- SendTerminateTask
- SendAuditConnection
- SendCreateConnection
- SendModifyConnection
- SendDeleteConnection
- SendAuditRetransmissionTimerExpired
- SendReconnect
- SendCompleteCallTask
- SendConnect
- SendConnectComplete
- SendOpenComplete
- SendTerminateTask
- SendAuditConnection
- SendCreateConnection
- SendModifyConnection
- SendDeleteConnection
- SendAuditRetransmissionTimerExpired
- SendReconnect
- SendCompleteCallTask
- SendConnect
- SendConnectComplete
- SendOpenComplete
- SendTerminateTask

The Decoder deals with Commands, Responses and Tasks. It retrieves messages from the Gateway Handler Queue, and passes Commands to RCVD Cmd Queue, Responses to RCVD Rsp Queue and Tasks to Call Handler Task Queue.

- If cmd in RCVDCOMDQUEUES
  1. If term exists for that terminal:
     - Pass cmd to the Term Object.
   2. Else:
     - Error
- If resp in RCVDRSPQUEUE
  1. Pass resp to the related Term Object.
  2. Else:
     - Error
- If task in GWCONTROLTASKQUEUE
  1. If task in Call Handler Task Queue:
     1. Get Call Id from the task
     2. Pass task to that Call Object
     3. After processing, remove call object
     4. Return Error

- Call ID Generator generates a Call ID
- Call ID or Call Identifier is a hexadecimal string, with a maximum of 32 characters.
if new message to be sent
If message is a command
Push command in SENT COMD QUEUE.
Push task in ENCODER TASK QUEUE.
Else
Push response in SENT COMD QUEUE.
Push task in ENCODER TASK QUEUE.

Encoder Message Queue contains messages sent by Encoders residing on Gateway Handlers. This is a central Queue shared by all the Gateway Handlers. Service Encoder retrieves messages from this Queue and takes action accordingly.

Encoder Task Queue contains a struct containing Trans-IDs to be piggy-backed and Queue Info GW controller when wishes to send several messages in the same UDP Packet, notifies by passing an indication and Queue Info to the Encoder Task Queue. Encoder retrieves Mesgs corresponding to these Trans-IDs and Queue info, piggybacks them into a single UDP packet.

Service encoder receives messages from Encoder Msg Queue. These messages contain Actual Payload, Destination Address, Port to be used, originating gateway. Service Encoder transmits this message through the UDP API.

Encoder Task Queue contains a struct containing Trans-IDs to be piggy-backed and Queue Info GW controller when wishes to send several messages in the same UDP Packet, notifies by passing an indication and Queue Info to the Encoder Task Queue. Encoder retrieves Mesgs corresponding to these Trans-IDs and Queue info, piggybacks them into a single UDP packet.

Service Encoder receives messages from Encoder Msg Queue. These messages contain Actual Payload, Destination Address, Port to be used, originating gateway. Service Encoder transmits this message through the UDP API.

Encoder Task Queue contains a struct containing Trans-IDs to be piggy-backed and Queue Info GW controller when wishes to send several messages in the same UDP Packet, notifies by passing an indication and Queue Info to the Encoder Task Queue. Encoder retrieves Mesgs corresponding to these Trans-IDs and Queue info, piggybacks them into a single UDP packet.

Encoder Task Queue contains a struct containing Trans-IDs to be piggy-backed and Queue Info GW controller when wishes to send several messages in the same UDP Packet, notifies by passing an indication and Queue Info to the Encoder Task Queue. Encoder retrieves Mesgs corresponding to these Trans-IDs and Queue info, piggybacks them into a single UDP packet.

Encoder Task Queue contains a struct containing Trans-IDs to be piggy-backed and Queue Info GW controller when wishes to send several messages in the same UDP Packet, notifies by passing an indication and Queue Info to the Encoder Task Queue. Encoder retrieves Mesgs corresponding to these Trans-IDs and Queue info, piggybacks them into a single UDP packet.
Procedures and Termination State Machine

STATE UNREGISTERED

Steps

This is the start state.

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

3. the state machine checks whether it wants to give its own new address so it replies with the new address

4. Simply sends an acknowledgment message back to the Gateway.

- Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

Makes new Audit Capabilities Media message for termination and sends to Gateway

On receiving AuditReply + Capabilities from the gateway the termination state machine

1. Sends a MODIFY back to the gateway.

- DIALPLAN
- Notify OFF-HOOK
- LocalControlDesc Stream mode SENDRECV
• EMBED Event ON-HOOK
• EMBED EVENT DIGITSCOLLECTED
• EMBED SIGNAL DIALTONE

2. Switches to ONHOOK state.

IF GET CAPABILITY TASK
   Send Capabilities = NULL

STATE ONHOOK
Steps

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

3. the state machine checks whether it wants to give its own new address so it replies with the new address

4. Simply sends an acknowledgment message back to the Gateway.

   • Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

1. Makes new Audit Capabilities Media message for termination and sends to Gateway

2. switch state to Unregistered

Else IF NOTIFY OFFHOOK

1. Send Notify Reply
2. Switch State to Dialing

Else If Add Reply and SDP PORT IS NOT PORT CHOOSE

- Send Subtract Real + Subtract EPH to Caller

Else IF SUBTRACT REPLY

1. Sends a MODIFY back to the gateway.

- DIALPLAN
- Notify OFF-HOOK
- SIGNAL EMPTY
- LocalControlDesc Stream mode SENDRECV
- EMBED Event ON-HOOK
- EMBED EVENT DIGITSCOLLECTED
- EMBED SIGNAL DIALTONE

Else IF RING_TASK

1. Send ADD Command (REAL)
2. Event OFFHOOK (REAL)
3. SIGNAL RING SIGNAL (REAL)
4. Send ADD Command (EPH)
5. ALIAS(EPH) =$
6. MEDIA DESC(EPH)
7. LOCALCTRLDESC(EPH)
8. STREAM PRESENT(EPH)
9. FILL REMOTE DESC (EPH)
10. STREAM MODE = SENDRECV (EPH)
11. Switch State Ringing

Else IF DELETE TASK

Software Research Group
1. SUBTRACT REAL

2. SUBTRACT EPH

Else IF GET CAPABILITY TASK

1. SEND CAPABILITY TASK + SDP

**STATE DIALING**

*Steps*

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

3. the state machine checks whether it wants to give its own new address so it replies with the new address

4. Simply sends an acknowledgment message back to the Gateway.

- Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

1. Makes new Audit Capabilities Media message for termination and sends to Gateway

2. Switch state to Unregistered

IF NOTIFY ONHOOK

1. SEND NOTIFY RESPONSE
2. SEND MODIFY

- DIALPLAN
- Notify OFF-HOOK
- SIGNAL EMPTY
- EMBED Event ON-HOOK
- EMBED EVENT DIGITSCOLLECTED
- EMBED SIGNAL DIALTONE
- Switch State ONHOOK

ELSE IF DIGIT MAP COMPLETION EVENT

- SEND NOTIFY RESPONSE
- IF SUCCESSFULLY LOCATED CALLED TERMINATION
  - SEND SETUP TASK
  - Switch State DIGITS COLLECTED
- ELSE IF NOT SUCCESSFUL
  - MODIFY BUSY TONE (SIGNAL) ONHOOK (EVENT)
  - Switch State WAITING FOR ONHOOK

ELSE IF GET CAPABILITIES TASK

- SEND CAPABILITIES NULL

STATE DIGITS COLLECTED

Steps

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
-Service change Version

-Service change Profile

5. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

6. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

7. the state machine checks whether it wants to give its own new address so it replies with the new address

8. Simply sends an acknowledgment message back to the Gateway.
   - Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

3. Makes new Audit Capabilities Media message for termination and sends to Gateway

4. Switch state to Unregistered

ELSE IF NOTIFY ONHOOK

   Reply NOTIFY
   SEND MODIFY
   - DIALPLAN
   - Notify OFF-HOOK
   - SIGNAL EMPTY
   - EMBED Event ON-HOOK
   - EMBED EVENT DIGITSCOLLECTED
   - EMBED SIGNAL DIALTONE
   - Switch State ONHOOK
   - SEND FREE CALL TASK

Else IF SETUP RETURN TASK

1. ADD REAL TERM
2. ADD EPH TERM
3. SET EPH MEDIA
4. SET EPH STREAM =RECVONLY
5. Switch State WAITING FOR CONTEXT CREATION

Else IF SETUP ERROR TASK
1. MODIFY (REAL)
2. SIGNAL BUSYTONE (REAL)
3. EVENT (ONHOOK)
4. Switch State WAITING FOR ONHOOK

Else IF GET CAPABILITY TASK
SEND CAPABILITY NULL

STATE WAITING FOR CONTEXT CREATION

Steps

IF Received serviceChange from the gateway

Input parameters
-Service change delay
-Service change Address
-Service change Version
-Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it
2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply
3. the state machine checks whether it wants to give its own new address so it replies with the new address
4. Simply sends an acknowledgment message back to the Gateway.
   - Programs terminations in the null context
IF Service change for Termination. Additional Steps taken

5. Makes a new Audit Capabilities Media message for termination and sends to Gateway

6. Switch state to Unregistered

Else If Notify On Hook

1. Send Notify Reply
2. Send MODIFY DETECT OFF HOOK AND DIAL DIGITS
3. Switch State On Hook

Else If Reply Add

1. If SDP present, send Setup Complete Task and Switch State Media Opened
2. Else If not present, send Subtract and Terminate Task

**STATE MEDIA OPENED**

*Steps*

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

5. The termination state machine examines Version and if not supported, responds with Version not supported or otherwise negotiates it

6. The state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

7. The state machine checks whether it wants to give its own new address so it replies with the new address

8. Simply sends an acknowledgment message back to the Gateway.

   - Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

7. Makes a new Audit Capabilities Media message for termination and sends to Gateway
8. Switch state to Unregistered

Else If Notify OnHook

1. Reply Notify, Send Terminate Task and Send Modify Connection Detect Off hook and Digits

2. Switch State OnHook

Else If Ring Back Task

1. Send Modify + Ring Back Signal + Remote Descriptor
2. Switch State Ring Back

Else If Setup Error Task

1. Send Modify + Busy Tone Signal + OnHook Event
2. Switch State WAITING FOR ON HOOK

Else IF Delete Task

1. Send Modify + Busy Tone Signal + On Hook Event
2. Send Subtract Real + Subtract Ephemeral
3. Switch State WAITING FOR ON HOOK

Else If Get Capabilities Task

1. Send Capabilities Null

STATE RING BACK

Steps

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

9. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

10. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

11. the state machine checks whether it wants to give its own new address so it replies with the new address

12. Simply sends an acknowledgment message back to the Gateway.
• Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

9. Makes new Audit Capabilities Media message for termination and sends to Gateway

10. Switch state to Unregistered

Else IF NOTIFY ONHOOK

1. REPLY NOTIFY

2. SEND TERMINATE TASK

Else IF MODIFY REPLY

SEND RING BACK COMPLETE TASK

Else IF OPEN TASK

1. MODIFY(EHP)

2. SET LOCALCTRLDESC(EHP)

3. SET STREAM MODE SENDRECY(EHP)

4. MODIFY(REAL)

5. SET SIGNAL (EMPTY)

6. SEIZURE EVENT

7. ON HOOK EVENT

8. CLEAR FORWARD EVENT

9. FAX TONE EVENT

10. Switch State ACTIVE

11. AUDIT (REAL) : OBSERVED_EVENTS | EVENTS | SIGNALS | DIGITMAP

12. AUDIT (EHP) : MEDIA

Else IF DELETE TASK

1. SUBTRACT (REAL)

2. SUBTRACT (EHP)
3. Switch State ONHOOK

Else IF GET CAPABILITIES TASK

SEND CAPABILITIES NULL

STATE ACTIVE
Steps

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

a. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

b. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

c. the state machine checks whether it wants to give its own new address so it replies with the new address

d. Simply sends an acknowledgment message back to the Gateway.

- Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

1. Makes new Audit Capabilities Media message for termination and sends to Gateway

2. Switch state to Unregistered

Else IF NOTIFY ONHOOK

1. REPLY NOTIFY

2. SEND TERMINATE TASK
3. Switch State ACTIVE AND ONHOOK

Else If Notify Seizure
    1. Send Seizure Detected task

Else If Notify Clear Forward Event
    1. Send Clear Forward Detected Task

Else If Notify Fax Tone Event
    1. Send Setup Fax Task
    2. Switch State Fax Tone

Else IF MODIFY REPLY
    1. SEND COMPLETE CALL TASK

Else IF DELETE TASK
    1. SUBTRACT (REAL)
    2. SUBTRACT (EPH)
    3. AUDIT STATS (BOTH)
    4. Switch State WAITING FOR ONHOOK

Else IF GET CAPABILITY TASK
    1. SEND CAPABILITY TASK NULL

Else If SEND SEIZURE TASK
    1. MODIFY + SEIZURE SIGNAL

Else If SEND CLEAR FORWARD
    1. MODIFY + WARNING TONE

Else If CHANGE CODEC TASK
    IF SDP IS NULL
    1. SEND TERMINATE TASK
    Else

Software Research Group
SEND MODIFY + STREAM SENDRECV + LOCAL DESC

Switch State FAX CHANGE OVER

Else if Notify Flash Hook
  • Notify Response
  • Switch State Adialing

Else If Connect Task
  • Send Add + SDP

STATE FAX IN PROGRESS

Steps
If Modify Reply
  Switch State Active
Else If Complete Fax Task
  SEND MODIFY REAL + EPH + MEDIA + REMOTE DESC

STATE FAX TONE

Steps
IF SETUP RETURN TASK
  MODIFY REAL + EPH + MEDIA + LOCALCTRL + RECVONLY

Else If Modify Reply AND FAX CODEC SUPPORED
  1. Send SETUP COMPLETE TASK
  2. Switch State FAX IN PROGRESS
 Else if FAX CODEC NOT SUPPORTED
  1. Send Subtract + Terminate Task
  2. Switch State WAITING FOR ON HOOK

STATE FAX CHANGE OVER

Steps
If Received MODIFY REPLY
  1. Send Ring Complete Task
  2. Switch State ACTIVE

STATE ACTIVE AND ONHOOK

Steps
IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

3. the state machine checks whether it wants to give its own new address so it replies with the new address

4. Simply sends an acknowledgment message back to the Gateway.

• Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

2. Makes new Audit Capabilities Media message for termination and sends to Gateway

3. Switch state to Unregistered

Else IF SUBTRACT REPLY

1. Sends a MODIFY (REAL) back to the gateway.

• DIALPLAN

• Notify OFF-HOOK

• SIGNAL EMPTY

• LocalControlDesc Stream mode SENDRECV

• EMBED Event ON-HOOK

• EMBED EVENT DIGITSCOLLECTED
- EMBED SIGNAL DIALTONE

2. Switch State ONHOOK

Else IF DELETE TASK

1. SUBTRACT (REAL)

2. SUBTRACT (EPH)

3. Switch State ONHOOK

**STATE WAITING FOR ONHOOK**

*Steps*

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply

3. the state machine checks whether it wants to give its own new address so it replies with the new address

4. Simply sends an acknowledgment message back to the Gateway.

- Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

a. Makes new Audit Capabilities Media message for termination and sends to Gateway

b. Switch state to Unregistered

Else IF NOTIFY ONHOOK
1. SEND NOTIFY RESPONSE

2. Sends a MODIFY (REAL) back to the gateway.
   - DIALPLAN
   - Notify OFF-HOOK
   - SIGNAL EMPTY
   - LocalControlDesc Stream mode SENDRECV
   - EMBED Event ON-HOOK
   - EMBED EVENT DIGITSCOLLECTED
   - EMBED SIGNAL DIALTONE

3. Switch State ONHOOK

   Else IF GET CAPABILITY TASK
       Send CAPABILITY TASK NULL

   Else If Timer Expired
       Send Modify Busy Tone Signal

**STATE DIALING**

*Steps*

IF Received serviceChange from the gateway

Input parameters

- Service change delay
- Service change Address
- Service change Version
- Service change Profile

1. the termination state machine examines Version and if not supported responds with Version not supported or otherwise negotiates it

2. the state machine checks whether MGC wishes to handoff to another MGC if yes than it sends new MGC address as reply
the state machine checks whether it wants to give its own new address so it replies with the new address

Simply sends an acknowledgment message back to the Gateway.

- Programs terminations in the null context

IF Service change for Termination. Additional Steps taken

1. Makes new Audit Capabilities Media message for termination and sends to Gateway
2. Switch state to Unregistered

Else If Notify On Hook

1. Modify + Off Hook + Dial Plan + empty Signals
2. Switch state On Hook

Else If Notify Digit Map Completion Event

1. Notify Reply + Send Setup Task + Switch State Digits Collected

Else IF GET CAPABILITIES Task

SEND CAPABILITIES TASK NULL

STATE ADIALING

If Notify Digits

- Send Reply
- Send TempSetup Task
- Switch State AdigitsCollected

STATE ADIGITSCOLLECTED

If Ringback Task

- Send Modify
- Ringback Signal
Soft Switch Design Document

- SDP selected by new Termination in Remote
- Switch state ARingback

**STATE ARINGBACK**

If Modify Reply
- Send Ringback Complete task

If Open Task
- Send Modify Send/Recv
- Switch State Aactive

**STATE AACTIVE**

If Notify Flash
- Switch State (Flash Hook)
- Send Notify Reply
- Send Invite Task
Call Handler State Chart

- **Call in Start State**
  - Capability Task
  - **Call in Wait for Setup Complete State**
    - Setup Complete
  - **Call in Wait for Ring Complete State**
    - Ring Complete
  - **Call in Wait for Called Term OffHook State**
    - Complete Call
  - **Call in Active State**
    - Terminate/Delete

---

*Software Research Group*
Procedures and Call Handler State Machine

Start

In the Start state, on receiving stimuli Setup from Gateway Controller Call Handler sends Get Capability task to the Called termination and on receiving Capability task Call Handler responses by sending Setup Return. Call Handler moves to Wait for Setup Complete state.

Retrieve local Termination Struct
Retrieve remote Termination Struct
If Remote Termination is in OnHookState
    Retrieve local and remote SDP
    Compare and decide on a common set of SDPs
    Make SetupReturn task
    Add set of SDPs
    Set local Termination id in task
    Retrieve GwController id of local termination
    Push task in GwController task queue of that gateway
Change state to WaitForSetupComplete

Else if
Make SetupError task
Set fields
Retrieve local termination Object
Retrieve GwController id from it
Push task in GwController task queue of that gateway

**Wait for Setup Complete**

In the *Wait for Setup Complete*, one stimulus can transit the Call Handler. On receiving Setup Complete from Gateway Controller

- Call Handler returns Ring to the Termination
- Call Handler enter *Wait for Make Complete* state.

Retrieve SDP1 from the message
Retrieve connection Id from message
Make Ring task
Add SDP 1 to task
Retrieve GwController id of remote termination
Push task in GwController task queue of that gateway
Change state to WaitForRingComplete

**Wait For Ring Complete**

In the *Wait for Ring Complete* state, one stimulus can turn on the State machine. When Ring Complete is received from the Gateway Controller,

- Call Handler moves to *Wait for Called Term Off Hook* state
- Call Handler sends Ring back to the Gateway Controller

If RingComplete task

Retrieve SDP2 from the message
Make RingBack task
Add SDP2 to task
Retrieve GwController id of local termination
Push task in GwController task queue of that gateway
Change state to WaitForCalledTerminationOffHook
**Wait for Called Term Off Hook**

In the *Wait for Called Term Off Hook* state, one stimulus can activate the Call Handler. The event Complete Call from Call Handler

- Causes a switch to the *Active* state
- Call Handler sends Open in response.

If Complete Call task
- Make Open Task
- Retrieve GwController id of local termination
- Push task in GwController task queue of that gateway
- Change State to Active

**Active**

In the *Active* state, on receiving stimuli Terminate the Call Handler responds by

- Sending a Delete message to the Gateway Controller (provided there are no more Active Calls it ceases to exist.)

If Terminate task
- Make delete task
- Retrieve GwController id of local and remote terminations
- Push task in GwController task queue of those gateways

**Application Programming Interface**

The list of functions for each component does not include those functions that are used internally (i.e. within the class itself).

**Call Agent API**

The following list of functions is available to the application of Call Agent

*The following three functions though not of the Call Agent are called solely be the call agent object*

ConfigureCallHandler

To configure the callhandler and set its callhandler task queue pointer and Manager pointer.

Report current Calls

Software Research Group
To show the total number of calls underway and their statistics that are being handled by the call handler currently.

ConfigureHeaderDecoder

To configure the Header Decoder and set its Encoder Message Queue pointer and manager pointer.

AddGatewayHandler

To add a new gateway handler.

RemoveGatewayHandler

To remove a gateway handler.

StartGatewayHandler

To start a gateway handler.

StopGatewayHandler

To stop a gateway handler.

AddTermination

To add a new termination in a gateway.

RemoveTermination

To remove a termination in a gateway.

ConfigureTermination

To set the configuration of a gateway.

StartCallAgent

To start the Call Agent.

StopCallAgent

To remove the Call Agent.
Queue API

Following list of functions is available in the implementation of queue

Push

To push an item in queue.

Pop

To pop an item in queue.

Size

To get the number of items present in the queue.

Find

To find an item in the queue.

Erase

To delete an item from the queue.

GatewayController API

startGatewayController

To Start the GatewayController object

stopGatewayController

To halt the GatewayController object

addTermination

To add a new termination (to a specified gatewayhandler).

RemoveTermination

To remove a termination from GatewayHandler

ConfigureGatewayController

Used to configure the gateway controller by giving it pointers to the queues and the manager
**Termination API**

*SetTerminationAlias*

To set a particular Alias (e.g. “0GW1”) for a termination.

*ProcessStimulusRequest*

This function is used to activate the state machine of the termination object in response to a command.

*ProcessStimulusResponse*

This function is used to activate the state machine of the termination object in response to a reply.

*ProcessStimulusTask*

This function is used to activate the state machine of the termination object in response to a task.

*processStimulusRetransmissionTimerExpired*

Called by the timer process for envoking requests and responses retransmission timers

*ProcessStimulusTimerExpired*

To process timers started be the termination itself.

*FreeTermination*

Called in order to turn the status of a termination to unregistered.

**GatewayHandler API**

*AddTermination*

Adds a termination in this object.

*RemoveTermination*

Removes a specified termination from this object

*StartGatewayHandler*

Starts the threads for this gatewayhandler
StopGatewayHandler

Stops the threads related to this gatewayHandler

ConfigureGatewayHandler

Used to configure the object with pointers to queues and manager

Manager API

AddTermination

To add a new termination to the termination Alias vs. termination Id map.

RemoveTermination

To remove a specified termination from the termination Alias vs. termination Id map.

getTerminationOfPhoneNo

To search for a termination object based on its phone no.

setTerminationOfPhoneNo

To set the phone number of a specified termination.

unsetTerminationOfPhoneNo

To remove the phone number of a particular termination based on either its Id or its alias.

GetGatewayHandlerId

To get the Id of a gatewayHandler based on its alias.

unsetGatewayHandlerId

To remove the Id of a specified gatewayHandler.

SetGatewayHandlerQueueOfTermination

To set the queue for messages to a specified termination.

UnSetGatewayHandlerQueueOfTermination

To remove the queue for messaging a specified termination.
Soft Switch Design Document

GetGatewayHandlerQueueOfTermination

To search for the messaging queue for a specified termination.

GetGwhIdVSTermId

Get the gateway Id for a specified termination.

SetGwhIdVSTermId

Set the gateway Id for a specified termination

UnsetGwhIdVSTermId

Unset the gateway Id for a specified termination.

stopGatewayHandler

To stop a gatewayHandler

startGatewayHandler

To start a gatewayHandler

Pending Issues

1. TCP support
2. Multiple Commands/Actions/Transactions in one PDU
3. Three or more commands in an action
4. Three way calling and call waiting
5. FAX/MUX/MODEM support
6. Service Change FORCED from MGC to Gateway
7. MGC to MGC takeover in the case of advent of MGC failure
8. Dynamic computation of retransmission timers based on network traffic conditions.
9. Lockstep mode not supported yet.
10. MIB not implemented.