Internet Engineering Task Force P. Sijben
Internet Draft Mike Buckley
John Wachter
John Seegers
Chia Li
Lucent Technologies
Rex Coldren
AG Communication Systems

Date 9 February 1999
Expires 9 August 1999

Toward the PSTN/Internet Inter-Networking
MEDIA DEVICE CONTROL PROTOCOL
Version 0.3
<draft-sijben-megaco-mdcp-01.txt>

Abstract

This document describes a protocol for use between media devices and their controllers. The scope of such media devices includes voice, video and data applications. This work started with the decomposition of gateways into functional elements, viz. media gateway controller, media gateway and signaling gateway. A protocol is needed to allow media gateway controllers and media gateways to communicate. Since then the scope has been broadened to the control of media devices that do not have a gateway function, such as IVR systems, multipoint processors, etc.
A model is introduced to describe the functionality of media devices and the various operations on that model are described.

Table of contents

1 Introduction

Gateways provide interface and interworking for devices on the packet network to and from devices on other networks. As different applications for gateways arose, so too did the need to scale and diversify the physical implementation of the gateway. To facilitate this, the gateway decomposition effort was introduced into standards. Several different reference models were introduced to represent the functional elements and interfaces of a decomposed gateway. This document will illustrate some of these models to give a frame of reference, and will proceed to propose a protocol for use between two of the functional elements in the model.

It should be noted that the original application for gateway decomposition was the scaling of trunking gateways. These are gateways used between Switched Circuit Network (SCN) trunks and packet networks. After that it was found that similar interfaces applied when decomposing other types of gateways and media devices. Consequently, the scope of gateway decomposition has grown and the requirements for protocols used in such decompositions have needed to be changed accordingly.

We introduce the name "Media Device Control Protocol", MDCP, to be illustrative of this change in scope. Similarly, we use the terms "media device" and "media device controller" instead of "media gateway" and "media gateway controller" to indicate the broader scope.

2 Scope

This document describes a protocol for use between media devices and their controllers. The transport protocol used for the protocol will initially be UDP/IP but other transports may be foreseen. The scope of such media devices includes voice, video and data applications.

2.1 Applications

The scope of media devices in this Recommendation includes the following applications:

- **Trunking Gateways** that interface between SCN networks and packet networks. The SCN interface includes trunking equipment commonly found in the SCN between central office switching centers.
- **Access Gateways** that interface UNI interfaces like ISDN (PRI and BRI) and traditional analogue voice terminal interfaces to packet networks.
- **Residential Gateways,** which are access gateways for a small number of voice terminals that can be co-located with a cable modem or set top box.
- **Network Access Servers** that convert modem signals to and from SCN networks and provide data access to packet networks.
- **Multipoint Processors** that provide multipoint connectivity between terminals on SCN networks and on packet networks.
- **Interactive Voice Response Systems** that provide automatic voice response and switching services in response to DTMF signals received from SCN or packet network terminals.
- **Transcoding Devices** that allow terminals within a packet network to exchange media streams even if they do not support the same codecs.

2.2 Reference Configurations

Figure 1 to 4 show reference models of the decomposed gateway. The first was introduced in ETSI TIPHON and the second is being used by ITU-T Study Group 16 (SG16). We introduce two more diagrams to clarify points not illustrated by either of these models. To simplify later discussions in this document, we will use only the reference point nomenclature shown in the fourth diagram.

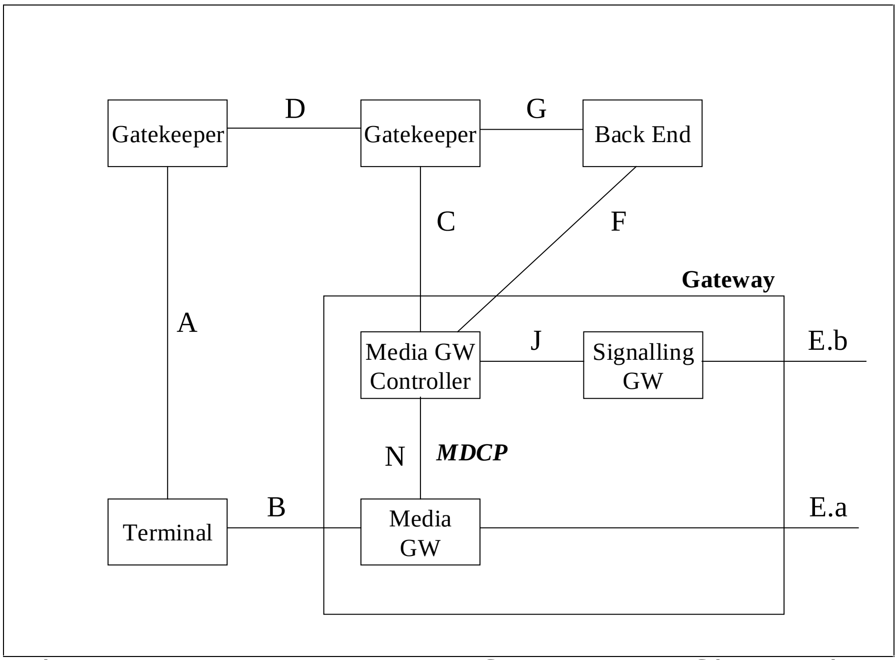

Figure 1: ETSI TIPHON Reference Configuration

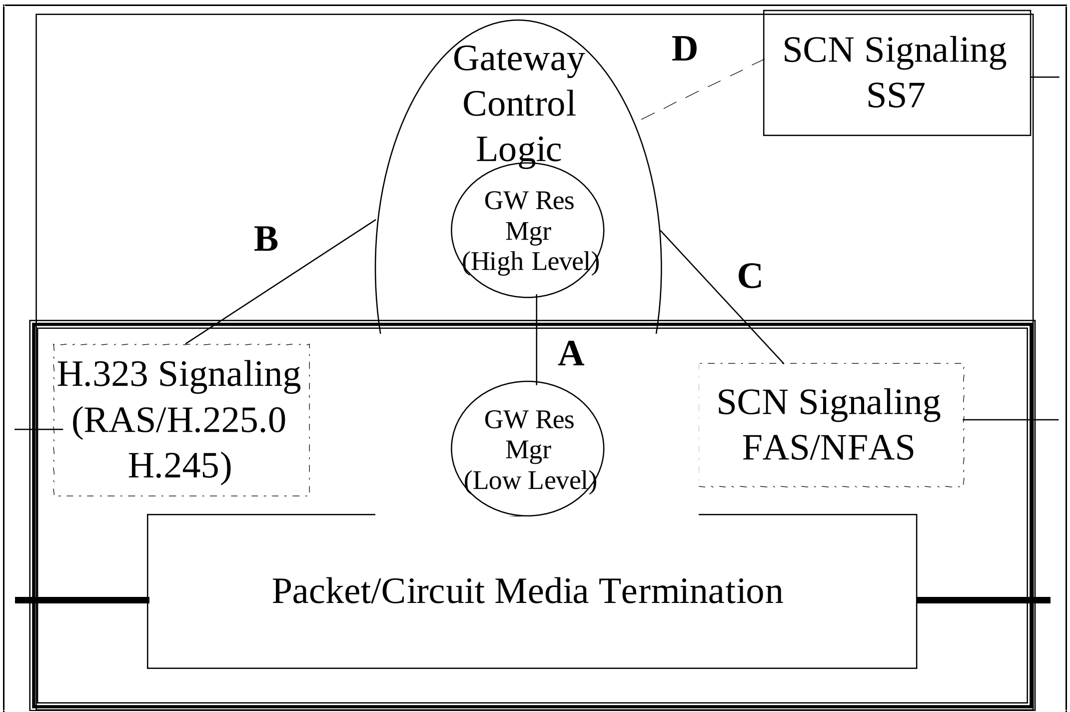

Figure 2: ITU-T SG16 Reference Configuration

The SG16 model distinguishes between Facility Associated Signaling (FAS) and Non-Facility Associated Signaling (NFAS). We feel that certain signaling types can and should be dealt with by the media control protocol. Certain H.245 logical channel negotiations may be more easily handled from the MD as well (i.e., autonomous negotiations). Consequently, we show the TIPHON "N" reference point factored to include these considerations in the following reference illustration. We append to the "N" reference point the relevant ITU reference point identifier (e.g.-“N.A”) to illustrate this factoring. MDCP provides a common signaling and control framework for interfaces N.A, N.B and N.C.

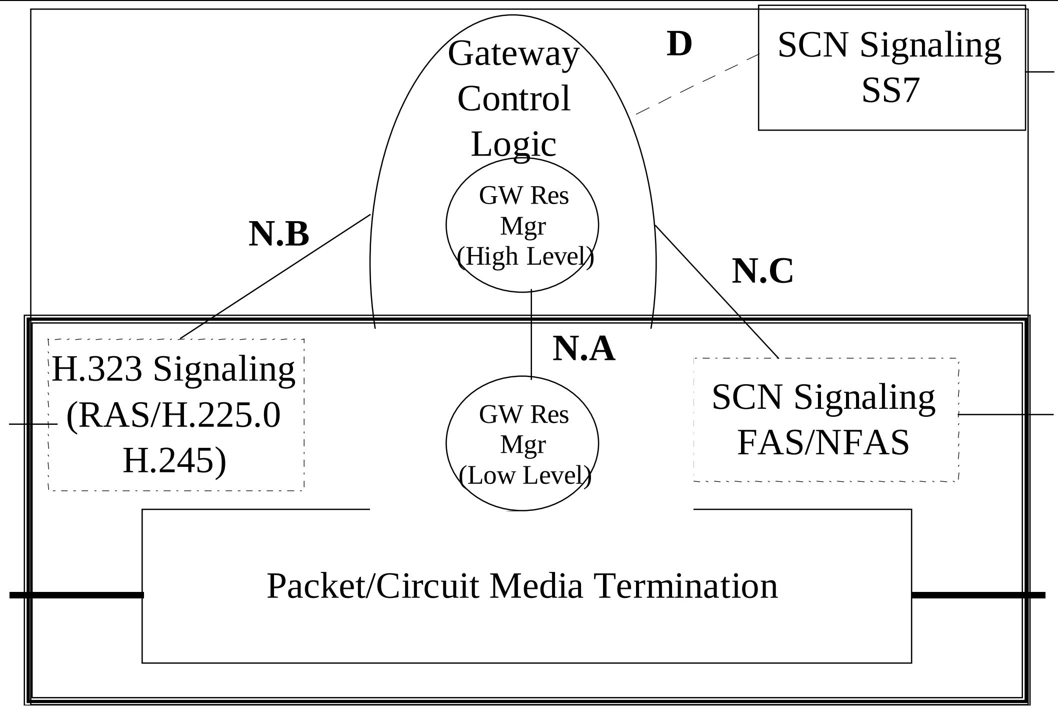

Figure 3: MDCP Illustrative Reference Configuration

To illustrate further, we show an example configuration that combines TIPHON, ITU SG16 and the MDCP illustrative reference configurations. The example begins to show some of the possible internal functions of the media gateway as well as some sample signaling types used at the reference points.

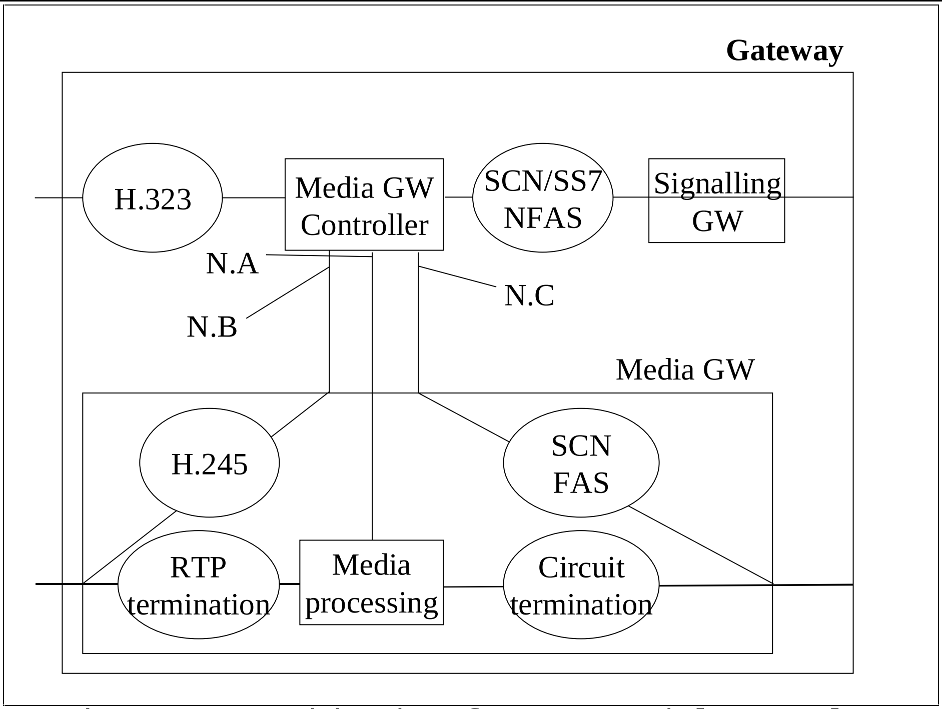

Figure 4: Combined reference model example.

3 References

TS 101 313 V0.4.1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Network architecture and reference configurations; Phase II: Scenario 1 + Scenario 2",
ETSI TIPHON, January, 1999

T. Taylor, P. Sijben, L. Spergel, G. Kimchi, R. Scholl, S. Petrack, E. Zimmer, H. Schink, W. Wimmreuter, O. Hersent, M. Holdredge, J. Vandenameele, B. van Doorselaer,
"Requirements for the Reference Point ("N") between Media Gateway Controller and Media Gateway", <draft-tiphon-arch-gway-decomp-00.txt>, November, 1998

P. Sijben, M. Buckley, T. Fritz, J. Wachter, J. Segers, C. Li, "Toward the PSTN/Internet Inter-Networking MEDIA DEVICE CONTROL PROTOCOL", <draft-sijben-Megaco-mdcp-00.txt>, November, 1998

L. Spergel, J. Wachter, "Media Device Control Protocol (MDCP) Call Flows", TIPHON 11TD043, December, 1998

4 Definitions, Symbols and Abbreviations

4.1 Definitions

Call: A call consists of a number of associated connections (audio and/or video and/or data) between edge points and media resources inside a media device.

Conference: A conference is a group of one or more associated calls.

Connection: Within the scope of this proposal, the term connection is used to identify paths over which media flows inside media devices. These paths are identified by the resources between which the media flows. Connections are unidirectional.

Control Function: A command, indication or response exchanged between a media device controller and a media device.
Control Session: A media device controller that wants to make use of a media device has to establish a control session with that media device. Once a control session is established between controller and media device, messages carrying control functions may be exchanged to allow the controller to perform connection control. There are two types of control sessions in media device control. (1) A media gateway controller may control conferences and calls in multiple media devices. Conversely, conferences in one media device may be controlled by multiple media gateway controllers. (2) The resource management control session. For every media device, there is one active MDC acting as a resource manager. This resource manager is notified, for instance, of failing hardware in the media device.
Edge point: An edge point is an abstract representation of a physical connection between a media device and the outside world. We distinguish between media edge points and signaling edge points. Media edge points carry media streams to and/or from media devices; signaling edge points carry signaling information.
Gatekeeper (GK): This H.323 entity performs many functions including authentication, authorization, alias resolution and call routing for H.323 entities.
Media Device (MD): A device in a packet network that processes media streams. Examples include media gateways, IVR systems, multipoint controller, etc.
Media Device Controller (MDC): Controls the parts of the call state that pertain to connection control for a number of separate voice or other multimedia channels. It contains all connection control logic. Examples of media device controllers are media gateway controllers and multipoint controllers.
Media Gateway (MG): A device that terminates SCN and packet network media streams and maps media from packet to switched networks. It may offer transcoding functions as well as media mapping functions.
Media Gateway Controller (MGC): Controls the parts of the call state that pertain to connection control for a number of separate voice or other multimedia channels. It maps SCN signaling and call control information into the packet network call state and control information. This device includes a Signaling Interface for communicating with other signaling elements (e.g. Gatekeepers). The Controller contains all connection control logic.
Multipoint Processor: an entity on the network that provides centralized processing of audio, video, and/or data streams in a multiparty conference. The multipoint processor provides mixing, switching or other media processing under the control of a media device controller.
Resource: A resource is an abstract representation of media device functionality. We distinguish between

edge resources and media resources. The difference between the two is that media resources originate or terminate media streams inside media devices, while edge resources carry information between media devices and the outside world.

Signaling Gateway (SG): A functional entity connected to the SCN that terminates SS7 or other signaling links. In general, the Signaling Gateway function maintains only enough information about the call state to manage the protocol interfaces. After processing the incoming SCN signaling data, the Signaling Gateway will deliver this information to the appropriate Media Gateway Controller. The gateway could pass the signaling information to several separate Media Gateway Controllers to make economic use of the powerful SS7 interface.

4.2 Abbreviations

ATM	Asychronous Transfer Mode
DTMF	Dual Tone Multi Frequency
GK	GateKeeper
IP	Internet Protocol
ISDN	Integrated Services Digital Network
MC	Multipoint Controller
MCU	Multipoint Control Unit
MD	Media Device
MDC	Media Device Controller
MF	Multi Frequency
MG	Media Gateway
MGC	Media Gateway Controller
MP	Multipoint Processor
RTP	Real-Time Transport Protocol
SCN	Switched Circuit Network
SG	Signaling Gateway
SS7	Signaling System Number 7
UDP	User Datagram Protocol

5 Functional Model of Media Devices

MDCP uses the following approach as shown in Figure 5. Media devices consist of a number of resources that must be connected and controlled to establish media connections. There are two kinds of resources in the MDCP internal architecture model:

1. Edge points: Internal representations of external connections from a media device to the outside world. Examples of edge point types are RTP Stream, UDP Port, SCN Bearer Channel (e.g. DS0), ATM SVC, and Signaling Backhaul. Edge points have associated with them properties that can be controlled by the MDC via the control protocol. Different edge point types have different sets of properties. Properties include, for instance, the allocation of an echo canceller or DTMF detector to the edge point, or the codec being used for the media stream that flows through the edge point.
2. Media Resources: These operate on media streams but do not send or receive media directly to or from the outside of the media device. An example of a Media Resource is a conference bridge. Media Resources also

have Properties. A property of a conference bridge, for instance, would be the number of ports on the bridge.

Not all Resource types are required to be supported in all media devices. This allows media devices to be tailored to specific applications. MDCP allows vendor specific extensions to existing resource types, and also supports the addition of new resource types. Figure 5 gives an example of the functional model for a basic call. Note that the two interfaces N.A and N.B terminate at the same edge point in the media device. The reason is that they both pertain to the same call. For backhaul of facility associated signaling there is a separate edge point.

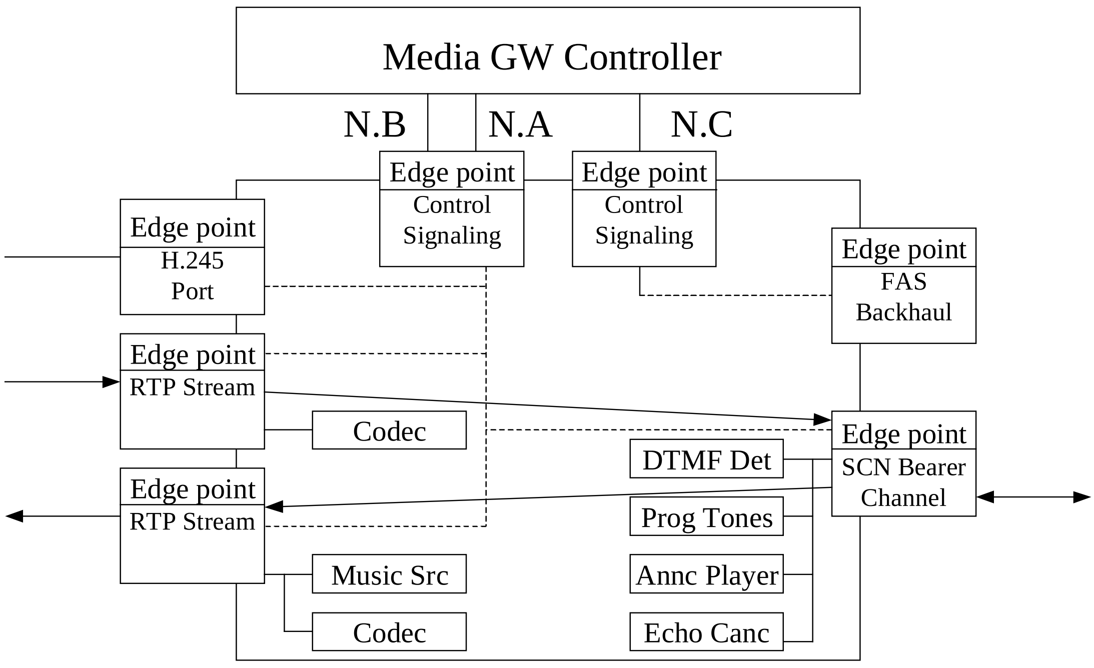

Figure 5: MDCP Functional Model Example.

6 Control Model

A media device is made up of a number of resources that must be connected and controlled to make up media connections. A connection is an association of edge points and media resources that operates unidirectionally on one media type. A call is defined as a collection of one or more edge points and/or media resources, and zero or more connections. A conference is defined as a collection of one or more calls. For example:

- A point-to-point voice call is be made up of two unidirectional voice connections.
- A multicast call is made up of a number of

unidirectional connections that connect to multicast stream edge points (e.g. having as properties multicast IP addresses or multicast ATM VCs).
- A multipoint conference is made up of a number of calls, all of which connect to a conference bridge.
- A point-to-point multimedia call is made up a series of voice, video and data Connections.

The conference and connection states are maintained in the media device controller. A call may encompass media streams of different types (audio, video and data). In the following subsections we describe the notions of control session, connection, call, conference as they are used within the scope of the control protocol in more detail. In particular, connections, calls and conferences are used here as concepts internal to a media device.

6.1 Control Sessions

There are two types of control sessions in media device control. First, a media device controller may control conferences and calls in multiple media devices. If a MDC uses resources in a media device, a control session will be established between the two. This type of control session is called a connection control session. For every call, the media device knows the identity of the MDC controlling that call. Potentially, all conferences in a media device are controlled by separate controllers. Therefore establishing a control session for connection control should be an operation with low cost.

The second type of control session is the resource management control session. For every media device, there is one active MDC acting as resource manager. This resource manager is notified, for instance, of failing hardware in the media device. The resource manager is also notified if the media device loses connections to connection managers (MDCs). There may be a backup resource manager that takes over the task of resource management if the primary resource manager fails. As there is only one active resource management control session at the same time, there is no need for low cost establishment of resource management control session.

The resource manager may send security tokens to the media device. These can be used to authenticate connection managers (MDCs) quickly.

6.1.1 Connection Control Sessions

Connection control sessions are established automatically when a MDC controls a connection in a media device. Authentication is done by means of security tokens. The media device’s resource manager may send tokens to the media device. If a media device receives a message requesting it to set up a conference, this request may be authenticated using these tokens. Details on the exact information contained in the tokens is for further study.

A media device controller may control conferences and calls in multiple media devices. If a MDC uses resources in a media device, a control session will be established between the two. For every call, the media device knows the identity of the MDC controlling that call.

6.1.2 Resource Management Control Sessions

Since each MD is subject to the resource management function residing in a particular MDC, the following discussion of the registration and control handover of that function involves references to MDCs. As the discussion focuses on the resource management function, we refer to these as "resource managers."

6.1.2.1 Registration

Registration must take place for a resource manager to establish a control session with a media device. There is one active resource manager per media device. In order to deal with failing resource managers, a second resource manager may be registered as backup. If the primary resource manager fails, the backup may take over.

Registration can be initiated by resource managers as well as by media devices. In the first case, the resource manager has learned the address of the media device and issues a registration command to it. The registration command may also be sent to a multicast group by a media device in search of a resource manager. After the registration phase, the relationship between media device and resource manager is strictly master/slave.

Depending on the issuer of the Registration command, several fields may be wild-carded.

By sending parameter lists back and forth using successive RegistrationResponse commands, a media device and a controller may use the Registration command to negotiate the parameters of the control session. In the registration process, the controller and media device involved in the negotiation do not repeat the entries which they agree upon. This negotiation ends when all the parameters are agreed upon or when a termination command is received.

6.1.2.2 Control Handover

There are three ways by which control can be handed over:

- **Timeout on heartbeat:** Should the MD discover that its resource manager does not respond within the acknowledgment timing interval on messages or heartbeat indications, the MD may conclude that the resource manager has died and may send a Registration command to any of its secondary controllers with the option Primary set.
- **Secondary discovers**: Alternatively, a secondary MDC may discover the death of the primary MDC and may issue a Registration command with the Primary flag set. Note that the new primary must have been registered as secondary first.
- **Voluntary handover**: The third option is that the MDC hands over control itself. It names its successor by sending a Registration message with the address of

its successor filled in as controller. This successor must be registered as secondary.

6.2 Conferences

A conference comprises one or more related calls. The simplest case is that of a single call. A multiparty conference would consist of a number of calls and a conference bridge resource. The MDC that controls a conference keeps track of all calls for the conference that connect to a given conference bridge so that it can control them independently or collectively (e.g. release individual calls from the conference versus releasing the entire conference).
Each conference is identified by a *ConferenceID*. *ConferenceIDs* are unique within Control Sessions.

6.3 Calls

A Call identifies the resources and connections between them that pertain to one point-to-point connection. A call may encompass media streams of different types (audio, video, data). A call can involve one or more connections.
Each call is identified by a *CallID*. *CallIDs* are unique within Control Sessions.

6.4 Connections

Connections describe how resources are connected. If there is a media stream between two resources, there is a connection between these resources. Connections are unidirectional.
Each Connection is identified by a pair of *ResourceIDs.* When listed in control functions, the from-resource is always listed first and the to-resource is listed second.

6.5 Resources

Resources are abstract representations of functionality in media devices. A resource is allocated (created) by means of the Reserve command. Resources have properties that specify their operation. A property of an RTP resource, for instance, is the codec used. Properties may have attribute. The echo canceller property of a SCN circuit resource, for instance, may have its status attribute set to ’enabled’ and its type attribute to ’G.165.’
Each Resource is identified by a *ResourceID*. *ResourceIDs* are unique within Control Sessions.

6.5.1 Resource Categories

There are several categories of resources.
Media Edge Points. These are representations of physical connections from a media device to the outside world. Media edge points are resources with a number of properties describing the nature of the connection to the outside world (DS0, RTP, ATM) and the capabilities that are supported for the outside connection. These capabilities include, for instance, tone generation, DTMF detection, etc.

1. **Media Resources.** Media resources are resources that operate on media streams but do not connect to the outside of the media device. An example of a media resource is a conference bridge. A conference bridge is a resource that receives media streams from a number of media edge points, manipulates the incoming streams and sends manipulated streams to other media edgepoints.
2. **Signaling Edge Points.** These also represent connections from a media device to the outside world. These connections are used only for transport of signaling information to and from MDCs. A signaling edge point is allocated on the media device end of interface N.A for every conference in the media device. A signaling edge point may have a script associated with it. This allows some of the connection control to be moved from the MGC to the media device, reducing the signaling traffic and the load on the MDC.

6.5.2 Resource Types

In this section the types of resources used in the model are described. Note that a media device does not have to support all resource types mentioned.

Resource types are described by their *properties*, the *actions* they support and the *events* they may generate.

- Properties describe the state of a resource. They are set when the resource is created and can be modified by means of control functions. Properties of media edge points include such things as codec used, SCN circuit ID, whether a DTMF detector is reserved for the edge point, etc. When a resource is created, not all its properties have to be set explicitly. Properties that are not set explicitly get a default value. For example, the default codec on an RTP edge point is G.711.
- Resources can perform actions on request from the controller (like *play DTMF-0 for 1 second* on a media edge point for which a tone player is reserved).
- Resources may generate events. For example, an edge point may generate an event in response to a remote connection being lost or the QoS value on a RTCP connection falling below a minimum threshold. An edge point on which a DTMF detector is active may generate an event in response to a DTMF tone being detected.

All resources have properties. Every resource type has a property *ResourceID* by means of which it can be identified within the media device. Other properties are specific to certain resource types. In the ensuing sections we introduce the resource types and the properties they have. The resource types are described in more detail in Section 11.

6.5.2.1 Media Edge Points

Media edge points represent either bi- or uni-directional interfaces to the outside world depending upon edge point type. Media edge points may also

represent an interface to channel associated in-band signaling. In this case, the media edge point needs to be able to recognize signals and generate events to notify the controller of the signal received.
The following bi-directional interfaces are supported:

1) DS0
2) NxDS0
3) DTMF trunk DS0
4) MF trunk DS0
5) NAS trunk DS0
6) SCN analog line
7) ATM VC

Uni-directional interfaces:

1) RTP Stream
2) Frame Relay VC

All media edge points have a ResourceID. Media edge points are associated with calls, so they also have a CallID and a ConferenceID. Other properties depend on the resource type. Edge points that terminate circuit networks, for instance, have an echo cancellation property; RTP edge points have an RTCP property. These two properties are examples of complex properties. A complex property is also known as a *package*. Packages have attributes that describe in detail how they operate.

6.5.2.2 Media resources

Media resources are very similar to media edge points. The difference is that they connect streams within media devices, instead of connecting to the outside world. Currently, the only example of a media resource that we have is a conference bridge.

6.5.2.3 Signaling Edge Points

Signaling edge points represent signaling interfaces between media devices and the outside world. They currently come in three different types, viz. MDCP signaling edge point, H.245 signaling edge point and facility associated signaling edge point. The MDCP signaling edge point is used to receive and send MDCP messages from/to the controller. MDCP signaling edge points have a scripting property. By associating scripts with MDCP signaling edge points, it is possible to put extra intelligence in media devices. Scripts prescribe how to handle events generated by resources without sending Indications up to the controller. If a script does not say how a certain event may be handled, then an Indication is still sent to the controller.
The H.245 signaling edge point is for receiving and sending H.245 messages. H.245 messages received on a H.245 signaling edge point may be backhauled to the controller or they may be handled by means of a script in the associated MDCP signaling edge point.
In some cases it is desirable to have facility associated signaling terminate at media gateways (see ITU reference architecture in Section 4.2). In order to support this, we introduce a FAS edge point. Its only function is to relay messages.

6.5.3 Packages

Some properties are simple numbers, while others indicate a resource’s ability to perform certain functions. The latter, more complex properties are also knows as packages. A number of packages are supported, such as announcement playing, DTMF detection, DTMF generation, echo cancellation, RTCP, etc. Section 15 contains an overview of the packages. The packages listed form an initial set. More packages may be added.

6.5.4 Vendor Specific Extensions

Vendors may extend standardized resources and may create new ones. Standardized resources may be extended with new properties, actions and events. Standardized properties, actions and events may *not* be modified. Vendor specific additions receive:

1. A an identification that shows it is a vendor specific extension
2. A vendor ID that follows H.245 and consists of three sub-fields as defined by H221NonStandard of ITU-T Recommendation H.225.
3. A number unique per vendor.

This scheme is used both for Vendor specific resources and properties, actions and events.

7 Messages

Control of the MD by the MDC takes place through the exchange of messages between the MDC and MD.

7.1 Message Structure

Each message is made up of a number of control functions (See Section 0). A sequence of related control functions makes up a transaction. Messages may contain information relevant to several concurrent transactions and concurrent sessions.

The message must contain an integral number of control functions. The maximum number of control functions that may be included in a message varies as the maximum length of the messages may vary. The maximum length of the message is negotiated during the registration procedure.

7.2 Transport

Messages may be carried over a reliable or unreliable underlying network.

Since an unreliable transport protocol is assumed, all messages are explicitly acknowledged. Acknowledgment is performed per session. For this, each command is accompanied by three values:

- Identification of the conference, call, or resource to which the command pertains (ConferenceID, CallID, or ResourceID).
- *Send Sequence Number* –This value allows the sender to determine when the receiver of the command has returned receipt confirmation.
- *Receive Sequence Number* – the *Send Sequence Number*

expected in the next command. This value serves as a receipt confirmation of previously received commands.

-

7.3 Security

Each control session may be encrypted. A *CryptoToken* field is included in the message header to provide the security mechanism and is based on the procedures defined in the ITU-T Recommendation H.235. The use of encryption is determined during the registration procedure. Initial encryption keys and algorithms will typically be entered in the MDC and MD manually. Updates to the encryption keys may be done a Modify command on the SessionKey parameter of the control session resource. An alternative is to use the Registration control function, in that case the command may be issued by both MGC and MG. Integration with the framework in H.235 is for further study.

8 Control Functions

Control of the MD by the MDC is performed using a set of control functions. Control functions are the means whereby specific commands, indications and responses are conveyed between the MDC and MD.

8.1 Notation

The control functions are denoted in EBNF syntax, A short recap follows below:

STRING	a token
<identifier>	an identifier
<def> ::= ………	a definition
<clause1> | <clause2>	clause 1 or clause 2
{ <clause> }	any finite number of clause
[<clause>]	clause is optional
// comment	

8.2 Control Function Definitions

There are two kinds of control functions, connection related functions and management functions.

8.2.1 Connection control functions

The connection can be executed by any controller. Access control is performed through the security mechanism.

The functions are:

1. **Reserve** *Reserve* is the command used toreserve resources in a media devices for setting up a conference.
2. **ReserveConfirm** *ReserveConfirm* is the response from the MD to a successful Create command.
3. **ReserveReject** *ReserveReject* is the response to a failed Reserve command.
4. **Connect** *Connect* is the command from the MDC to the MD to establish a *Connection* in the MD. The command connects resources for a certain conference.
5. **ConnectConfirm** *ConnectConfirm* is the response from the MD to a successful Connect command.
6. **ConnectReject** *ConnectReject* is the response to a failed connect command.

7. **Disconnect** *Disconnect* is the command to remove the connections between two resources.
8. **DisconnectConfirm** *DisconnectConfirm* is the response to a successful *Disconnect* command
9. **DisconnectReject** *DisconnectReject* is the response to a failed *Disconnect* command
10. **Modify** *Modify* is the command from the MDC to the MD to change the properties of a resource within the MD.
11. **ModifyConfirm** *ModifyConfirm* is the response from the MD to a *Modify* command.
12. **ModifyReject** *ModifyReject* is the response to a failed modify command.
13. **Request** Request an action of a resource.
14. **RequestConfirm** Positive response to the Request command.
15. **RequestReject** Negative response to the Request command
16. **Indication** *Indication* is a control function sent from the MD to the MDC in response to an event occurring within the MD. E.g. a connection being lost or a DSP board breaking down. Events may be caught in a scripting processor in that case no indication follows upon the event. All events that are not caught internally are sent to the controller with mention if the resource object that caused them.
17. **Delete** *Delete* is the command from the MDC to the MD to delete a Conference, Call, Connection or Resource within the MD.
18. **DeleteConfirm** *DeleteConfirm* is the response from the MD to a *Delete* command.
19. **DeleteReject** *DeleteReject* is the response to an unsuccessful *Delete* command
20. **Query** *Query* is the command from the MDC to the MD requesting information. The *Query* command can be used to get all kinds of information from the MD.
21. **QueryResponse** *QueryResponse* is the response from the MD to the *Query* command.
22. **QueryReject** *QueryReject* is the response to a failed *Query* command.
23. **Acknowledgment** *Acknowledgment* is a dummy control function sent to acknowledge the correct receipt of a specific message when no return message is pending for a Conference. See also Section 9 for implicit acknowledgments.

8.2.2 Management control functions

Only one entity at a time may execute the management control functions. These functions are used to control access to the media device and to perform real-time resource management. Note that a managemer may also act as a connection controller and execute the commands described above.

1. **Registration** *Registration* is a control function from either the resource manager or the media device requesting initiation of the Registration transaction necessary for a control session.
2. **RegistrationResponse** *RegistrationResponse* is the

response from either the resource manager or the media device to a *Registration* control function.

3. **RegistrationReject** *RegistrationReject* is the error to a *Registration* or a *RegistrationResponse* command.
4. **Termination** *Termination* is a control function sent by either the media device or resource manager to terminate a control session or registration transaction.

8.3 Control Function Formats

8.3.1 General

Each control function is identified by a unique *ControlFunctionID*, and a identification of the conference, call, or resource to which the control function applies, as well as a send and receive sequence number, described earlier. A number of other parameters, specific to each control function type are associated with the control function.

All reject control function take the following form.

```
<genericrej> ::= <ConferenceID> <SendSeqNo> <RcvSeqNo>
                   {<error>}

<error> ::= [<ConferenceID>|<CallID>|<ResourceID>] <ErrorValue>
              [<ErrorparamX>][<ErrorparamY>][<ErrorText>]
```

Defined error messages are :

Error	Parameters	Meaning
E_SYN		Syntax error in message
E_PusR	X,Y	Property X unsupported[1] in Resource Y
E_PunR	X,Y	Property X unavailable[2] in Resource Y
E_RusT	X	Resource Type X unsupported
E_RunT	X	Resource Type X unavailable
E_Val	X,Y	Value X out of range for resource Y
E_AusR	X,Y	Action X unsupported for Resource Y
E_AunR	X,Y	Action X unavailable for Resource Y
E_RnID	X	Resource ID X does not exist

[1] Unsupported means not implemented in the media device.

[2] Unavailable means currently not available because resources are depleted

E_RID	X	Resource ID X already exists
E_nID	X	ID X does not exist
E_ID	X	ID X already exists
E_nCON	X,Y	Connection between X and Y does not exist
E_2ID		Two IDs in Connection required
E_OoR		Out of resources
E_SECU		Rejected for security reasons

For each reject message only the allowable error values will be given.

8.3.2 Reserve

The Reserve command is used to reserve resources (edge points and media resources) for a conference. The command has a number of mandatory- and a number of optional parameters. The parameters are used to set properties of the resource that is reserved. For instance, the Reserve command may be used to reserve echo cancelling functionality on an edge point that is created by setting the status attribute of the EchoCanceller property to 'Reserved.'

```
<Reserve> ::= RESERVE <ConferenceID> <SendSeqNo> <RcvSeqNo>
 (<NewResource>)

<NewResource> ::= <ResourceTypeID> {<property>}

<property>:== <setproperty> | <multicardproperty>

<setproperty>:==    <type><value>
// WHERE VALUE MAY BE WILDECARDED
<multicardproperty>:==    ({<setproperty>})
```

A multi card is used when the controller wishes to allow the media device to choose from a set of possibilities. E.g. for H.323 FastStart where the remote site is offered

8.3.3 ReserveConfirm

The ReserveConfirm command is sent as a positive response to the Reserve command. It gives information about the resource that has been reserved and properties that were filled in by the media device. The command takes the following form:

```
<Resconf> ::= <ConferenceID> <SendSeqNo> <RcvSeqNo>
                    { <ResourceID> <props> }
```

8.3.4 ReserveReject

Legal errors are: E_SYN, E_PusR, E_PunR, E_RusT, E_RunT, E_Val, E_RID, E_Oor, E_SECU

8.3.5 Connect

Using the *Connect* command new connections are created or additions are made to existing connections. *Connect* can be used to create a connection with multiple branches in a single command.

The connection created with this command is unidirectional. The first resource in the command is the source of the connection and the second is the sink.For efficient operation, the command allows new resources to be reserved and connected together in one go.
The command takes the following form:

```
<connect> ::= CONNECT <ConferenceID> <SendSeqNo> <RcvSeqNo>
                  {<newconnection>}

<newconnection>:==  (<connfrom>,<connto>)

<connfrom>:==<resourceornew>
<connto>:==<resourceornew>

<resourceornew>:==<NewResource> | <ResourceID>
```

8.3.6 ConnectConfirm

The connect confirm is sent as a positive response to the Connect command. It gives information about the connection that has been established and properties of resources that are filled in by the MD. The command takes the following form:

```
<connconf> ::= <ConferenceID> <SendSeqNo> <RcvSeqNo>
                  { <ResourceID> { <property> } }
```

8.3.7 ConnectReject

Legal errors are: E_SYN, E_PusR, E_PunR, E_RusT, E_RunT, E_Val, E_RID, E_RnID, E_2ID, E_Oor, E_SECU

8.3.8 Disconnect

The Disconnect command is used to remove the connection between two resources.

```
<disconnect> ::= DISCONNECT <ConferenceID> <SendSeqNo> <RcvSeqNo>
                  <CallID> {<connection>}

<connection>:==     (<connfrom>,<connto>)

<connfrom>:==<ResourceID>
<connto>:==<ResourceID>
```

8.3.9 DisconnectConfirm

```
<disc> ::= DISCCONF <ConferenceID> <SendSeqNo> <RcvSeqNo>
```

8.3.10 DisconnectReject

The following errors are valid: E_SYN, E_nID, E_nCON, E_SECU

8.3.11 Modify

Modify is the command from the MDC to the MD to change a the properties, or attributes thereof, of a resource within the MD.
The command takes the following form:

```
<mod> ::= Modify <ConferenceID> <SendSeqNo> <RcvSeqNo>

              <resourceID> {<property>}
```

8.3.12 ModifyConfirm

The command takes the same form as the CreateConfirm

8.3.13 ModifyReject

Legal errors are: E_SYN, E_PusR, E_PunR, E_Val, E_RnID, E_Oor, E_SECU

8.3.14 Request

The Request command is used for requesting resources to perform actions. The command takes the following form:

```
<request>::= <ConferenceID> <SendSeqNo> <RcvSeqNo>
                 <ResourceID> <ActionID> <props>
```

8.3.15 RequestConfirm

The command takes the same form as CreateConfirm.

8.3.16 RequestReject

Legal errors are: E_SYN, E_PusR, E_PunR, E_AusR, E_AunR, E_Val, E_RnID, E_Oor, E_SECU

8.3.17 Indication

The command takes the following form:

```
<ind> ::= <ConferenceID> <SendSeqNo> <RcvSeqNo>
                 <resourceID> <EventID> [<value>]
```

8.3.18 Delete

Delete is the command from the MDC to the MG to delete a Conference, Connection or resource within the MD. Deleting a resource object will free it up for use in

```
<del>::= Delete <ConferenceID> <SendSeqNo> <RcvSeqNo>

              { <CallID> | <connectionID> | <resourceID> }
```

other Connections. The delete command takes the following form:

8.3.19 DeleteConfirm

The *DeleteConfirm* response is empty and indicates a positive confirmation of a *Delete* command.

```
<delc> ::= DELCONF <ConferenceID> <SendSeqNo> <RcvSeqNo>
```

8.3.20 DeleteReject

Legal errors are: E_SYN, E_RnID, E_SECU

8.3.21 Query

The query command can be used to query the state of the MD as a whole or of individual objects in the model. The command contains one or more of the following parameters:

Parameter	Purpose
Info	returns general information on the MD, vendor and type
AllConferences	returns a list of all *ConferenceIDs*
<ConferenceID>	returns a list of all *CallIDs*
FreeResources	returns a list of supported resources

<ResourceType>	returns a list of supported properties of a resource type
<CallID>	returns a list of resources involved in a call
<ResourceID>	returns resource properties

The Query command can be used for many uses. For instance, a *Query* on an RTCP enabled IP connection could give the QoS statistics.

8.3.22 QueryResponse

The command takes the following form:

```
<QueryResponse>::= <ConferenceID> <SendSeqNo> <RcvSeqNo> (
                 <string> // for Info
                 | {<conferenceID>} // for AllConferences
                 | {<CallID> //for <ConferenceID>
                 | {<resourcetype>} //for FreeResources
                 | {<ResourceID>} // for <CallID>
                 | {<property>} // for <resourceID>
            )

<resourcetype> ::= <EdgePointType> | <MediaResourceType>
```

The case when a query response becomes too long for the message size (see below) is for further study.

8.3.23 QueryReject

Legal errors are: E_SYN, E_RnID, E_nID, E_Oor, E_SECU

8.3.24 Registration

The command has the following form:

```
<reg> ::= Register <ConferenceID> <SendSeqNo> <RcvSeqNo>
            (Primary|Secondary|Peer) {<regparam>}


<regparam>::= ControllerAddress <value>
             | ControlleeAddress <value>
             | ProtocolVersionID <value>
             | BufferInController <value >
             | BufferInControllee <value >
             | hartbeatInterval <value >
             | ackTimer <value >
             | resultTimer <value >
             | KeyUpdate <value>
```

The value <ackTimer> is used to indicate the amount of time in which a command must be acknowledged. The value is defined in milliseconds a typical value would be less than 500 milliseconds.

The value of <resultTimer> gives the maximum time within which a command must return a result. The value is defined in millseconds, a typical value would be

less than a second.
The media device will send a heartbeat *Indication* every <heartbeatInterval> seconds. The reception of this *Indication* will be acknowledged by the controller. The value is defined in seconds, a typical value might be 10 seconds.

8.3.25 RegistrationResponse

The command takes the following form:

```
<regresp> ::= <ConferenceID> <SendSeqNo> <RcvSeqNo> (<reg> | <ok>)
```

8.3.26 RegistrationReject

Legal errors are: E_SYN, E_Oor, E,SECU

8.3.27 Acknowledgment

The command takes no parameters as its only purpose is to acknowledge receipt of a message. This is achieved by the SendSeqNo and RcvSeqNo values in the message.

```
<ack> ::= ACK <ConferenceID> <SendSeqNo> <RcvSeqNo>
```

8.3.28 Termination

The command has the following form:

```
<term> ::= Terminate <ConferenceID> <SendSeqNo> <RcvSeqNo>
           (<Primary|Secondary>) <ControllerAddress>
           <ControlleeAddress>
```

9 Encoding

The section describes the binary encoding scheme that is proposed. It is still work in progress!

9.1 Message Encoding

The general format for the encoding of messages is shown below.

Note that a protocol ID is not necessary because that has been negotiated during the registration.

<table>
<tr><td>Include Crypto</td><td>Message Length</td></tr>
<tr><td colspan="2">Crypto Token (optional)</td></tr>
<tr><td colspan="2">Message Contents</td></tr>
</table>

Include Crypto indicates if the *Crypto Token* is included in the message. It is a single bit and is set to 1 to indicate the *Crypto Token* is included; otherwise, this bit is set to 0.

Message Length specifies the length of the *Message Contents* in octets; message length is a 15 bit unsigned integer quantity and is transmitted most significant bit first.

The *Crypto Token* the encoding of the crypto token is for further study; it will follow H.235 but may not use ASN.1 since we want to be able to handle very simple devices.

Message Contents is a series of one or more Control Functions. The message must contain an integral number of Control Functions and the maximum number included in a message varies as the length of the messages may vary. However, the total length of this field must be less than or equal to the value negotiated during the registration procedure.

9.2 Control Function Encoding

The encoding of Control Functions consist of two parts, a fixed header whose format is the same for all Control Functions and a variable portion consisting of elements that are Control Function specific as shown in the figure below.

ControlFunctionID
Conference ID
SendSequenceNumber
ReceiveSequenceNumber
Length
Elements

ConferenceID determines to which conference the command applies. The element *Identifier* is a 32 bit unsigned integer that is transmitted most significant bit first.

Send Sequence Number –This value allows the sender to determine when the receiver of the command has returned receipt confirmation. The length of the *Send Sequence Number* is one octet. The sequence number is incremented by one (modulo 128) for each command sent with the exception of the *Acknowledgment* command. In the case of the *Acknowledgment* command, this field should contain the value of the previous command's send sequence number. When an *Acknowledgment* command is received, the value of the *Send Sequence Number* shall be ignored.

Receive Sequence Number – the *Send Sequence Number* expected in the next command. This value serves as a receipt confirmation of previously received commands. A *receive sequence number* of *N* acknowledges receipt of all received commands with a sequence number up to and including *N-1* (using modulo 128 arithmetic). This procedure uses a fixed length window of 127; windows of this size are not expected to be a problem since the number of pending commands at any particular time is expected to be small giving an effective window size much less than 127. The length of the *Receive Sequence Number* is one octet.

ControlFunctionID specifies the type of command. It is a single octet in length and takes a value from the table below:

Control Function ID	Control Function
00	Ack
01	Create
02	CreateConfirm
03	CreateReject
04	Connect
05	ConnectConfirm
06	ConnectReject
...	
25	RegistrationResponse
26	RegistrationReject
27	Termination

Length gives the total number of octets contained in the *elements* field. It is two octets in length and is transmitted most significant bit first

Elements give the parameters to the specified command. These elements consist of *CallIDs* , *ResourceIDs,* and parameter lists.

9.3 Element Encoding

All elements share a common format as shown in figure below.

<table>
<tr><td>Type</td><td>Identifier Length</td></tr>
<tr><td colspan="2">Identifier</td></tr>
<tr><td colspan="2">Content Length</td></tr>
<tr><td colspan="2">Contents</td></tr>
</table>

Type specifies the category of the element. It is 3 bits in length and takes one of the values from the table below:

Element Category	Type
Standard Resource	0
Vendor Specific Resource	1

Call	2
Property	3
Vendor Specific Property	4
Event	5
Vendor Specific Event	6
Action	7
Vendor Specific Action	8
etc...	

All values not specified in the above table are reserved for future use.

The *Identifier Length* field gives the length of the element identifier in octets; this field is 5 bits in length.

The encoding of the *Identifier* field are dependent on the value of the *Type* field and are specified below.

The *Content* field is specific to the specified element and may not be present at all; this field gives the parameters associated with the element.

9.3.1 Standard Resources

Standard Resources types have an *Identifier* format as shown below:

ResourceTypeID

ResourceID gives the specific invocation number and is a single octet in length. This is unique within the Control Session and is a single octet in length.

9.3.2 Vendor Specific Resources

The encoding of Vendor Specific Resource types must allow for the identification of the specific vendor. To achieve this goal, the vendor specific resources have an *Identifier* format as shown below:

VendorID
ResourceTypeID

The module numbering is vendor specific.

The *VendorID* determines which vendor specified the resource. . (The ID follows H.245) It is 4 octets in length.. The Vendor ID consists of three sub-fields as defined by H221NonStandard of ITU-T Recommendation H.225. It is encoded as follows:

The first octet of the Vendor ID specifies the t35Country code as defined in ITU-T Recommendation T.35.

The second octet specifies the t35Extension and is assigned nationally.

The third and forth octets are treated as an unsigned 16 bit integer quantity that specifies the manufacturer code. This field is assigned nationally.

ResourceTypeID is a number within the scope of the vendor ID.

9.3.3 CallID

The CallID element *Identifier* is a 32 bit unsigned integer that is transmitted most significant bit first.

10 Resources Types

10.1 Media Edge Points Category

As explained before, media edge points represent interfaces between media devices and the outside world for media channels. Some media channels carry in-band signaling, meaning that media edge points may also generate events to relay signaling information to the controller. This section gives an overview of media edge point types and their properties.

10.1.1 DS0

Properties

Name	Type (value range)	Meaning
ResourceID	Number	Unique identity of the resource in the current control session
ConferenceID	Number	Identifies the conference to which the resource is assigned.
CallID	Number	Identifies the call to which the resource is assigned.
InResourceID	Number	Identity of the resource that is the source for the stream sent out
OutResourceID	Number	Identity of the resource that this resource sends media to.
Circuit	Number	
Facility	Number	Facility and Circuit identify DS0 Circuit.
SCN Voice	Package	
Echo cancel	Package	
Call progress tones	Package	
Announcement play	Package	
DTMF detect	Package	

Actions

Name	Type (value range)	Meaning
Continuity test	(co1, co2)	Generate continuity test tones (single tone or return tone, go tone in dual tone procedure).

Events

Name	Type (value range)	Meaning
Continuity Test	(co1)	Continuity tone detected.

NAS DS0
Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		
CallID		
InConnectionID		
OutResourceID		
Connection type	PPP/SLIP/...	
Connect speed		(to set max)
IP address		
Header compression	Boolean	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning
Connection lost		
Connected	speed, type	

10.1.2 SCN Analog
Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		
CallID		
InConnectionID		
OutConnectionID		
SCN voice	Package	
Echo cancel	Package	

Actions

Name	Type (value range)	Meaning
metering pulse	length milliseconds	
Reverse polarity		
Power ring		
... *more channel associated signaling*		

Events

Name	Type (value range)	Meaning
on-hook		
off-hook		
Flash		

10.1.3 ATM VC
Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		
CallID		
InConnectionID		
OutConnectionID		
Address	ATMaddress	
VCC		use this one,

VCI/VPI		or this one
Codec	RTPtype	
QoS mode		
Media crypt	Package	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning
Connection lost		

10.1.4 RTP Stream

Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		
CallID		
Direction	in/out	
PeerResourceID		Identifies the resource to which media is sent or from which it is received.
MyIP address		
MyRTP port		
RemodeIP address		
RemoteRTP port		
Silence suppression	boolean	
codec	RTP type	
Jitter buffer size		
packetization		
Frames per packet		
Media crypt	Package	
RTCP	Package	
RSVP	Package	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning
Connection lost		

10.2 Media Resources Category

10.2.1 Conference Bridge

Properties

Name	Type (value range)	Meaning
NrPorts	Number	Number of parties connected to bridge.
ConferenceID	Number	Identification of conference.
DTMF detect		
Announcement		
Playing		

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning

10.3 Signaling Edge Points

Signaling edge points transport signaling information to and from media devices. MDCP messages, for instance, terminate at signaling edge points. Currently, four types of signaling edge points are supported: MDCP signaling edge points, management signaling edge points, H.245 signaling edge points and facility associated signaling edge points.

10.3.1 MDCP Signaling Edge Point

When a new control connection is built, a new MDCP signaling edge point is created. Creation of MDCP signaling edge points occurs when the media device receives Reserve or Connect command with a new ConferenceID. If the ResourceType in a Reserve or Connect command is MDCPSignal, the creation is explicit. A MDC may also create a resource of a different type (i.e., not of type MDCPSignal) with a new ConferenceID. In this case, the creation of the signaling edge point for the new conference is implicit.

One of the reasons for explicitly having MDCP signaling edge point is their scripting property. The scripting property allows the amount of communication between MDC and media device to be reduced. A script serves to catch events generated by resources and to handle them, where these events would normally have been sent to the MDC in Indication messages. If a script cannot handle a certain event, an Indication will still be sent.

Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		
MyAddress		Address of the interface on the MD (e.g. IP addr+UDP port)
ControllerAddress		IP Address of the interface of the MDC (e.g. IP addr+UDP port)
Protoversion		Protocol version ID
BufferInController		
BufferInControllee		
SessionKey		Encryption key of the control session
Script	Package	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning .

10.3.2 Management Signaling Edge Point

This is a special type of the edge point that terminates the communication channel to the resource manager. It has some additional Events and Actions:

Properties

Name	Type (value range)	Meaning
ResourceID		
MyAddress		Address of the interface on the MD (e.g. IP addr+UDP port)
ControllerAddress		IP Address of the interface of the resource manager (e.g. IP addr+UDP port)
Protoversion		Protocol version ID
BufferInController		
BufferInControllee		
HeartBeatInterval		
AckTimer		
ResultTimer		
SessionKey		Encryption key of the control session

Actions

Name	Type (value range)	Meaning
Handover	OldControlAdress, newControlAddress	To hand control of conferences (and associated calls and resources) to a new controller.

Events

Name	Type (value range)	Meaning
Heartbeat		For the semantics of the Heartbeat see Section 6.1.2.2.
Resource broken	Resource Type, explanation	To indicate that hardware (functionality) has broken down, or has been removed

New Resources detected	Resource Type, explanation	To indicate that new hardware (functionality) has been detected
Control connection severed	ControllerID	To indicate that a connection with a certain controller has been lost.

The AckTimer indicates the amount of time that may elapse between a message being sent and an acknowledgment being received. The media device takes this value into account in determining when to resend messages. If no acknowledgment is received after three retries, then the media device may assume that the communication channel is faulty. Conversely, if a media device fails to respond to a message it receives within the AckTimer constraint, then the controller will resend the message. If the controller does not receive an acknowledgment after three retries, then it may assume the communication channel to be faulty and let the conference terminate.
The ResultTimer has a similar purpose. It pertains, however, to the amount of time allotted to confirmation and rejection messages. If no result message is received after three retries, then the device waiting for the result may assume that the communication partner is faulty. Conferences cannot be guaranteed to be kept alive in this case.
The values of AckTimer and ResultTimer apply to all command exchanges the media device is involved in, for all signaling interfaces to all controllers (both management and connection control).

10.3.3 H.245 Signaling Edge Point

H.245 may be supported in two ways: as backhaul or autonomous. H.245 backhaul is performed by sending H.245 messages received on the H.245 edge point up to the MDC, through the associated MDCP signaling edge point. Autonomous operation is performed by associating a script with the MDCP signaling edge point.
If a conference is being set up between IP users, multiple H.245 edge points may be associated with a MDCP signaling edge point. There may be a H.245 signaling edge point for every call in the conference.
By means of a script associated with the MDCP signaling edge point, the MD may negotiate the media connection parameters autonomously.
The definition would be:

Properties

Name	Type (value range)	Meaning
ResourceID		
ConferenceID		

CallID		
Signaling type	H.245	
MyAddress		IP address + port number of the edge point.
Remote Address		IP address + port number of remote side.

Actions

Name	Type (value range)	Meaning
SendMessage		

Events

Name	Type (value range)	Meaning
MessageReceived		

10.3.4 FAS Edge Point

For some applications it may be desirable or necessary to do some of the signaling from the media gateway box rather than from the controller. A separate edge point type exists to pass on facility associated signaling (from MG to MGC in *Indication* control functions, from MGC to MG in *Request* control functions). The contents of these messages is for further study, it may be the SS7 message, a part of it or the information in the form of a meta protocol.

In the case of the connection setup protocols like ISUP/SS7, Q.931 and H.225 one signaling resource instance per signaling connection would be required this would be able to cover all calls relevant to that signaling connection. The definition of these is for further study.

Properties

Name	Type (value range)	Meaning
ConferenceID		
ResourceID		
Signaling type	ISUP/SS7, Q.931, H.225, H.245, QSIG etc.	
Facility		
Circuit		

Actions

Name	Type (value range)	Meaning
SendMessage		

Events

Name	Type (value range)	Meaning
MessageReceived		

11 Packages

In this Section we describe the properties with type Package in some more detail. These complex properties have a number of attributes. The attributes are to the properties what properties are to resources.

11.1 Announcement Player

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning
Start	MessageID, Duration	
Record	MessageID, MaxDuration	
Stop	MessageID	

Events

Name	Type (value range)	Meaning
MessageReceived	MessageID	

11.2 SCN Voice

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning
SetInGain	value	Set gain level for input channel from SCN
SetOutGain	Value	Set gain level for output channel to SCN

Events

Name	Type (value range)	Meaning
Activity Detected		

11.3 Call Progress Tones Player

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning
Start	ToneID[3], Duration (ms)	
Stop	ToneID	

Events

Name	Type (value range)	Meaning

11.4 DTMF Detector

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning
DTMFReceived	Digit (0-9,A-D) Duration (ms)	

11.5 DTMF Generator

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning
Start	ToneID (0-9, A-D), Duration (ms)	
Stop	ToneID	

Events

Name	Type (value range)	Meaning

11.6 Echo Canceller

Attributes

Name	Type (value range)	Meaning

[3] A list will be created to enumerate the tones.

Status	(Unavailable, Available, Unsupported, Reserved, Active)	
Type	G165, G168 etc.	To specify the type of echo canceller

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning

11.7 Media crypt

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	
Algorithm	(enumeration of encryption algorithms)	

Actions

Name	Type (value range)	Meaning
SetKey	key	
GetKey	key	

Events

Name	Type (value range)	Meaning

11.8 RTCP

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	
MyRTCPport	1-65535	
QoSparam		Can be used to read or measured Properties, or set to let MD detect underrun.
RemoteRTCPport	1-65535	

Actions

Name	Type (value range)	Meaning

Events

Name	Type (value range)	Meaning
QoS too low	QoS params	

11.9 Scripting

Scripting is a property of MDCP signaling edge points. A script describes how events generated by resources may be handled but also autonomous scripts/programs may be created. If the script does not describe how an event is handled, the default action is to send an Indication message to the controller controlling the resource that generated the script. Event handling may involve issuing commands to any resource associated with the MDCP signaling edge point.

The scripting resource is a very important part of the working of a MD. Scripting will allow the low level commands of the MDCP protocol to be aggregated into more complex activities.

An example would be to play a message, collect digits, play another message, collect more digits. (like the requirements described in draft-cromwell-navdec-media-req-00) At the level of the basic command set, this would require many messages to be exchanged between the controller and the media device. The scripting property allows the media device to handle the events by itself.

The scripting is for further study. The requirements for the language depend on the applications and the size of the MD. Possible candidates are Java, Perl, and Python, but also a new scripting language could be defined.

Attributes

Name	Type (value range)	Meaning
Status	(Unavailable, Available, Unsupported, Reserved, Active)	

Actions

Name	Type (value range)	Meaning
Download	Script ID, code	Download the scripting code and assign an ID to it
Delete	Script ID	Delete storage for script
StartScript	Script ID	Start execution of specified script.
StopScript	Script ID	Stop execution of current script.

Events

Name	Type (value range)	Meaning

Script IDs are valid in a certain scope. If the download is applied to a media or signaling edge point the scope is that edge point. If Download and Delete are applied to a control edge point, the scope of the script ID is all resources controlled from that control channel. If the commands are applied to the management edge point, the scope is the entire box.

11.9.1 Dial plans

Dial plans are a kind of scripting, they can be downloaded and attached to calls in the same way as scripts. Like the scripting language, the dial plans are for further definition.

12 Author's Addresses

Paul Sijben
Lucent Technologies
P.O. Box 18
1270 AA Huizen
The Netherlands
Tel: +31 35 687 4774
Mail sijben@lucent.com

John Wachter
Lucent Technologies
PO Box 636600-700 Mountain Avenue
Murray Hill, NJ 07974-0636 U S
U.S.A.
Tel: +1 908 582 2761
Mail: jwachter@lucent.com

Chia Li
Lucent Technologies
101 Crawfords Corner Rd
Holmdel, NJ 07733-3030
U.S.A.
Tel: +1 732 949 3101
Mail: chiali@lucent.com

Mike Buckley
Lucent Technologies
Mail: mikebuckley@attmail.com

John Segers
Lucent Technologies
P.O. Box 18
1270 AA Huizen
The Netherlands
Tel: +31 35 687 4724
Mail: **jsegers@lucent.com**

Rex Coldren
AG Communication Systems
P.O. Box 52179
2500 West Utopia Road
Phoenix, AZ 85072-2179
U.S.A.
Tel: +1 602 993 0500
Mail: coldrenr@agcs.com

13 Revision History

Version	**Date**	**Description**
0.0	13 November 1998	Sent to SG16 Torino meeting
0.1	30 November 1998	Revised structure and fixed some open issues; released as <**draft-sijben-megaco-mdcp-00.txt**> and TIPHON 11TD29.
0.2	18 December 1998	Restructured model and implications on commands. Added call flows. Released as TIPHON 11TD29r1.
0.3	24 January 1999	Simplified model and adapted commands to reflect this. Removed call flows to separate document.