

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.248.1

(03/2013)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Communication
procedures

Gateway control protocol: Version 3

Recommendation ITU-T H.248.1



International
Telecommunication
Union

Documents
Document Sharing Hub

ITU-T H-SERIES RECOMMENDATIONS
AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.248.1

Gateway control protocol: Version 3

Summary

To achieve greater scalability, this Recommendation decomposes the ITU-T H.323 Gateway function defined in Recommendation ITU-T H.246 into functional subcomponents and specifies the protocols these components use to communicate. This allows implementations of ITU-T H.323 gateways to be highly scalable and encourages leverage of widely deployed Switched Circuit Network (SCN) capabilities such as SS7 switches. This also enables ITU-T H.323 gateways to be composed of components from multiple vendors distributed across multiple physical platforms. The purpose of this Recommendation is to add capabilities currently defined for ITU-T H.323 systems and is intended to provide new ways of performing operations already supported in ITU-T H.323.

This revision contains a number of clarifications on:

- Signal behaviour on new Streams;
- time based Statistics and their applicability to multi flow streams;
- the impact of the Mode property on Statistics;
- Event behaviour with respect to new Streams;
- ServiceChange version handling.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.248	2000-06-15	16
1.1	ITU-T H.248.1 v1.1	2002-03-29	16
2.0	ITU-T H.248.1 v2	2002-05-22	16
2.1	ITU-T H.248.1 v2 (2002) Cor. 1	2004-03-15	16
3.0	ITU-T H.248.1 v3	2005-09-13	16
3.1	ITU-T H.248.1 v3 (2005) Amd. 1	2008-05-02	16
3.2	ITU-T H.248.1 v3 (2005) Amd. 2	2009-12-14	16
4.0	ITU-T H.248.1 v3	2013-03-16	16

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2013

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope	1
2	References.....	1
3	Definitions	4
	3.1 Terms defined elsewhere	4
	3.2 Terms defined in this Recommendation.....	4
4	Abbreviations and acronyms	5
5	Conventions	7
	5.1 Key words to indicate requirement levels	7
	5.2 Connection endpoint naming conventions	7
6	Connection model.....	9
	6.1 Contexts.....	10
	6.2 Terminations.....	11
	6.3 Wildcarding principles	17
7	Commands	19
	7.1 Descriptors.....	20
	7.2 Command application programming interface.....	45
8	Transactions.....	60
	8.1 Common parameters.....	62
	8.2 Transaction application programming interface.....	62
	8.3 Messages.....	64
9	Transport.....	65
	9.1 Ordering of commands	65
	9.2 Protection against restart avalanche	66
	9.3 Protection against Notify avalanche.....	67
10	Security considerations	67
	10.1 Protection of protocol connections.....	67
	10.2 Interim AH scheme.....	67
	10.3 Protection of media connections	68
11	MG-MGC control interface	68
	11.1 Multiple Virtual MGs	69
	11.2 Cold start	69
	11.3 Negotiation of protocol version.....	70
	11.4 Failure of an MG	70
	11.5 Failure of an MGC.....	70
	11.6 MGC-MG control association monitoring	71
	11.7 Compatibility.....	71
12	Package definition	72



	Page
12.1 Guidelines for defining packages	73
12.2 Guidelines to defining parameters to events and signals.....	78
12.3 Identifiers.....	78
12.4 Package registration.....	79
13 Profile definition.....	79
14 IANA ITU-T H.248 registrations	79
14.1 Package registration procedure.....	79
14.2 Error code registration procedure	81
14.3 ServiceChange reason registration procedures.....	81
14.4 Profile registration procedures	82
Annex A – Binary encoding of the protocol	83
A.1 Coding of wildcards	83
A.2 ASN.1 syntax specification	84
A.3 DigitMaps and path names	101
Annex B – Text encoding of the protocol.....	103
B.1 Coding of wildcards	103
B.2 ABNF specification	103
B.3 Hexadecimal octet coding	118
B.4 Hexadecimal octet sequence.....	118
Annex C – Tags for media stream properties	119
C.1 General media attributes.....	119
C.2 Mux properties.....	120
C.3 General bearer properties	120
C.4 General ATM properties.....	121
C.5 Frame relay	123
C.6 IP.....	123
C.7 ATM AAL 2	124
C.8 ATM AAL 1	125
C.9 Bearer capabilities	126
C.10 AAL 5 properties.....	134
C.11 SDP equivalents.....	134
C.12 H.245	135
Annex D – Transport over IP.....	136
D.1 Transport over IP/UDP using Application Level Framing (ALF)	136
D.2 Using TCP	139
Annex E – Basic packages	141
E.1 Generic	141
E.2 Base Root Package	143
E.3 Tone Generator Package.....	147

	Page
E.4 Tone Detection Package	148
E.5 Basic DTMF Generator Package	152
E.6 DTMF Detection Package	153
E.7 Call Progress Tones Generator Package.....	156
E.8 Call progress tones detection package.....	157
E.9 Analog Line Supervision Package.....	158
E.10 Basic Continuity Package.....	161
E.11 Network Package.....	163
E.12 RTP Package	167
E.13 TDM Circuit Package.....	171
E.14 Segmentation Package.....	172
E.15 Notification Behaviour	175
Annex F – ServiceChange Procedures.....	180
F.1 Introduction	180
F.2 Control Association Definition.....	182
F.3 Events leading to ServiceChange Procedures	182
F.4 ServiceChange Element Description	186
F.5 Use of ServiceChange parameters.....	188
F.6 ServiceChange versus TerminationState	191
Appendix I – Example call flows.....	193
I.1 Residential gateway to residential gateway call.....	193
Appendix II – ITU-T H.248 Package template.....	202
Appendix III – ITU-T H.248 profile definition template.....	206
Appendix IV – Practices on Statistics – Example use-cases	220
IV.1 Introduction	220
IV.2 Active statistics of a new termination	221
IV.3 Activating and deactivating statistics	223
IV.4 Resetting statistics	225
IV.5 Reporting statistics	226
IV.6 Reading statistics	227
Bibliography.....	231

Recommendation ITU-T H.248.1

Gateway control protocol: Version 3

1 Scope

This Recommendation defines the protocols used between elements of a physically decomposed multimedia gateway. There are no functional differences from a system view between a decomposed gateway, with distributed sub-components potentially on more than one physical device, and a monolithic gateway such as described in [ITU-T H.246]. This Recommendation does not define how gateways, multipoint control units or interactive voice response (IVR) units work. Instead it creates a general framework that is suitable for these applications.

Packet network interfaces may include IP, ATM or possibly others. The interfaces will support a variety of switched circuit network (SCN) signalling systems, including tone signalling, ISDN, ISUP, QSIG and GSM. National variants of these signalling systems will be supported where applicable.

Products claiming compliance with Version 1 of Recommendation ITU-T H.248.1 shall comply with all of the mandatory requirements of Recommendation ITU-T H.248.1 originally approved in 06/2000 and reissued in 03/2002.

Products claiming compliance with Version 2 of Recommendation ITU-T H.248.1 shall comply with all of the mandatory requirements of Recommendation ITU-T H.248.1 originally approved in 05/2002 and updated in 03/2004.

Products claiming compliance with this Recommendation shall comply with all of the mandatory requirements of Recommendation ITU-T H.248.1 approved in 03/2013.

Products shall indicate the version of the protocol in use by using ServiceChangeVersion as '1' to refer to Recommendation ITU-T H.248.1 (03/2002), '2' to refer to Recommendation ITU-T H.248.1 including its Corrigendum 1 (03/2004), and '3' to refer to Recommendation ITU-T H.248.1 (09/2005 and 03/2013).

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- | | |
|-----------------|---|
| [ITU-T E.106] | Recommendation ITU-T E.106 (2003), <i>International Emergency Preference Scheme (IEPS) for disaster relief operations.</i> |
| [ITU-T E.107] | Recommendation ITU-T E.107 (2007), <i>Emergency Telecommunications Service (ETS) and interconnection framework for national implementations of ETS.</i> |
| [ITU-T H.225.0] | Recommendation ITU-T H.225.0 (2009), <i>Call signalling protocols and media stream packetization for packet-based multimedia communication systems.</i> |

[ITU-T H.235.0]	Recommendation ITU-T H.235.0 (2005), <i>H.323 Security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems</i> .
[ITU-T H.245]	Recommendation ITU-T H.245 (2011), <i>Control protocol for multimedia communication</i> .
[ITU-T H.246]	Recommendation ITU-T H.246 (2006), <i>Interworking of H-series multimedia terminals with H-series multimedia terminals and voice/voiceband terminals on GSTN, ISDN and PLMN ISDN</i> .
[ITU-T H.248.4]	Recommendation ITU-T H.248.4 (2009), <i>Gateway control protocol: Transport over Stream Control Transmission Protocol (SCTP)</i> .
[ITU-T H.248.5]	Recommendation ITU-T H.248.5 (2009), <i>Gateway control protocol: Transport over ATM</i> .
[ITU-T H.248.8]	Recommendation ITU-T H.248.8 (2013), <i>Gateway control protocol: Error code and service change reason description</i> .
[ITU-T H.248.14]	Recommendation ITU-T H.248.14 (2009), <i>Gateway control protocol: Inactivity timer package</i> .
[ITU-T H.248.30]	Recommendation ITU-T H.248.30 (2007), <i>Gateway control protocol: RTCP extended performance metrics packages</i> .
[ITU-T H.323]	Recommendation ITU-T H.323 (2009), <i>Packet-based multimedia communications systems</i> .
[ITU-T I.363.1]	Recommendation ITU-T I.363.1 (1996), <i>B-ISDN ATM Adaptation Layer specification: Type 1 AAL</i> .
[ITU-T I.363.2]	Recommendation ITU-T I.363.2 (2000), <i>B-ISDN ATM Adaptation Layer specification: Type 2 AAL</i> .
[ITU-T I.363.5]	Recommendation ITU-T I.363.5 (1996), <i>B-ISDN ATM Adaptation Layer specification: Type 5 AAL</i> .
[ITU-T I.366.1]	Recommendation ITU-T I.366.1 (1998), <i>Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2</i> .
[ITU-T I.366.2]	Recommendation ITU-T I.366.2 (2000), <i>AAL type 2 service specific convergence sublayer for narrow-band services</i> , plus Corrigendum 1 (2002).
[ITU-T I.371]	Recommendation ITU-T I.371 (2004), <i>Traffic control and congestion control in B-ISDN</i> .
[ITU-T Q.763]	Recommendation ITU-T Q.763 (1999), <i>Signalling System No. 7 – ISDN user part formats and codes</i> , plus Amendment 3 (2004).
[ITU-T Q.765.5]	Recommendation ITU-T Q.765.5 (2004), <i>Signalling System No. 7 – Application transport mechanism: Bearer Independent Call Control (BICC)</i> .
[ITU-T Q.931]	Recommendation ITU-T Q.931 (1998), <i>ISDN user-network interface layer 3 specification for basic call control</i> , plus Amendment 1 (2002): <i>Extensions for the support of digital multiplexing equipment</i> .
[ITU-T Q.2630.1]	Recommendation ITU-T Q.2630.1 (1999), <i>AAL type 2 signalling protocol – Capability Set 1</i> .

[ITU-T Q.2931]	Recommendation ITU-T Q.2931 (1995), <i>Digital subscriber signalling system No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control</i> , plus Amendment 4 (1999).
[ITU-T Q.2941.1]	Recommendation ITU-T Q.2941.1 (1997), <i>Digital subscriber signalling system No. 2 – Generic identifier transport</i> .
[ITU-T Q.2961.1]	Recommendation ITU-T Q.2961.1 (1995), <i>Digital subscriber signalling system No. 2 – Additional traffic parameters: Additional signalling capabilities to support traffic parameters for the tagging option and the sustainable call rate parameter set</i> .
[ITU-T Q.2961.2]	Recommendation ITU-T Q.2961.2 (1997), <i>Digital subscriber signalling system No. 2 – Additional traffic parameters: Support of ATM transfer capability in the broadband bearer capability information element</i> , plus Corrigendum 1 (1999).
[ITU-T Q.2965.1]	Recommendation ITU-T Q.2965.1 (1999), <i>Digital subscriber signalling system No. 2 – Support of Quality of Service classes</i> , plus Amendment 1 (2000).
[ITU-T Q.2965.2]	Recommendation ITU-T Q.2965.2 (1999), <i>Digital subscriber signalling system No. 2 – Signalling of individual Quality of Service parameters</i> .
[ITU-T V.76]	Recommendation ITU-T V.76 (1996), <i>Generic multiplexer using V.42 LAPM-based procedures</i> , plus Corrigendum 1 (2005).
[ITU-T X.213]	Recommendation ITU-T X.213 (2001) ISO/IEC 8348:2002, <i>Information technology – Open Systems Interconnection – Network service definition</i> .
[ITU-T X.680]	Recommendation ITU-T X.680 (2008) ISO/IEC 8824-1:2008, <i>Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation</i> .
[ITU-T X.690]	Recommendation ITU-T X.690 (2008) ISO/IEC 8825-1:2008, <i>Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)</i> .
[ISO/IEC 10646]	ISO/IEC 10646 (2003), <i>Information technology – Universal Multiple-Octet Coded character Set (UCS)</i> .
[ATM Forum UNI 4.0]	Broadband Forum (1996), <i>ATM User-Network Interface (UNI) Signalling Specification – Version 4.0</i> .
[IETF RFC 791]	IETF RFC 791 (1981), <i>Internet protocol</i> .
[IETF RFC 1006]	IETF RFC 1006 (1987), <i>ISO Transport Service on top of the TCP, Version 3</i> .
[IETF RFC 2234]	IETF RFC 2234 (1997), <i>Augmented BNF for Syntax Specifications: ABNF</i> .
[IETF RFC 2327]	IETF RFC 2327 (1998), <i>SDP: Session Description Protocol</i> .
[IETF RFC 2373]	IETF RFC 2373 (1998), <i>IP Version 6 Addressing Architecture</i> .
[IETF RFC 2401]	IETF RFC 2401 (1998), <i>Security Architecture for the Internet Protocol</i> .

[IETF RFC 2402]	IETF RFC 2402 (1998), <i>IP Authentication Header</i> .
[IETF RFC 2406]	IETF RFC 2406 (1998), <i>IP Encapsulating Security Payload (ESP)</i> .
[IETF RFC 2409]	IETF RFC 2409 (1998), <i>The Internet Key Exchange (IKE)</i> .
[IETF RFC 2460]	IETF RFC 2460 (1998), <i>Internet Protocol, Version 6 (IPv6) Specification</i> .
[IETF RFC 3550]	IETF RFC 3550 (2003), <i>RTP: A Transport Protocol for Real-Time Applications</i> .
[IETF RFC 3551]	IETF RFC 3551 (2003), <i>RTP Profile for Audio and Video Conferences with Minimal Control</i> .
[IETF RFC 3556]	IETF RFC 3556 (2003), <i>Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth</i> .
[IETF RFC 5226]	IETF RFC 5226 (2008), <i>Guidelines for Writing an IANA Considerations Section in RFCs</i> .
[IETF RFC 5615]	IETF RFC 5615 (2009), <i>H.248/MEGACO Registration Procedures</i> .

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 access gateway: A type of gateway that provides a user-network interface (UNI) such as ISDN.

3.2.2 descriptor: A syntactic element of the protocol that groups related properties. For instance, the properties of a media flow on the MG can be set by the MGC by including the appropriate descriptor in a command.

3.2.3 media gateway (MG): The media gateway converts media provided in one type of network to the format required in another type of network. For example, an MG could terminate bearer channels from a switched circuit network (e.g., DS0s) and media streams from a packet network (e.g., RTP streams in an IP network). This gateway may be capable of processing audio, video and ITU-T T.120 alone or in any combination, and will be capable of full duplex media translations. The MG may also play audio/video messages and perform other IVR functions, or may perform media conferencing.

3.2.4 media gateway controller (MGC): Controls the parts of the call state that pertain to connection control for media channels in an MG.

3.2.5 multipoint control unit (MCU): An entity that controls the setup and coordination of a multi-user conference that typically includes processing of audio, video and data.

3.2.6 residential gateway: A gateway that interworks an analogue line to a packet network. A residential gateway typically contains one or two analogue lines and is located at the customer premises.

3.2.7 SCN FAS signalling gateway: This function contains the SCN signalling interface that terminates SS7, ISDN or other signalling links where the call control channel and bearer channels are collocated in the same physical span.

3.2.8 SCN NFAS signalling gateway: This function contains the SCN signalling interface that terminates SS7 or other signalling links where the call control channels are separated from bearer channels.

3.2.9 stream: Bidirectional media or control flow received/sent by a media gateway as part of a call or conference.

3.2.10 stream endpoint (SEP): Relates to a stream associated to a Termination. A SEP is identified by the tuple of *StreamID* and *TerminationID* values.

3.2.11 stream endpoint pair (SEPP): Two associated stream endpoints (SEP) of two Terminations within the same Context. The two SEPs share thus the same *StreamID* value. There is always a point-to-point topology for stream endpoint pair configurations (e.g., there is no further associated SEP on a third Termination, etc.).

3.2.12 trunk: A communication channel between two switching systems such as a DS0 on a T1 or E1 line.

3.2.13 trunking gateway: A gateway between SCN network and packet network that typically terminates a large number of digital circuits.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AAD	Average Acknowledgement Delay
AAL	ATM Adaptation Layer
ABNF	Augmented Backus-Naur Format
ADEV	Average Deviation
ALC	Automatic Level Control
ALF	Application Level Framing
AH	Authentication Header
ATM	Asynchronous Transfer Mode
BER	Basic Encoding Rules
C	Context
CAS	Channel Associated Signalling
DLSR	Delay since last SR
DNS	Domain Name System
DTMF	Dual Tone Multi-Frequency
ESP	Encapsulating Security Payload
FAS	Facility Associated Signalling
GSM	Global System for Mobile communications
GW	Gateway
IANA	Internet Assigned Numbers Authority (superseded by ICANN)
ICANN	Internet Corporation for Assigned Names and Numbers
ICV	Integrity Check Value
IEPS	International Emergency Preference Scheme



IKE	Internet Key Exchange
IP	Internet Protocol
IS	In-Service
ISUP	ISDN User Part
IVR	Interactive Voice Response
LD	Local Descriptor
LSR	Last SR timestamp
MG	Media Gateway
MGC	Media Gateway Controller
MID	Message Identifier
MTP3	Message Transfer Part 3
MTU	Maximum Transmission Unit
MWD	Maximum Waiting Delay
NDA	Non Disclosure Agreement
NFAS	Non-Facility Associated Signalling
OoS	Out-of-Service
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RD	Remote Descriptor
RR	Receiver Report
RSA	Rivest, Shamir and Adleman algorithm
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTPD	Round-Trip Propagation Delay
Rx	Receive
SC	ServiceChange
SCN	Switched Circuit Network
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SEP	Stream End Point
SEPP	Stream End Point Pair
SG	Signalling Gateway
SR	Sender Report
SS7	Signalling System No. 7
T, Term	Termination
TCP	Transmission Control Protocol

Tx	Transmit
UDP	User Datagram Protocol
VBD	Voice Band Data
VMG	Virtual Media Gateway

5 Conventions

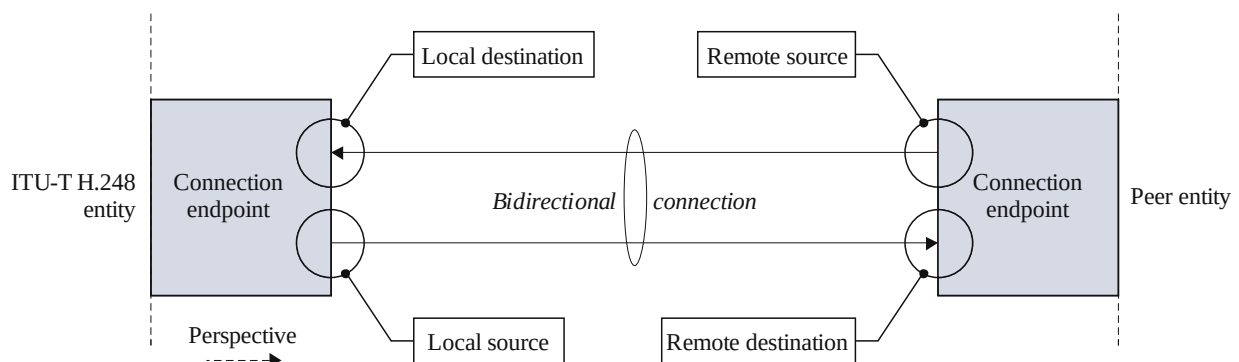
5.1 Key words to indicate requirement levels

In this Recommendation, "shall" refers to a mandatory requirement, while "should" refers to a suggested but optional feature or procedure. The term "may" refers to an optional course of action without expressing a preference.

5.2 Connection endpoint naming conventions

5.2.1 Generic concept

There are basically four connection endpoints (concerning sources and sinks of traffic) in case of a bidirectional connection. ITU-T H.248 uses a specific terminology for naming these endpoints from the perspective of an ITU-T H.248 entity. Figure 1 provides a conceptual overview. The naming scheme is generic because it is technology-independent.



H.248.1 v3(13)_F01

Figure 1 – Connection endpoint naming conventions – Generic concept

The geographical locations of the endpoints are termed as 'local' (also known as 'near-end') and as 'remote' (also known as 'far-end'). The traffic directions are termed as 'source' (also known as 'egress', 'outgoing', 'outbound', 'transmit', or 'Tx') and as 'destination' (also known as 'ingress', 'incoming', 'inbound', 'receive', or 'Rx').

5.2.2 Scope of Local and Remote Descriptors

The Local and Remote Descriptors are defined in clause 7.1.8. Both descriptors have different connection endpoints in scope (see Figure 2). The Local Descriptor (LD) is related to the "local destination" endpoint, the Remote Descriptor (RD) is related to the "remote destination" endpoint. The media gateway is primarily responsible for the resources and configuration of the "local source" endpoint (NOTE – There are a few protocol elements which allow the MGC to control certain endpoint aspects). The "remote source" endpoint is completely out of scope of ITU-T H.248 entities.

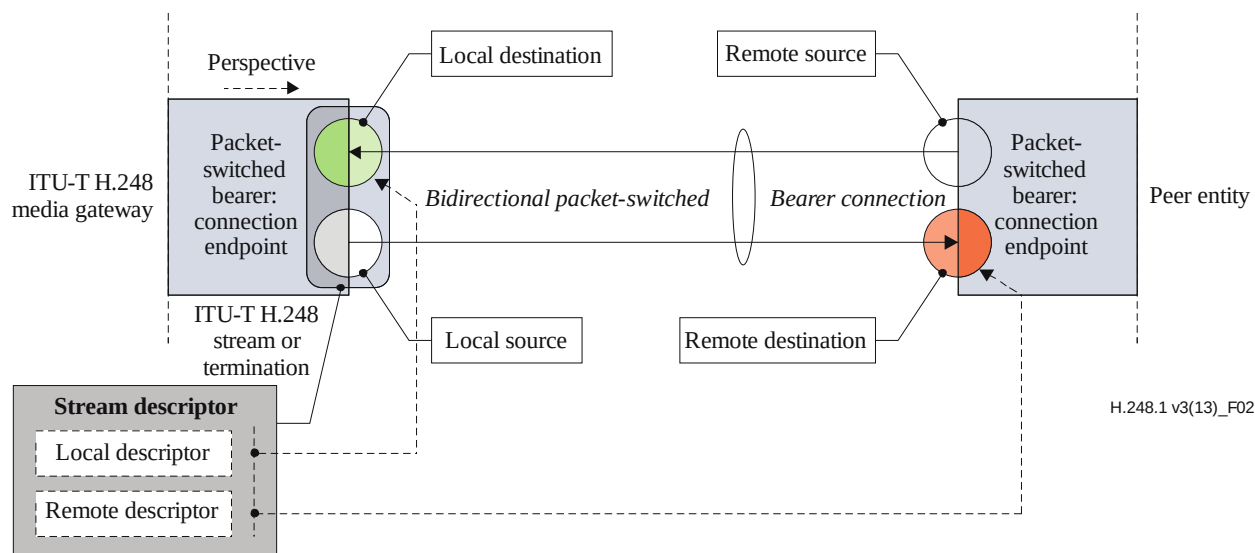


Figure 2 – Connection endpoint naming conventions – Scope of Local and Remote Descriptors

Figure 2 underlines that just the traffic sink endpoints of ITU-T H.248 streams are controlled via the Local and Remote Descriptors.

5.2.3 Concrete "connection" types

An example shall illustrate the application of the naming conventions for a specific combination of a bearer technology and protocol layer.

5.2.3.1 Example "IP connection"

Figure 3 shows an IP connection, e.g., behind an ITU-T H.248 IP stream or termination, or below an IP-based ITU-T H.248 control association.

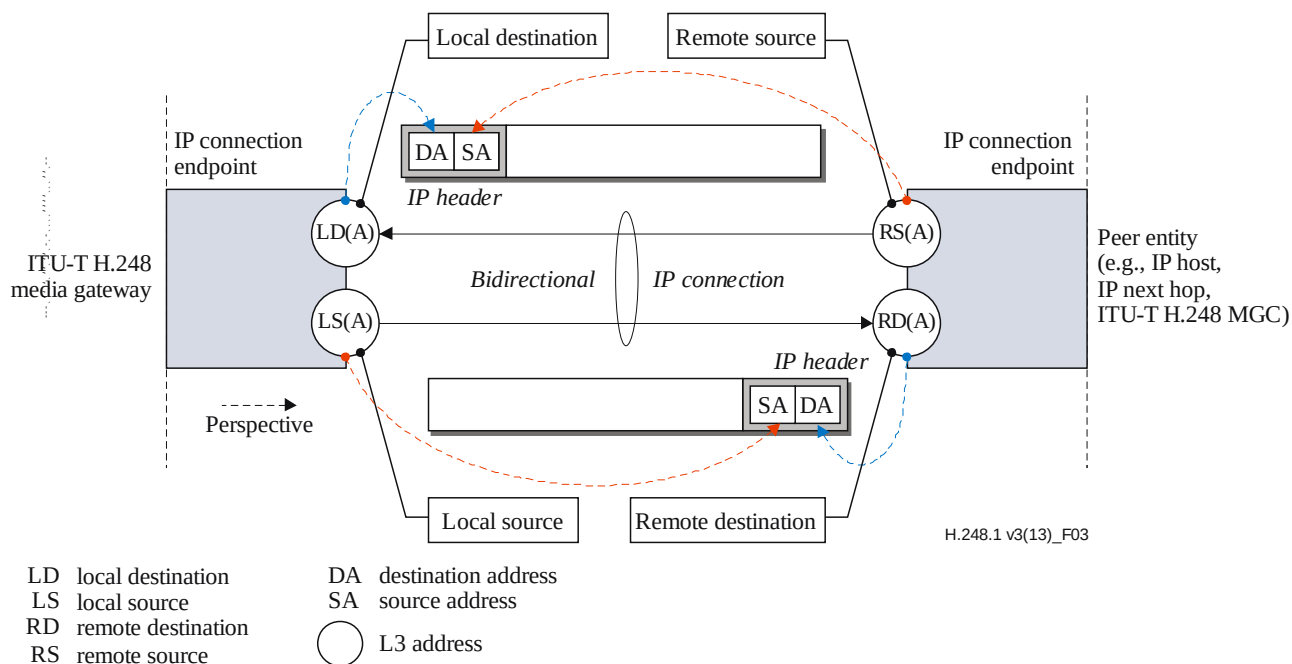


Figure 3 – Connection endpoint naming conventions – IP connection

The two local or remote connection endpoints may be assigned to the same or different IP interfaces. They are then so called symmetrical or asymmetrical (connection) addresses.

6 Connection model

The connection model for the protocol describes the logical entities, or objects, within the Media Gateway that can be controlled by the Media Gateway Controller. The main abstractions used in the connection model are terminations and contexts.

A *termination* sources and/or sinks one or more streams. In a multimedia conference, a termination can be multimedia and sources or sinks multiple media streams. The media stream parameters, as well as bearer parameters are encapsulated within the termination.

A *context* is an association between a collection of terminations. There is a special type of context, the *NULL* Context, which contains all terminations that are not present in any other context and therefore not associated to any other termination. For instance, in a decomposed access gateway, all idle lines are represented by terminations in the NULL Context.

Following is a graphical depiction of these concepts. The diagram of Figure 4 gives several examples and is not meant to be an all-inclusive illustration. The asterisk box in each of the contexts represents the logical association of terminations implied by the context.

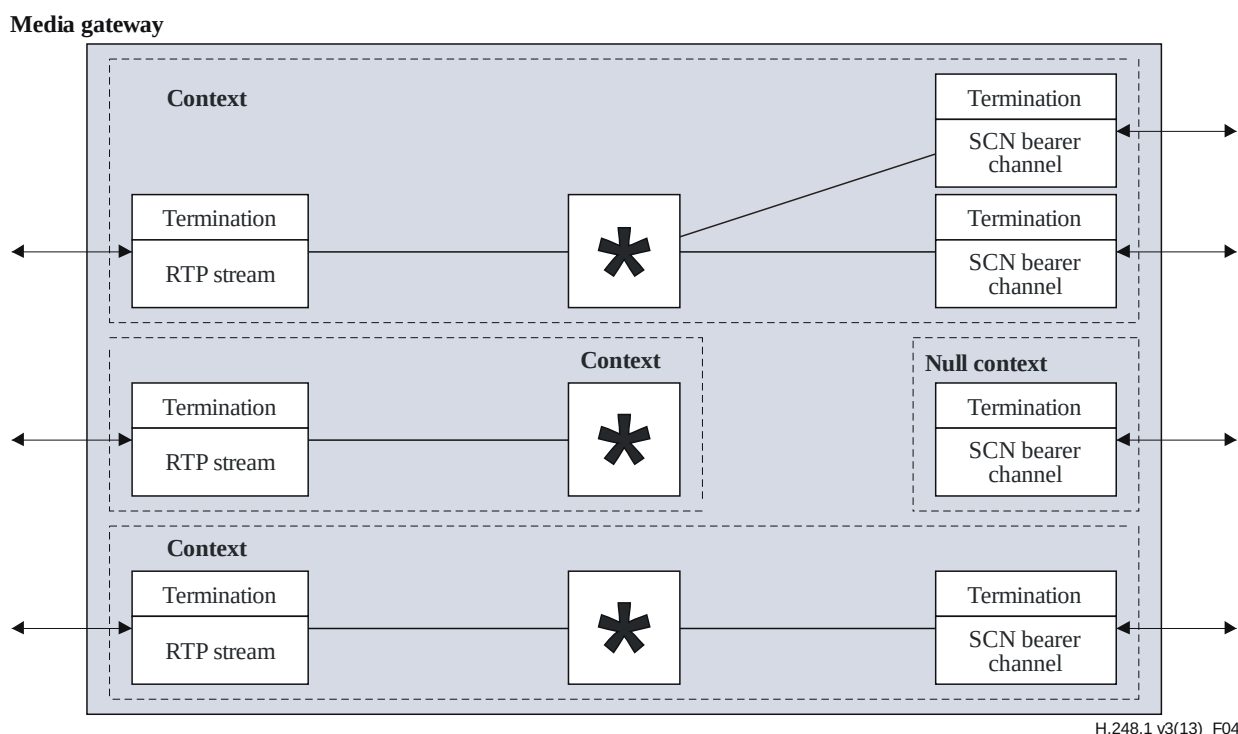


Figure 4 – Example of ITU-T H.248.1 connection model

The example in Figure 5 shows an example of one way to accomplish a call-waiting scenario in a decomposed access gateway, illustrating the relocation of a termination between contexts. Terminations T1 and T2 belong to Context C1 in a two-way audio call. A second audio call is waiting for T1 from Termination T3. T3 is alone in Context C2. T1 accepts the call from T3, placing T2 on hold. This action results in T1 moving into Context C2, as shown in Figure 6.

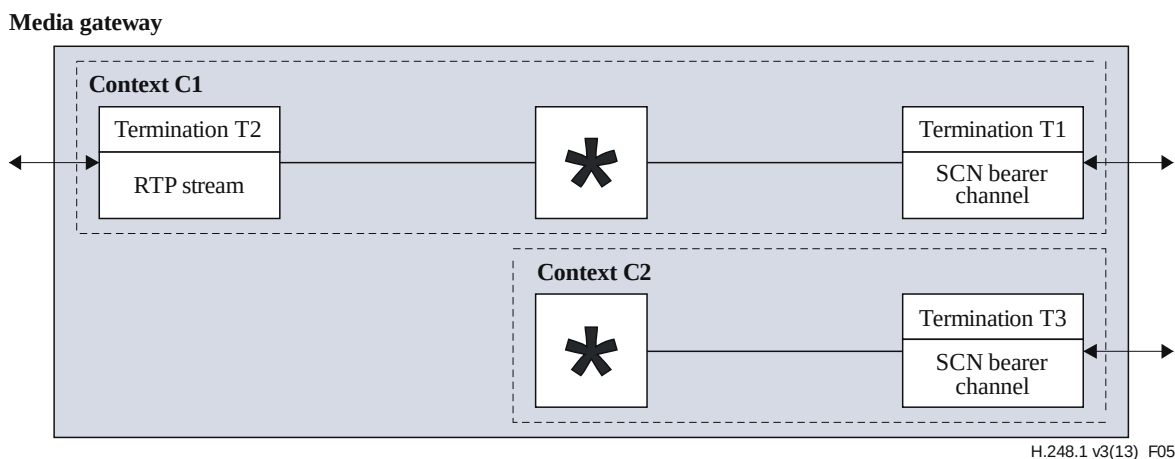


Figure 5 – Example call waiting scenario/alerting applied to T1

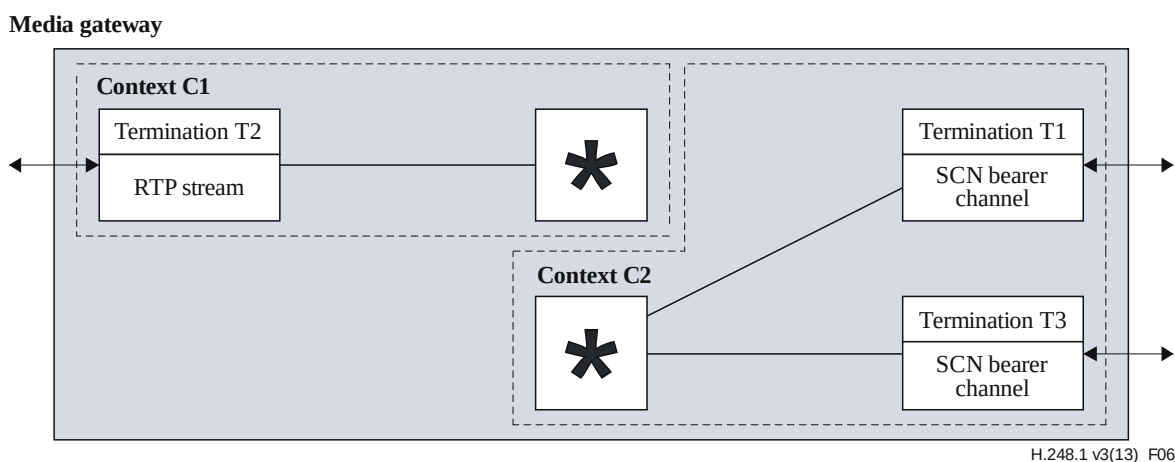


Figure 6 – Example call waiting scenario/answer by T1

6.1 Contexts

A context is an association between a number of terminations. The context describes the topology (who hears/sees whom) and the media mixing and/or switching parameters if more than two terminations are involved in the association.

There is a special context called the *NULL* Context. It contains terminations that are not present in any other context and therefore not associated to any other termination. Terminations in the *NULL* Context can have their parameters examined or modified, and may have events detected on them.

In general, an Add Command is used to add terminations to contexts. If the MGC does not specify an existing context to which the termination is to be added, the MG creates a new context. A termination may be removed from a context with a Subtract Command, and a termination may be moved from one context to another with a Move Command. A termination shall exist in only one context at a time.

The maximum number of terminations in a context is an MG property. Media gateways that offer only point-to-point connectivity might allow at most two terminations per context. Media gateways that support multipoint conferences might allow three or more terminations per context.

6.1.1 Context attributes and descriptors

The attributes of contexts are:

- ContextID.
- The Topology Descriptor (who hears/sees whom).
The topology of a context describes the flow of media between the terminations within a context. In contrast, the Mode Property of a termination ("SendOnly"/"RecvOnly"/...) describes the flow of the media at the egress/ingress of the media gateway.
- The priority indicator is used for a context in order to provide the MG with information about a certain precedence handling for a context. The MGC can also use the priority indicator to control autonomously the traffic precedence in the MG in a smooth way in certain situations (e.g., restart), when a lot of contexts must be handled simultaneously. Priority 0 is the lowest priority and a priority of 15 is the highest priority.
- An indicator for an emergency call is also provided to allow a preference handling in the MG.
- An indicator for an IEPS call is provided to allow the features and techniques of e.g., [ITU-T E.106] and [ITU-T E.107] to be achieved. See [b-ITU-T H-Sup.9] for more information.
- A ContextAttribute Descriptor that enables extra context attributes to be defined by using the packages extension mechanism (see clause 7.1.19).

In general, if a context attribute is completely omitted from an ITU-T H.248 action, the attribute of the corresponding context retains its prior value.

6.1.2 Creating, deleting and modifying contexts

The protocol can be used to (implicitly) create contexts and modify the parameter values of existing contexts. The protocol has commands to add terminations to contexts, subtract them from contexts, and to move terminations between contexts. Contexts are deleted implicitly when the last remaining termination is subtracted or moved out.

6.2 Terminations

A termination is a logical entity on an MG that sources and/or sinks media and/or control streams. A termination is described by a number of characterizing properties, which are grouped in a set of descriptors that are included in commands. Terminations have unique identities (TerminationIDs), assigned by the MG at the time of their creation.

Terminations representing physical entities have a semi-permanent existence. For example, a termination representing a TDM channel might exist for as long as it is provisioned in the gateway. Terminations representing ephemeral information flows, such as RTP flows, would usually exist only for the duration of their use.

Ephemeral terminations are created by means of an Add Command. They are destroyed by means of a Subtract Command. In contrast, when a physical termination is Added to or Subtracted from a context, it is taken from or to the NULL Context, respectively.

Terminations may have signals applied to them (see clause 7.1.11). Terminations may be programmed to detect events, the occurrence of which can trigger notification messages to the MGC, or action by the MG. Statistics may be accumulated on a termination. Statistics are reported to the MGC upon request (by means of the AuditValue Command, see clause 7.2.5) and when the termination ceases to exist or is returned to the NULL Context due to a Subtract Command.

Multimedia gateways may process multiplexed media streams. For example, [b-ITU-T H.221] describes a frame structure for multiple media streams multiplexed on a number of digital 64 kbit/s

channels. Such a case is handled in the connection model in the following way. For every bearer channel that carries part of the multiplexed streams, there is a physical or ephemeral "bearer termination". The bearer terminations that source/sink the digital channels are connected to a separate termination called the "multiplexing termination". The multiplexing termination is an ephemeral termination representing a frame-oriented session. The Multiplex Descriptor for this termination describes the multiplex used (e.g., [b-ITU-T H.221] for an [b-ITU-T H.320] session) and indicates the order in which the contained digital channels are assembled into a frame.

Multiplexing terminations may be cascades (e.g., [b-ITU-T H.226] multiplex of digital channels feeding into a [b-ITU-T H.223] multiplex supporting an [b-ITU-T H.324] session).

The individual media streams carried in the session are described by Stream Descriptors on the multiplexing termination. These media streams can be associated with streams sourced/sunk by terminations in the context other than the bearer terminations supporting the multiplexing termination. Each bearer termination supports only a single data stream. These data streams do not appear explicitly as streams on the multiplexing termination and they are hidden from the rest of the context.

Figures 7, 8, and 9 illustrate typical applications of the multiplexing termination and Multiplex Descriptor.

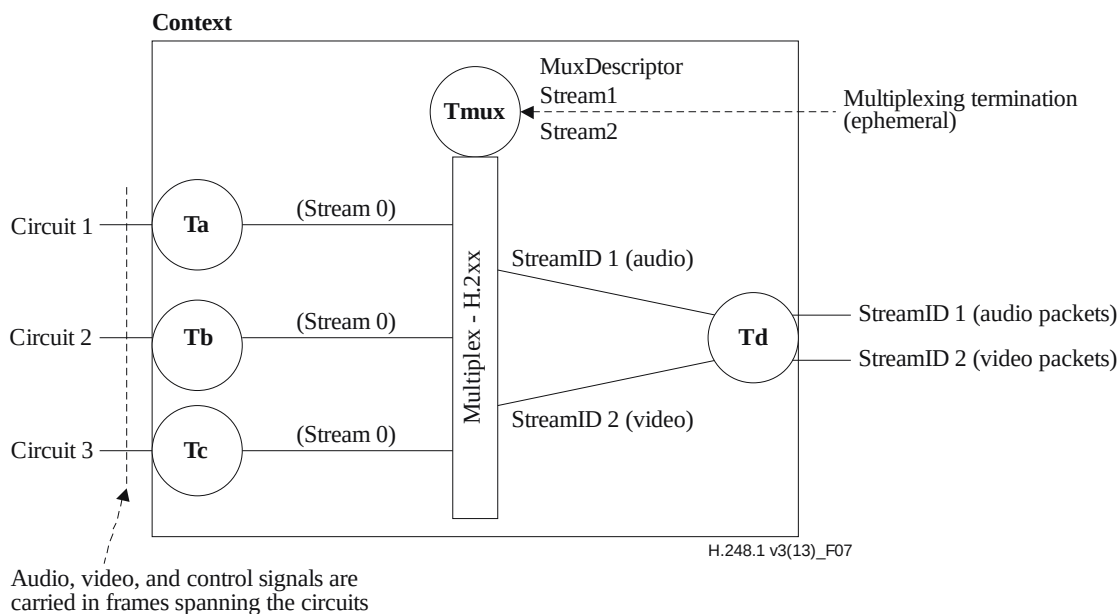


Figure 7 – Multiplexed termination scenario – Circuit-to-packet

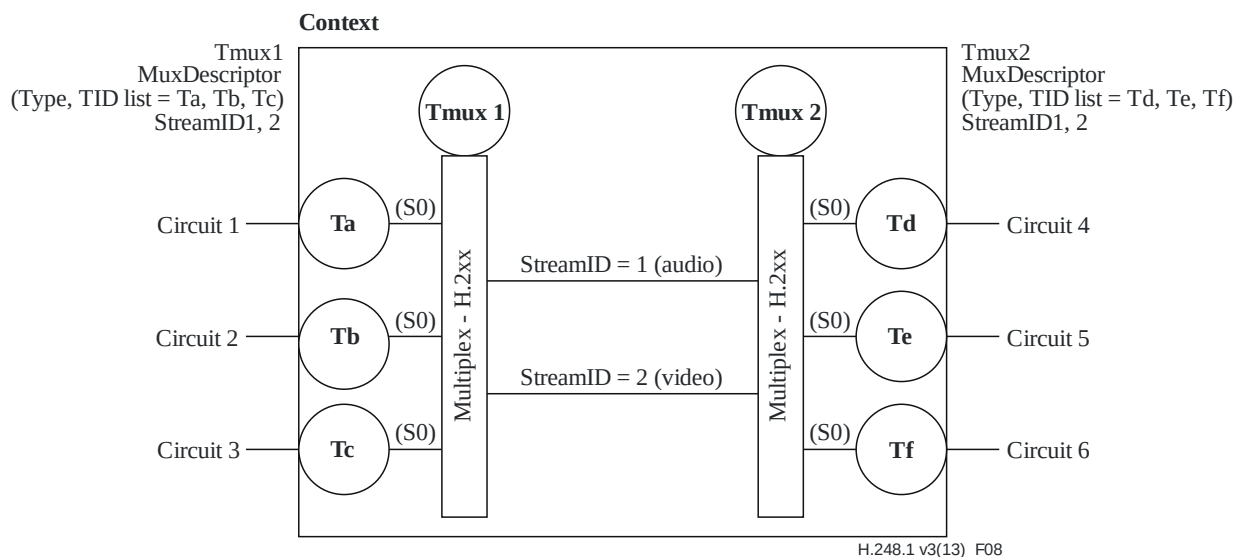


Figure 8 – Multiplexed termination scenario – Circuit-to-circuit

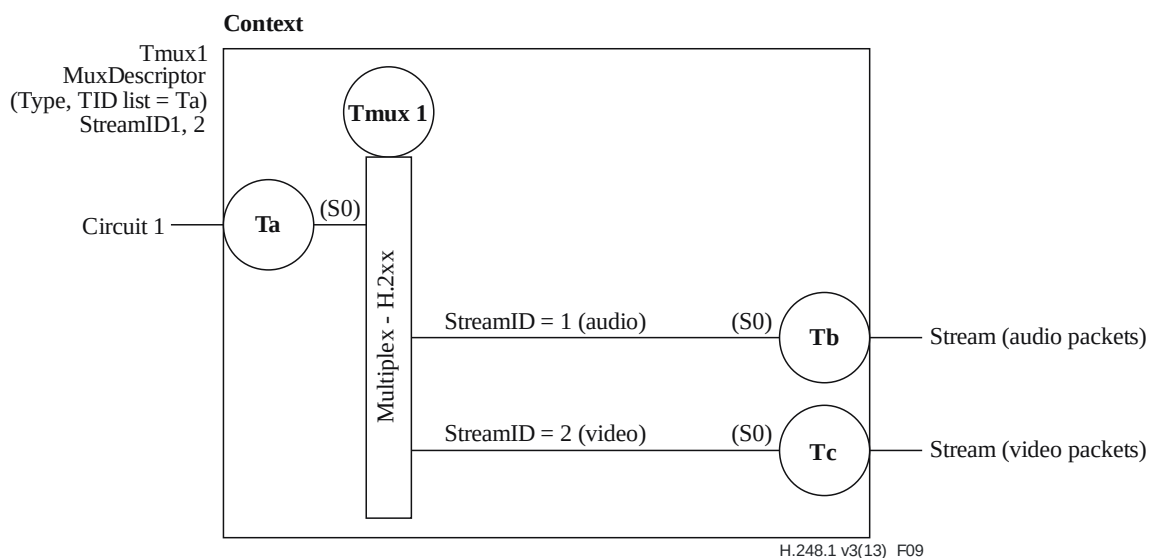


Figure 9 – Multiplexed termination scenario – Single-to-multiple terminations

Unlike the multiplexing terminations described in the previous paragraph the multiplexed bearer termination, which represent multiplexed bearers such as ATM AAL Type 2 bearers, carry no media streams. They are present strictly for the purpose of modeling the creation and destruction of the actual bearer. When a new multiplexed bearer is to be created, an ephemeral termination is created in a context established for this purpose. When the termination is subtracted, the multiplexed bearer is destroyed.

6.2.1 Termination dynamics

The protocol can be used to create new terminations and to modify property values of existing terminations. These modifications include the possibility of adding or removing events and/or signals. The termination properties, and events and signals are described in the ensuing subclauses. An MGC can only release/modify terminations and the resources that the termination represents which are in the NULL Context or which have been previously seized via, e.g., the Add Command.

6.2.2 TerminationIDs

Terminations are referenced by a TerminationID, which is an arbitrary schema chosen by the MG.

TerminationIDs of physical terminations are provisioned in the Media Gateway. The TerminationIDs may be chosen to have structure. For instance, a TerminationID may consist of trunk group and a trunk within the group.

A wildcarding mechanism using two types of wildcards can be used with TerminationIDs. The two wildcards are ALL and CHOOSE. The former is used to address multiple terminations at once, while the latter is used to indicate to a media gateway that it must select a termination satisfying the partially specified TerminationID. This allows, for instance, that an MGC instructs an MG to choose a circuit within a trunk group.

TerminationIDs can also be specified in a list. The use of TerminationIDList is recommended for cases where a hierarchical TerminationID structure is not possible and it is not desired to send individual commands for each TerminationID. Both Root and non-Root TerminationIDs should not be contained in a single TerminationIDList.

6.2.3 Packages

Different types of gateways may implement terminations that have widely differing characteristics. Variations in terminations are accommodated in the protocol by allowing terminations to have optional properties, events, signals and statistics implemented by MGs.

In order to achieve MG/MGC interoperability, such options are grouped into packages, and typically a termination realizes a set of such packages. More information on definition of packages can be found in clause 12. An MGC can audit a termination to determine which packages it realizes.

Properties, events, signals and statistics defined in packages, as well as parameters to them, are referenced by identifiers (IDs). Identifiers are scoped. For each package, PropertyIDs, EventIDs, SignalIDs, StatisticIDs and ParameterIDs have unique name spaces and the same identifier may be used in each of them. Two PropertyIDs in different packages may also have the same identifier, etc.

To support a particular package the MG must recognize all properties, signals, events and statistics defined in the package. It must also support all signal and event parameters. If the functionality behind these properties, signals, events and statistics is not implemented, the MG shall not return a syntax error or unknown ID error for any of these elements but should return error 501 "Not implemented".

The MG may support a subset of the values listed in a package for a particular property or parameter. If an unsupported value is specified by the MGC, the MG shall return error 501 "Not implemented".

When packages are extended, the properties, events, signals and statistics defined in the base package can be referred to using either the extended package name or the base package name. For example, if Package A defines Event e1, and Package B extends Package A, then B/e1 is an event for a termination implementing Package B. By definition, the MG must also implement the base package, but it is optional to publish the base package as an allowed interface. If it does publish A, then A would be reported on the Packages Descriptor in AuditValue as well as B, and Event A/e1 would be available on a termination. If the MG does not publish A, then only B/e1 would be available. If published through AuditValue, A/e1 and B/e1 are the same event.

For improved interoperability and backward compatibility, an MG may publish all packages supported by its terminations, including base packages from which extended packages are derived. An exception to this is in cases where the base packages are expressly defined as "Designed to be extended only".

In the case that the MG publishes the base PackageIDs in response to a Packages Descriptor Audit, it will also respond to a wildcarded PackageID Audit with the PackageID of the ITU-T H.248 Package Element where the element was originally defined.

In the case that the MG publishes only the extended PackageID in response to a Packages Descriptor Audit, the MG will respond to a wildcarded PackageID Audit using the extended ITU-T H.248 PackageIDs.

NOTE – Whilst a Subtract request does not explicitly request an AuditValue.req of Statistics with a wildcarded PackageID, it is implicit in the command. Similarly, an audit of a complete descriptor is equivalent to an audit with a wildcarded PackageID. Thus, the procedure described above will be used in both cases.

6.2.4 Termination properties and descriptors

Terminations have properties. The properties have unique PropertyIDs. Most properties have default values, which are explicitly defined in this protocol specification or in a package (see clause 12) or set by provisioning. If not provisioned otherwise, the properties in all descriptors except TerminationState and LocalControl default to empty/"no value" when a termination is first created or returned to the NULL Context. When a termination is first created or returned to the NULL Context, this state represents an "idle" line, trunk or other entity. The default contents of the two exceptions are described in clauses 7.1.5 and 7.1.7.

The provisioning of a property value in the MG will override any default value, be it supplied in this protocol specification or in a package. Therefore, if it is essential for the MGC to have full control over the property values of a termination, it should supply explicit values when adding the termination to a context. Alternatively, for a physical termination the MGC can determine any provisioned property values by auditing the termination while it is in the NULL Context.

There are a number of common properties for terminations and properties specific to media streams. The common properties are also called the TerminationState properties. For each media stream, there are local properties and properties of the received and transmitted flows.

Properties not included in the base protocol are defined in packages. These properties are referred to by a name consisting of the package name and a PropertyID. Most properties have default values described in the package description. Properties may be read-only or read/write. The possible values of a property may be audited, as can their current values. For properties that are read/write, the MGC can set their values. A property may be declared as "global" which has a single value shared by all terminations realizing the package. Related properties are grouped into descriptors for convenience.

When a termination is added to a context, the value of its read/write properties can be set by including the appropriate descriptors as parameters to the Add Command. Similarly, a property of a termination in a context may have its value changed by the Modify Command. Properties may also have their values changed when a termination is moved from one context to another as a result of a Move Command. In some cases, descriptors are returned as output from a command.

Setting properties on different terminations in the same context implicitly instructs the MG to perform certain functions. For example: if a [b-ITU-T G.711] codec is set on Termination A and a [b-ITU-T G.729] codec is set on Termination B, then the MG will activate a transcoding function as soon as a media flow is enabled between the two terminations (i.e., by setting the Mode Property to a state other than "Inactive" at each termination).

NOTE – To avoid unnecessary activation of MG resources, the MGC should set the Mode Property to "Inactive" for a given termination and stream until the properties to be used for that stream have been determined.

Copyright International Telecommunication Union/ITU-T

In general, if a descriptor is completely omitted from one of the aforementioned commands, the properties in that descriptor retain their prior values for the termination(s) upon which the command acts. On the other hand, if some read/write properties are omitted from a descriptor in a command (i.e., the descriptor is only partially specified), those properties will be reset to their default values for the termination(s) upon which the command acts, unless the package specifies other behaviour. For more details, see clause 7.1 dealing with the individual descriptors.

The above behaviour applies equally to signals and events and their respective parameters. The provisioning of Events Descriptors would include the RequestID to use and any parameters of the included events. Any provisioning in the MG with respect to Events Descriptors should be duplicated in the MGC to avoid error responses to Notify Commands from the MG.

The following table lists all of the possible descriptors and their use. Not all descriptors are legal as input or output parameters to every command.

Descriptor name	Description
Modem	Identifies modem type and properties when applicable. (Note)
Mux	Describes multiplex type for multimedia terminations (e.g., [b-ITU-T H.221], [b-ITU-T H.223], [ITU-T H.225.0]) and terminations forming the input mux.
Media	A list of media stream specifications (see clause 7.1.4).
TerminationState	Properties of a termination (which can be defined in packages) that are not stream specific.
Stream	A list of Remote/Local/LocalControl Descriptors for a single stream.
Local	Contains properties that specify the media flows that the MG receives from the remote entity.
Remote	Contains properties that specify the media flows that the MG sends to the remote entity.
LocalControl	Contains properties (which can be defined in packages) that are of interest between the MG and the MGC.
Events	Describes events to be detected by the MG and what to do when an event is detected.
EventBuffer	Describes events to be detected by the MG when event buffering is active.
Signals	Describes signals (see clause 7.1.11) applied to terminations.
Audit	In Audit commands, identifies which information is desired.
Packages	In AuditValue, returns a list of packages realized by the termination.
DigitMap	Defines patterns against which sequences of a specified set of events are to be matched so they can be reported as a group rather than singly.
ServiceChange	In ServiceChange, what, why ServiceChange occurred, etc.
ObservedEvents	In Notify or AuditValue, report of events observed.
Statistics	In Subtract and Audit, report of statistics kept on a termination or stream.
Topology	Specifies flow directions between terminations in a context.
ContextAttribute	Contains properties (which can be defined in packages) that affect the context as a whole.
Error	Contains an Error Code and optionally error text; it may occur in command replies and in Notify requests.
NOTE – Modem Descriptor has been deprecated in Recommendation ITU-T H.248.1 version 2 (05/2002).	

6.2.5 Root Termination

Occasionally, a command must refer to the gateway as an entity in itself, rather than a termination within it. A special TerminationID, "Root" is reserved for this purpose. Packages may be defined on Root. Root thus may have properties, events, signals and statistics. Accordingly, the Root TerminationID may appear in:

- a Modify Command – to change a property, send a signal or set an event;
- a Notify Command – to report an event;
- an AuditValue Command – to examine the values of properties and statistics implemented on Root;
- an AuditCapability Command – to determine what properties of Root are implemented;
- a ServiceChange – to declare the entire gateway in or out of service.

Any other use of the Root TerminationID is an error. Error Code 410 ("Incorrect identifier") shall be returned in these cases.

6.3 Wildcarding principles

This clause specifies the behaviour for wildcarding ContextIDs and TerminationIDs that shall be applied to all commands. In processing these commands two forms of wildcarding must be considered:

- 1) Context wildcarding;
- 2) Termination wildcarding.

For the purposes of the wildcarding procedures, a TerminationIDList consisting of more than one TerminationID is considered a wildcarded TerminationID. When executing a transaction that contains wildcarded contexts and optionally wildcarded terminations, all commands in the transaction are executed in order for a particular instance of ContextID before moving to a subsequent ContextID instance. In the case that there are multiple commands in a transaction, only when the TerminationID (wildcarded or specific) specified in the first command matches a specific instance of a ContextID are subsequent commands in the transaction executed. If a TerminationID (wildcarded or specific) of the subsequent command(s) in that transaction does not match the specific ContextID instance, then Error Code 431 ("No TerminationID matched a wildcard") is returned and processing of subsequent instances of the wildcard ContextID are stopped unless the command that generated the error is marked optional.

The execution of particular wildcard combinations is discussed below.

6.3.1 ContextID specific with TerminationID wildcarded

In the case where the ContextID is specific, when ALL is used in the TerminationID of a command, the effect is identical to repeating the command with each of the matching TerminationIDs. The use of ALL does not address the Root termination. Since each of these commands may generate a response, the size of the entire response may be large. Thus, if the wildcard matches more than one TerminationID in the context, all possible matches are attempted, with results reported for each one. If none of the terminations referenced by the wildcarded TerminationID are in the specific context, then Error Code 431 ("No TerminationID matched a wildcard") is returned. No errors are returned for individual terminations specified by the wildcarded TerminationID that are not in the specified context.

For example: Assume that a gateway has four terminations: t1/1, t1/2, t2/1 and t2/2. Assume that Context 1 has t1/1 and t2/1 in it and that Context 2 has t1/2 and t2/2 in it.

The command:

```
Context=1{Command=t1/*{Descriptor/s}}
```

returns:

```
Context=1{Command=t1/1{Descriptor/s}}
```

6.3.2 ContextID wildcarded (ALL) with TerminationID specific

In the case where the ContextID is wildcarded (i.e., ContextID = ALL) and the TerminationID is fully specified, the effect is identical to a command specifying the non-NULL context that contains the specified termination. Thus, a search must be made to find the context and only one instance of the command is executed. No errors are reported for contexts that do not contain the specified termination. If the termination is not contained in any (non-NULL) context, then Error Code 435 ("TerminationID is not in specified context") is returned, though Error Code 431 ("No TerminationID matched a wildcard") may be returned in order to maintain backward compatibility. If there are no contexts other than NULL in existence, Error Code 411 ("The transaction refers to an unknown ContextID") is returned. Use of this form of action rather than one specifying the ContextID is discouraged but may be useful, for example in correcting conflicting state between MG and MGC.

For example: Taking the above gateway configuration, the command:

```
Context=*{Command=t1/1{Descriptor/s}}
```

returns:

```
Context=1{Command=t1/1{Descriptor/s}}
```

6.3.3 ContextID wildcarded (ALL) with TerminationID wildcarded

In the case where the ContextID is wildcarded (i.e., ContextID = ALL) and the TerminationID is wildcarded, the effect is identical to repeating the command with each of the TerminationIDs matching the wildcard for each non-NULL context that contains one or more of those matching TerminationIDs. Thus if the wildcard matches more than one TerminationID in the specific instance of the wildcarded ContextID, all possible matches are attempted, with results reported for each one. No errors are reported for contexts that do not contain a termination matching the wildcarded TerminationID. No errors are returned for individual terminations specified in the wildcarded TerminationID that are not in a specific instance of the wildcarded ContextID. If there are no matches to the wildcarded ContextID and TerminationID, then Error Code 431 ("No TerminationID matched a wildcard") is returned.

For example: Taking the above gateway configuration, the command:

```
Context=*{Command=t1/*{Descriptor/s}}
```

returns:

```
Context=1{Command=t1/1{Descriptor/s}}
```

```
Context=2 {Command=t1/2{Descriptor/s}}
```

In the case that multiple commands are contained in a wildcarded TerminationID and/or wildcarded ContextID request, then if the first command does not match the first ContextID and TerminationID instance, then the subsequent command in the request will not be executed for that instance.

6.3.4 Wildcarded responses

If individual responses are not required, a wildcard response may be requested. In such a case, a single response is generated, which contains the UNION of all of the individual responses which otherwise would have been generated, with duplicate values suppressed. For instance, given Termination Ta with Properties p1=a, p2=b and Termination Tb with Properties p2=c, p3=d, a UNION response would consist of a wildcarded TerminationID and the sequence of Properties

p1=a, p2=b,c and p3=d. Wildcard responses may be particularly useful in the Audit commands. If a wildcard UNION response is used in conjunction with a wildcarded context, then a single response is sent with the UNION of all the individual terminations referenced by the TerminationID. The response would contain Context=ALL, a wildcarded TerminationID and the sequence of properties.

If an error occurs during the execution of a wildcarded request that specifies a wildcarded response, special handling is required to provide useful information about the error(s) while still maintaining a modest sized response. When a wildcarded response is requested, all instances (as specified above) of the command shall be executed even if one or more result in errors, but later commands in the transaction will not be executed (unless optional was specified). Multiple command responses shall be returned for the command that encountered the error. The first command response shall be the normal wildcard response containing the UNION of responses for those commands that succeeded. If none of them succeeded, the UNION shall be empty. Additional command responses for each TerminationID that failed shall be returned with the appropriate Error Descriptor.

For example, the command:

Context={Command=t1/{Descriptor/s}}

Response to an error:

Context={Command=t1/{Union response descriptors},
Command=t1/3{Error=errorcode}}

The encoding of the wildcarding mechanism is detailed in Annexes A and B.

7 Commands

The protocol provides commands for manipulating the logical entities of the protocol connection model, contexts and terminations. Commands provide control at the finest level of granularity supported by the protocol. For example, commands exist to add terminations to a context, modify terminations, subtract terminations from a context, and audit properties of contexts or terminations. Commands provide for complete control of the properties of contexts and terminations. This includes specifying which events a termination is to report, which signals/actions are to be applied to a termination and specifying the topology of a context (who hears/sees whom).

Most commands are for the specific use of the Media Gateway Controller as command initiator in controlling Media Gateways as command responders. The exceptions are the Notify and ServiceChange Commands: Notify is sent from Media Gateway to Media Gateway Controller, and ServiceChange may be sent by either entity. Below is an overview of the commands; they are explained in more detail in clause 7.2.

- 1) Add: The Add Command adds a termination to a context. The Add Command on the first termination in a context is used to create a context.
- 2) Modify: The Modify Command modifies the properties, events and signals of a termination.
- 3) Subtract: The Subtract Command disconnects a termination from its context and returns statistics on the termination's participation in the context. The Subtract Command on the last termination in a context deletes the context.
- 4) Move: The Move Command atomically moves a termination to another context.
- 5) AuditValue: The AuditValue Command returns the current state of properties, events, signals and statistics of terminations.
- 6) AuditCapability: The AuditCapability Command returns all the possible values for termination properties, events and signals allowed by the Media Gateway.
- 7) Notify: The Notify Command allows the Media Gateway to inform the Media Gateway Controller of the occurrence of events in the Media Gateway.



- 8) **ServiceChange:** The ServiceChange Command allows the Media Gateway to notify the Media Gateway Controller that a termination or group of terminations is about to be taken out of service or has just been returned to service. ServiceChange is also used by the MG to announce its availability to an MGC (registration), and to notify the MGC of impending or completed restart of the MG. The MGC may announce a handover to the MG by sending it a ServiceChange Command. The MGC may also use ServiceChange to instruct the MG to take a termination or group of terminations in or out of service.

These commands are detailed in clauses 7.2.1 through 7.2.8.

7.1 Descriptors

The parameters to a command are termed descriptors. A descriptor consists of a name and a list of items. Some items may have values. Many commands share common descriptors. This subclause enumerates these descriptors. Descriptors may be returned as output from a command. Descriptors shall be returned in the order that they were received. In any such return of descriptor contents, an empty descriptor is represented by its name unaccompanied by any list. Parameters and parameter usage specific to a given command type are described in the subclause that describes the command.

7.1.1 Specifying parameters

Command parameters are structured into a number of descriptors. In general, the text format of descriptors is `DescriptorName=<someID>{parm=value, parm=value...}`.

Parameters may be fully specified, overspecified or underspecified:

- 1) Fully specified parameters have a single, unambiguous value that the command initiator is instructing the command responder to use for the specified parameter.
- 2) Underspecified parameters, using the CHOOSE value, allow the command responder to choose any value it can support.
- 3) Overspecified parameters have a list of potential values. The list order specifies the command initiator's order of preference of selection. The command responder chooses one value from the offered list and returns that value to the command initiator.

If a required descriptor other than the Audit Descriptor is unspecified (i.e., entirely absent) from a command, the previous values set in that descriptor for that termination, if any, are retained. In commands other than Subtract, a missing Audit Descriptor is equivalent to an empty Audit Descriptor. The behaviour of the MG with respect to unspecified parameters within a descriptor varies with the descriptor concerned, as indicated in succeeding subclauses. Whenever a parameter is underspecified or overspecified, the descriptor containing the value chosen by the responder is included as output from the command.

Each command specifies the TerminationID the command operates on. This TerminationID may be "wildcarded". When the TerminationID of a command is wildcarded, the effect shall be as if the command was repeated with each of the TerminationIDs matched.

7.1.2 Modem Descriptor

The Modem Descriptor specifies the modem type and parameters, if any, required for use in, e.g., [b-ITU-T H.324] and text conversation. The descriptor includes the following modem types: ITU-T V.18, ITU-T V.22, ITU-T V.22 *bis*, ITU-T V.32, ITU-T V.32 *bis*, ITU-T V.34, ITU-T V.90, ITU-T V.91, Synchronous ISDN, and allows for extensions. By default, no Modem Descriptor is present in a termination.

Use of the Modem Descriptor is deprecated in Recommendation ITU-T H.248.1 Version 2 (05/2002) and subsequent versions. This means that the Modem Descriptor shall not be included as part of transmitted content and, if received, shall either be ignored or processed at the option of the

implementation. Modem type is to be specified as an attribute of data streams in the Local Descriptor and the Remote Descriptor.

7.1.3 Multiplex Descriptor

In multimedia calls, a number of media streams are carried on a (possibly different) number of bearers. The Multiplex Descriptor associates the media and the bearers. The descriptor includes the multiplex type:

- H.221;
- H.223;
- H.226;
- V.76;
- $N \times 64K$;
- possible extensions,

and a set of TerminationIDs representing the multiplexed bearers, in order. For example:

$$\text{Mux} = \text{H.221}\{\text{MyT3}/1/2, \text{MyT3}/2/13, \text{MyT3}/3/6, \text{MyT3}/21/22\}$$

The $N \times 64K$ multiplex type implements the $N \times 64K$ service (e.g., as defined by [ITU-T Q.931] Information Transfer Rate or [ITU-T Q.763] Transmission Medium Requirement). On the context side, it supports a single stream of wideband data. When a bearer termination is added implicitly to a context as a result of the creation of an $N \times 64K$ multiplexing termination, the Stream Descriptor for the bearer termination shall take on the same values as the Stream Descriptor defined for the multiplex termination, except that the bearer termination bandwidth shall be 64 kbit/s.

7.1.4 Media Descriptor

The Media Descriptor specifies the parameters for all the media streams. These parameters are structured into two descriptors: a TerminationState Descriptor, which specifies the properties of a termination that are not stream dependent, and one or more Stream Descriptors each of which describes a single media stream.

A stream is identified by a StreamID. The StreamID shall be in the range of 1 to 65535. The StreamID is used to link the streams in a context that belong together. Multiple streams exiting a termination shall be synchronized with each other. Within the Stream Descriptor, there are up to four subsidiary descriptors: LocalControl, Local, Remote and Statistics. The relationship between these descriptors is thus:

```
Media Descriptor
  TerminationState Descriptor
  Stream Descriptor
    LocalControl Descriptor
    Local Descriptor
    Remote Descriptor
    Statistics Descriptor
```

As a convenience, the LocalControl, Local, Remote or Statistics Descriptors may be included in the Media Descriptor without an enclosing Stream Descriptor. In this case, the StreamID is assumed to be one.

7.1.5 TerminationState Descriptor

The TerminationState Descriptor contains the ServiceStates Property, the EventBufferControl Property and properties of a termination (defined in packages) that are not stream specific.

The ServiceStates Property describes the overall state of the termination (not stream specific). A termination can be in one of the following states: "Test", "OutOfService", or "InService". The "Test" state indicates that the termination is being tested. The state "OutOfService" indicates that the termination cannot be used for traffic. The state "InService" indicates that a termination can be used or is being used for normal traffic. "InService" is the default state.

Values assigned to properties may be simple values (integer/string/enumeration) or may be underspecified, where more than one value is supplied and the MG may make a choice:

- Alternative values: multiple values in a list, one of which must be selected;
- Ranges: minimum and maximum values, a value between min and max must be selected, boundary values included;
- Greater Than/Less Than: value must be greater/less than specified value;
- CHOOSE Wildcard: the MG chooses from the allowed values for the property.

The EventBufferControl Property specifies whether events are buffered following detection of an event in the Events Descriptor, or processed immediately. See 7.1.9 for details.

7.1.5.1 TerminationState properties

7.1.5.1.1 ServiceStates

Property name:	ServiceStates
Description:	The value of this property indicates the current service state of the termination.
Type:	Enumeration
Possible values:	InService: The termination is in-service and functioning normally. OutOfService: The termination is out-of-service and not available for traffic. Test: The termination is undergoing testing.
Default:	InService
Defined in:	TerminationState
Characteristics:	Read/Write

7.1.5.1.2 EventBufferControl

Property name:	EventBufferControl
Description:	Specifies whether events are buffered following detection of an event in the Events Descriptor, or processed immediately. See clause 7.1.9.
Type:	Enumeration
Possible values:	LockStep: Events are buffered and processed per clause 7.1.9. OFF: Events are processed immediately.
Default:	OFF
Defined in:	TerminationState
Characteristics:	Read/Write

7.1.6 Stream Descriptor

A Stream Descriptor specifies the parameters of a single bidirectional stream. These parameters are structured into three descriptors: one that contains termination properties specific to a stream and one each for local and remote flows. The Stream Descriptor includes a StreamID which identifies the stream. Streams are created by specifying a new StreamID on one of the terminations in a context. A stream is deleted by setting empty Local and Remote Descriptors for the stream with ReserveGroup and ReserveValue in LocalControl set to "False" on all terminations in the context that previously supported that stream.

StreamIDs are of local significance between MGC and MG and they are assigned by the MGC. Within a context, StreamID is a means by which to indicate which media flows are interconnected: streams with the same StreamID are connected.

If a termination is moved from one context to another, the effect on the context to which the termination is moved is the same as in the case that a new termination were added with the same StreamIDs as the moved termination.

7.1.7 LocalControl Descriptor

a) Overview

The LocalControl Descriptor contains the Mode Property, the ReserveGroup and ReserveValue Properties and properties of a termination (defined in packages) that are stream specific, and are of interest between the MG and the MGC. Values of properties may be specified as in clause 7.1.1.

b) Mode Property (for directionality control)

The allowed values for the Mode Property are "SendOnly", "RecvOnly", "SendRecv", "Inactive" and "LoopBack". "SendOnly", "RecvOnly" and "LoopBack" are with respect to the exterior of the context, so that, for example, a stream set to mode = "SendOnly" does not pass received media into the context. When a stream is set to "LoopBack" on a termination, media received (Local Descriptor) on the termination will be looped back to the sending side (Remote Descriptor) of the termination and no media is passed between that termination and other terminations in the context. The looped back media shall be sent according to the Remote Descriptor. The default value for the Mode Property is "Inactive". Signals and events are not affected by the Mode Property. Statistics may or may not be affected by the Mode property depending on the semantic of the statistic. For example: if the Mode was set to "SendOnly" and the Octet Received Statistic was set, then the statistic would not be affected. If the Octet Sent Statistic was set, then it would be affected by the mode. The LocalControl Mode Property takes precedence over any mode specified in the Local and Remote Descriptors. However, duplication and use of mode information in the SDP should be avoided. Due to the default of LocalControl Mode Property being "Inactive", if mode information was added to the Local and Remote Descriptor SDP and the LocalControl Mode was not explicitly sent, the effective mode would still be inactive.

c) Reserve Properties (for resource handling control)

The boolean-valued Reserve Properties, ReserveValue and ReserveGroup, of a termination indicate what the MG is expected to do when it receives a Local and/or Remote Descriptor.

If the value of a Reserve Property is "True", the MG shall reserve resources for all alternatives specified in the Local and/or Remote Descriptors for which it currently has resources available. It shall respond with the alternatives for which it reserves resources. If it cannot support any of the alternatives, it shall respond with a reply to the MGC that contains empty Local and/or Remote Descriptors. If media begins to flow while more than a single alternative is reserved, media packets may be sent or received on any of the alternatives and must be processed, although only a single alternative may be active at any given time.

If the value of a Reserve Property is False, the MG shall choose one of the alternatives specified in the Local Descriptor (if present) and one of the alternatives specified in the Remote Descriptor (if present). If the MG has not yet reserved resources to support the selected alternative, it shall reserve the resources. If, on the other hand, it already reserved resources for the termination addressed (because of a prior exchange with ReserveValue and/or ReserveGroup equal to "True"), it shall release any excess resources it reserved previously. Finally, the MG shall send a reply to the MGC containing the alternatives for the Local and/or Remote Descriptor that it selected. If the MG does not have sufficient resources to support any of the alternatives specified, it shall respond with Error Code 510 ("Insufficient resources").

The default value of ReserveValue and ReserveGroup is "False". More information on the use of the two Reserve Properties is provided in clause 7.1.8.

d) Descriptor usage

A new setting of the LocalControl Descriptor completely replaces the previous setting of that descriptor in the MG. Thus, to retain information from the previous setting, the MGC must include that information in the new setting. If the MGC wishes to delete some information from the existing descriptor, it merely resends the descriptor (in a Modify Command) with the unwanted information stripped out.

NOTE – The Mode Property is also known as "StreamMode" in the encoding definitions in Annexes A and B. These terms are interchangeable within ITU-T H.248.

7.1.7.1 LocalControl Properties

7.1.7.1.1 StreamMode

Property name: StreamMode

Description: The value of this property indicates the current service state of the termination.

Type: Enumeration

Possible values:

- Inactive: The termination does not pass any media for the stream.
- SendOnly: The termination passes media for the stream from the interior to the exterior of the context.
- RecvOnly: The termination passes media for the stream from the exterior to the interior of the context.
- SendRecv: The termination passes media for the stream both into and out of the context.
- LoopBack: The termination loops received media for the stream back to the sender.

NOTE – The term "media" underlines that the StreamMode settings only affect the "media flow" component(s) of the ITU-T H.248 Stream, but not potential "control flow" component(s) (like, e.g., in case of an ITU-T H.248 Stream, carrying a RTP session with an RTP media flow and RTCP control component). It may be noted that the separation of media and control components is consistent with the SDP understanding of "media streams", see [b-IETF RFC 4566].

Default: Inactive

Defined in: LocalControl

Characteristics: Read/Write

7.1.7.1.2 ReserveGroup

Property name: ReserveGroup

Description: Specifies whether the MG should reserve the resources to support a single media group or as many media groups as it can, as they are defined in the Local and Remote Descriptors. See clause 7.1.8.

NOTE – The term "media group" designates the content of the ASN.1 production "PropertyGroup" (see Annex A) in the binary encoding, or an individual SDP session description in the text encoding.

Type: Boolean

Possible values: True: The MG is to reserve all possible media groups indicated in the Local and Remote Descriptors.

False: The MG is to reserve a single media group from each of the Local and the Remote Descriptors according to the policies described in clause 7.1.8.

Default: False

Defined in: LocalControl

Characteristics: Read/Write

7.1.7.1.3 ReserveValue

Property name: ReserveValue

Description: Specifies whether the MG should reserve the resources to support a single set of property values (e.g., a single codec and its associated attributes) or as many such sets as it can, as they are defined in the Local and Remote Descriptors. See clause 7.1.8.

Type: Boolean

Possible values: True: The MG is to reserve resources to serve as many as possible of the sets of property values indicated in the selected media group (if ReserveGroup is False) or in each media group (if ReserveGroup is True) in the Local and Remote Descriptors.

False: The MG is to reserve a single set of property values from those indicated in the selected media group (if ReserveGroup is False) or in each media group (if ReserveGroup is True) in the Local and Remote Descriptors.

NOTE – A single set of property values may consist of a single media type (e.g., audio or video) related media format complemented by a list of supplementary media formats. Supplementary media formats are for example:

- Comfort noise ([b-IETF RFC 3389]),
- RTP Payload for DTMF digits, telephony tones and telephony signals ([b-IETF RFC 4733]),
- Voiceband data (VBD) services (according to [b-ITU-T V.152]),
- or other auxiliary media (e.g., inband signalling) associated to the main media flow.

Default: False
Defined in: LocalControl
Characteristics: Read/Write

7.1.8 Local and Remote Descriptors

7.1.8.1 Syntax for Local Descriptors and Remote Descriptors

The MGC uses Local and Remote Descriptors to reserve and commit MG resources for media decoding and encoding for the given stream(s) and termination to which they apply. The MG includes these descriptors in its response to indicate what it is actually prepared to support. The MG shall include additional properties and their values in its response if these properties are mandatory yet not present in the requests made by the MGC (e.g., by specifying detailed video encoding parameters where the MGC only specified the payload type).

To avoid ambiguity when requesting the MG to reserve and commit resources, the MGC should supply as much information as needed when using underspecification (i.e., CHOOSE) so that the MG can make an unambiguous selection. For example when using CHOOSE without specifying the required application type (e.g., "media name" in case of SDP encoding), further information may be needed (e.g., attribute lines in case of SDP encoding).

Local refers to the media received by the MG and Remote refers to the media sent by the MG.

7.1.8.1.1 Specific syntax for ITU-T H.248 text encoding

7.1.8.1.1.1 Rules

When text encoding the protocol, the descriptors consist of session descriptions as defined in SDP ([IETF RFC 2327]). In session descriptions sent from the MG to the MGC, the SDP must comply with [IETF RFC 2327]. In session descriptions sent from the MGC to the MG, the following exceptions to the syntax of [IETF RFC 2327] are allowed:

- the "s = ", "t = " and "o = " lines are optional;
- the use of CHOOSE is allowed in place of a single parameter value; and
- the use of alternatives is allowed in place of a single parameter value.

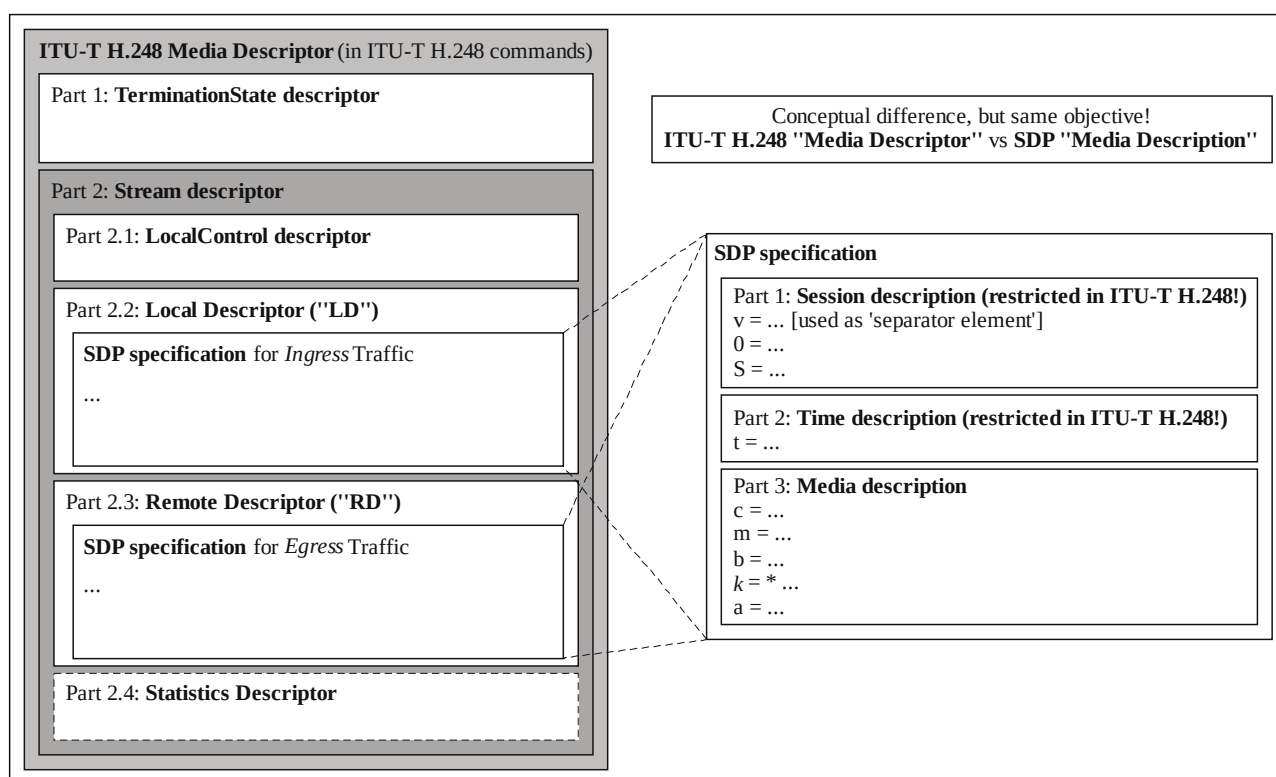
A Stream Descriptor specifies a single bidirectional media stream, a session description consists of a session-level section followed by a media-level section and so a single session description must not include more than one media description ("m = " line). A session description should not contain duplicate SDP lines on session-level and media-level. If duplication occurs the media level line takes precedence.

A Stream Descriptor may contain additional session descriptions as alternatives. Each media stream for a termination must appear in distinct Stream Descriptors. When multiple session descriptions are provided in one descriptor, the "v = " lines are required as delimiters; otherwise they are optional in session descriptions sent to the MG. Implementations shall accept session descriptions that are fully conformant to [IETF RFC 2327] according to the above restrictions.

If an MG does not understand the SDP provided, rather than ignoring that piece of SDP, an appropriate error response shall be provided.

7.1.8.1.1.2 Summary – ITU-T H.248 usage of SDP

Figure 10 provides an overview of the structure of the ITU-T H.248 Media Descriptor. SDP is used, within the ITU-T H.248 Stream Descriptor, in the Local Descriptor (LD) and in the Remote Descriptor (RD). There are thus separate SDP specifications for ingress traffic (provided by the ITU-T H.248 LD) and egress traffic (provided by the ITU-T H.248 RD). The (SDP) media description within that SDP block is reflected in the LD and RD and determines the ITU-T H.248 media gateway behaviour.



NOTE – An ITU-T H.248 profile description (see sections 6.15 and 6.16 in Appendix III) may also provide information regarding the support of SDP lines.

H.248.1 v3(13)_F10

Figure 10 – Overview Structure of ITU-T H.248 Media Descriptor (ITU-T H.248 "Media Descriptor" vs SDP "Media Description")

7.1.8.1.2 Specific syntax for ITU-T H.248 binary encoding

When binary encoding the protocol, the descriptor consists of groups of properties (tag-value pairs) as specified in Annex C. Each such group may contain the parameters of a session description.

7.1.8.2 Semantics for Local Descriptor and Remote Descriptor

Below, the semantics of the Local and Remote Descriptors are specified in detail. The specification consists of two parts. The first part specifies the interpretation of the contents of the descriptor. The second part specifies the actions the MG must take upon receiving the Local and Remote Descriptors. The actions to be taken by the MG depend on the values of the ReserveValue and ReserveGroup Properties of the LocalControl Descriptor.

7.1.8.2.1 Specification depth of Local Descriptor/Remote Descriptor content

Either the Local or the Remote Descriptor or both may be:

- unspecified (i.e., absent);
- empty;
- underspecified through use of CHOOSE in a property value;
- fully specified; or
- overspecified through presentation of multiple groups of properties and possibly multiple property values in one or more of these groups.

7.1.8.2.2 Interpretation of Local Descriptor/Remote Descriptor by the MG

Where the descriptors have been passed from the MGC to the MG, they are interpreted according to the rules given in clause 7.1.1, with the following additional comments for clarification:

- a) An unspecified Local or Remote Descriptor is considered to be a missing mandatory parameter. It requires the MG to use whatever was last specified for that descriptor. It is possible that there was no previously specified value, in which case the descriptor concerned is ignored in further processing of the command.
- b) An empty Local (Remote) Descriptor in a message from the MGC signifies a request to release any resources reserved for the media flow received (sent).
- c) If multiple groups of properties are present in a Local or Remote Descriptor or multiple values within a group, the order of preference is descending.
- d) Underspecified or overspecified properties within a group of properties sent by the MGC are requests for the MG to choose one or more values which it can support for each of those properties. In case of an overspecified property, the list of values is in descending order of preference.

7.1.8.2.3 Resource reservation rules for ReserveValue and ReserveGroup properties

Subject to the above rules, subsequent action depends on the values of the ReserveValue and ReserveGroup Properties in LocalControl.

7.1.8.2.3.1 ReserveValue = "False" AND ReserveGroup = "True"

If ReserveGroup is "True", the MG reserves the resources required to support as many as possible of the requested property group alternatives that it can currently support.

7.1.8.2.3.2 ReserveValue = "True" AND ReserveGroup = "False"

If ReserveValue is "True", the MG reserves the resources required to support as many as possible of the requested property value alternatives that it can currently support.

NOTE – If a Local or Remote Descriptor contains multiple groups of properties, and ReserveGroup is "True", then the MG is requested to reserve resources so that it can decode or encode the media stream according to any of the alternatives. For instance, if the Local Descriptor contains two groups of properties, one specifying packetized ITU-T G.711 A-law audio and the other ITU-T G.723.1 audio, the MG reserves resources so that it can decode one audio stream encoded in either ITU-T G.711 A-law format or ITU-T G.723.1 format. The MG does not have to reserve resources to decode two audio streams

simultaneously, one encoded in ITU-T G.711 A-law and one in ITU-T G.723.1. The intention for the use of ReserveValue is analogous.

7.1.8.2.3.3 ReserveValue = "True" OR ReserveGroup = "True"

If ReserveGroup is "True" or ReserveValue is "True", then the following rules apply:

- If the MG has insufficient resources to support all alternatives requested by the MGC, and the MGC requested resources in both Local and Remote, the MG should reserve resources to support at least one alternative each within Local and Remote.
- If the MG has insufficient resources to support at least one alternative within a Local (Remote) Descriptor received from the MGC, it shall return an empty Local (Remote) in response.
- In its response to the MGC, when the MGC included Local and Remote Descriptors, the MG shall include Local and Remote Descriptors for all groups of properties and property values it reserved resources for. If the MG is incapable of supporting at least one of the alternatives within the Local (Remote) Descriptor received from the MGC, it shall return an empty Local (Remote) Descriptor.
- If the Mode Property of the LocalControl Descriptor is "RecvOnly", "SendRecv", or "LoopBack", the MG must be prepared to receive media encoded according to any of the alternatives included in its response to the MGC.

7.1.8.2.3.4 ReserveValue = "False" AND ReserveGroup = "False"

If ReserveGroup is "False" and ReserveValue is "False", then the MG should apply the following rules to resolve Local and Remote to a single alternative each:

- The MG chooses the first alternative in Local for which it is able to support at least one alternative in Remote.
- If the MG is unable to support at least one Local and one Remote alternative, it returns Error Code 510 ("Insufficient resources").
- The MG returns its selected alternative in each of Local and Remote.

NOTE – The above rules allow the MG to prioritize the selection of the same codec in both the Local and Remote Descriptors; however, it also permits the MG to choose different codecs in each descriptor.

A new setting of a Local or Remote Descriptor completely replaces the previous setting of that descriptor in the MG. Thus, to retain information from the previous setting, the MGC must include that information in the new setting. If the MGC wishes to delete some information from the existing descriptor, it merely resends the descriptor (in a Modify Command) with the unwanted information stripped out.

7.1.9 Events Descriptor

7.1.9.1 Introduction to events

The Events Descriptor parameter contains a RequestID and a list of events that the Media Gateway is requested to detect and report. The RequestID is used to correlate the request with the notifications that it may trigger. Requested events include, for example, fax tones, continuity test results, and on-hook and off-hook transitions. The RequestID is omitted if the Events Descriptor is empty (i.e., no events are specified).

7.1.9.2 Specification, parameterization and identification of events

Each event in the descriptor contains the event name, an optional StreamID, an optional KeepActive flag, an optional NotifyBehaviour flag, an optional ResetEventsDescriptor flag and optional parameters. The event name consists of a PackageID (where the event is defined) and an EventID. The ALL wildcard may be used for the EventID, indicating that all events from the specified

package have to be detected. The default StreamID is 0, indicating that the event to be detected is not related to a particular media stream. Where an MGC has previously requested a Termination-level Event (i.e., default) and the MGC subsequently creates an additional stream, in order for the existing event/s to be applied to the new stream the MGC should also apply the event/s via an EventsDescriptor on the new stream.

Events can have parameters. This allows a single event description to have some variation in meaning without creating large numbers of individual events. Further event parameters are defined in the package.

If a DigitMap completion event is present or implied in the Events Descriptor, the EventDM Parameter is used to carry either the name or the value of the associated DigitMap. See clause 7.1.14 for further details.

7.1.9.3 Reporting of detected events

When an event is processed against the contents of an active Events Descriptor and found to be present in that descriptor ("recognized"), the default action of the MG is to send a Notify Command to the MGC. Notification may be deferred if the event is absorbed into the current dial string of an active DigitMap (see clause 7.1.14). The sending of a Notify Command may be influenced by the NotifyBehaviour flag. Moreover, event recognition may cause currently active signals to stop, or may cause the current Events and/or Signals Descriptor to be replaced, as described at the end of this subclause. Unless the Events Descriptor is replaced by another Events Descriptor, it remains active after an event has been recognized.

7.1.9.4 Buffering (and deferred reporting) of detected events

7.1.9.4.1 Event buffer concept

If the value of the EventBufferControl Property equals "LockStep", following detection of such an event, normal handling of events is suspended. Any event which is subsequently detected and occurs in the EventBuffer Descriptor is added to the end of the EventBuffer (a FIFO queue), along with the time that it was detected. The MG shall wait for a new Events Descriptor to be loaded. A new Events Descriptor can be loaded either as the result of receiving a command with a new Events Descriptor, or by activating an embedded Events Descriptor.

If EventBufferControl equals "Off", the MG continues processing based on the active Events Descriptor.

In the case of an embedded Events Descriptor being activated, the MG continues event processing based on the newly activated Events Descriptor.

NOTE – For purposes of EventBuffer handling, activation of an embedded Events Descriptor is equivalent to receipt of a new Events Descriptor.

7.1.9.4.2 Event buffer control

When the MG receives a command with a new Events Descriptor, one or more events may have been buffered in the EventBuffer in the MG. The value of EventBufferControl, then determines how the MG treats such buffered events.

Case 1

If EventBufferControl equals "LockStep" and the MG receives a new Events Descriptor, it will check the FIFO EventBuffer and take the following actions:

- 1) If the EventBuffer is empty, the MG waits for detection of events based on the new Events Descriptor.

- 2) If the EventBuffer is non-empty, the MG processes the FIFO queue starting with the first event:
- If the event in the queue is in the events listed in the new Events Descriptor, the MG acts on the event and removes the event from the EventBuffer. The time stamp of the Notify shall be the time the event was actually detected. The MG then waits for a new Events Descriptor. While waiting for a new Events Descriptor, any events detected that appear in the EventBuffer Descriptor will be placed in the EventBuffer. When a new Events Descriptor is received, the event processing will repeat from step 1).
 - If the event is not in the new Events Descriptor, the MG shall discard the event and repeat from step 1).

Case 2

If EventBufferControl equals "Off" and the MG receives a new Events Descriptor, it processes new events with the new Events Descriptor.

If the MG receives a command instructing it to set the value of EventBufferControl to "Off", all events in the EventBuffer shall be discarded.

The MG may report several events in a single transaction as long as this does not unnecessarily delay the reporting of individual events.

For procedures regarding transmitting the Notify Command, refer to the appropriate annex or ITU-T H.248.x Recommendation for specific transport considerations.

The default value of EventBufferControl is "Off".

NOTE – Since the EventBufferControl Property is in the TerminationState Descriptor, the MG might receive a command that changes the EventBufferControl Property and does not include an Events Descriptor.

7.1.9.5 Interaction between events and signals

Normally, recognition of an event shall cause any active signals to stop. When KeepActive is specified in the event, the MG shall not interrupt any signals active on the termination on which the event is detected.

7.1.9.6 Shaping the events reporting traffic

The NotifyBehaviour flag may be used to indicate that the Notify Command is:

- sent immediately (NotifyImmediate: this is the default);
- never sent (NeverNotify);
- or regulated (i.e., sent or suppressed) according to the MGC load (RegulatedNotify).

See clause E.15 for more details on the use of NotifyBehaviour. When used with DigitMaps the notify behaviour occurs when the active DigitMap is completed. The Regulated Notify may have an alternate regulated embedded Events or Signals Descriptor associated with it. If a notification is regulated (i.e., suppressed), then this alternate regulated embedded descriptor shall be activated. If the Notify is not regulated, then the original embedded descriptor is triggered. If NotifyImmediate or NeverNotify is set, then any original embedded descriptor is triggered on detection of the event.

7.1.9.7 Resetting events

The result of encountering the ResetEventsDescriptor flag set against an event depends on whether or not the Events Descriptor containing the event is an embedded Events Descriptor. In the case of an embedded Events Descriptor the ResetEventsDescriptor flag causes the Events Descriptor for that termination to be reset to its state prior to when the embedded descriptor was activated (i.e., the last Events Descriptor explicitly set by a Modify Command or, if there has been no Modify Command since the termination was last reset by being put into the NULL Context, the

MG-provisioned Events Descriptor). In the case of a non-embedded Events Descriptor the ResetEventsDescriptor flag causes the active Events Descriptor for that termination to be reset by re-activating any DigitMap completion event(s) that have been matched and deactivated.

The ResetEventsDescriptor flag shall be set on physical terminations only. The ResetEventsDescriptor flag shall be acted on when the termination is in the NULL Context. If the ResetEventsDescriptor flag is encountered when the termination is out of the NULL Context, then it shall have no effect.

7.1.9.8 Hierarchical events and signals: embedded Events/Signals Descriptors

An event can include an embedded Signals Descriptor and/or an embedded Events Descriptor which, if present, replaces the current Signals/Events Descriptor when the event is recognized. It is possible, for example, to specify that the dial-tone signal be generated when an off-hook event is recognized, or that the dial-tone signal be stopped when a digit is recognized. A media gateway controller shall not send Events Descriptors with an event both marked KeepActive and containing an embedded Signals Descriptor.

Only one level of embedding is permitted. An embedded Events Descriptor shall not contain another embedded Events Descriptor; an embedded Events Descriptor may contain an embedded Signals Descriptor.

7.1.9.9 Consecutive Events Descriptors

An Events Descriptor received by a media gateway replaces any previous Events Descriptor. Event notification in process shall complete, and events detected after the command containing the new Events Descriptor executes, shall be processed according to the new Events Descriptor.

An empty Events Descriptor disables all event recognition and reporting. An empty EventBuffer Descriptor clears the EventBuffer and disables all event accumulation in "LockStep" mode: the only events reported will be those occurring while an Events Descriptor is active. If an empty Events Descriptor is activated while the termination is operating in "LockStep" mode, the EventBuffer is immediately cleared.

7.1.10 EventBuffer Descriptor

The EventBuffer Descriptor contains a list of events, with their parameters if any, that the MG is requested to detect and buffer when EventBufferControl equals LockStep (see clause 7.1.9).

7.1.11 Signals Descriptor

7.1.11.1 Introduction to Signals

Signals are MG-generated media such as tones and announcements as well as bearer-related signals such as hookswitch. More complex signals may include a sequence of such simple signals interspersed with and conditioned upon the receipt and analysis of media or bearer-related signals. Examples include echoing of received data as in the Basic Continuity Package (clause E.10). Signals may also request preparation of media content for future signals.

7.1.11.2 Overview of Signals Descriptor

A Signals Descriptor is a parameter that contains the set of signals that the Media Gateway is asked to apply to a termination. A Signals Descriptor contains a number of signals and/or sequential signal lists. A Signals Descriptor may contain zero signals and sequential signal lists. Support of sequential signal lists is optional.

7.1.11.3 Specification and identification of signals

Signals are defined in packages. Signals shall be named with a PackageID (in which the signal is defined) and a SignalID. No wildcard shall be used in the SignalID. Signals that occur in a Signals Descriptor have an optional StreamID parameter (default is 0, to indicate that the signal is not related to a particular media stream), an optional signal type (see below), an optional duration and possibly parameters defined in the package that defines the signal. This allows a single signal to have some variation in meaning, obviating the need to create large numbers of individual signals.

7.1.11.4 Stream-level signals

In case the MGC does not specify the StreamID parameter (or sets the StreamID parameter value explicitly to 0) the Signals Descriptor shall be applied to all streams (including any existing Streams and any new Stream defined by the command containing the Signals Descriptor).

Therefore, where an MGC has previously requested that a Signals Descriptor be applied to all streams in a termination and the MGC subsequently creates an additional stream, in order for the existing signals to be applied to the new stream, the MGC shall also apply the Signals Descriptor to the new stream.

7.1.11.5 Indication of signal completion

The optional NotifyCompletion Parameter allows an MGC to indicate that it wishes to be notified when the signal finishes playout. The possible cases are that the signal timed out (or otherwise completed on its own), that it was interrupted by an event, that it completed a cycle/iteration of the signal, that it was halted when a Signals Descriptor was replaced, or that it stopped or never started for other reasons. If the NotifyCompletion Parameter is not included in a Signals Descriptor, notification is generated only if the signal stopped or was never started for other reasons. For reporting to occur, the Signal Completion Event (see clause E.1.2) must be enabled in the currently active Events Descriptor. The optional parameter "RequestID" may be associated with a particular instance of a SignalID where multiple signals of the same SignalID are requested. This allows the MGC to distinguish between different Signal Completion ObservedEvents for that particular instance of a signal. The RequestID parameter shall not be included if the NotifyCompletion Parameter is not present.

7.1.11.6 Duration of Signals

The duration is an integer value that is expressed in hundredths of a second.

7.1.11.7 Types of Signals

There are three types of signals:

- OnOff (OO): the signal lasts until it is turned off either with an empty Signals Descriptor or with a replacement Signals Descriptor that removes the signal from the descriptor;
- TimeOut (TO): the signal lasts until it is turned off (as above) or a specific period of time elapses. The signal will remain in the Signals Descriptor until removed;
- Brief (BR): the signal will stop on its own unless a new Signals Descriptor is applied that causes it to stop; no timeout value is needed. The signal shall remain in the Signals Descriptor unless removed. Due to the brief nature of the signals, even if the new Signals Descriptor is empty or excludes the signal, the signal is not guaranteed to be stopped (or changes done by start cannot be undone) as it may have already completed playout.

If a signal of default type other than TO has its type overridden to type TO in the Signals Descriptor, the duration parameter must be present.

If the signal type is specified in a Signals Descriptor, it overrides the default signal type (see clause 12.1.4). It is not possible to change the semantics of a signal by overriding the signal type. If duration is specified for an on/off signal, it shall be ignored.

Copyright International Telecommunication Union/ITU

7.1.11.8 List of signals

A sequential signal list consists of a SignalListID and a sequence of signals to be played sequentially. Only the trailing element of the sequence of signals in a sequential signal list may be an on/off signal. The duration of a sequential signal list is the sum of the durations of the signals it contains plus the sum of the intersignal delay timings specified as parameters to the signals.

If an intersignal delay is specified for a signal that is not in a sequential signal list or is the last element in a sequential signal list, it shall be ignored and the signal shall be determined to have completed at the termination of the signal, prior to the application of the intersignal delay timing. The duration of a signal in a SignalList with an intersignal delay includes the intersignal delay timing.

Multiple signals and sequential signal lists in the same Signals Descriptor shall be played simultaneously.

7.1.11.9 Directionality of signals

Signals have directionality, which is identified by the direction parameter in the base syntax or by package defined direction parameters. The direction parameter can be used to indicate the direction that the tone is to be sent. External indicates that the tone is sent from the MG to an external point. Internal indicates that the tone is played into the Context to the other terminations. Bothway indicates both internal and external behaviour.

Signals are defined as proceeding from the termination towards the exterior of the context unless otherwise specified. If the signal direction is specified in a Signals Descriptor, it overrides the default signal direction. For those signals which have a direction parameter, the base protocol direction parameter takes precedence over any package-defined direction parameter when both are specified. If the MGC specifies a direction for a signal with which the MG cannot comply, the MG shall return Error Code 501 ("Not implemented"). When the same signal is applied to multiple terminations within one transaction, the MG should consider using the same resource to generate these signals.

7.1.11.10 Stop of signal generation

Production of a signal on a termination is stopped by application of a new Signals Descriptor, or detection of an event on the termination (see clause 7.1.9).

7.1.11.11 Consecutive Signals Descriptor

A new Signals Descriptor replaces any existing Signals Descriptor. Any already started and not yet stopped signals applied to the termination not in the replacement descriptor (which may be empty) shall be stopped irrespective of the signal type, and new signals are applied, except as follows. Signals present in the replacement descriptor and containing the KeepActive flag shall be continued if they are currently playing and have not already completed. If a replacement Signals Descriptor contains a signal that is not currently playing and contains the KeepActive flag, that signal shall be ignored. If the replacement descriptor contains a sequential signal list with the same identifier as the existing descriptor, then:

- the signal type and sequence of signals in the sequential signal list in the replacement descriptor shall be ignored; and
- the playing of the signals in the sequential signal list in the existing descriptor shall not be interrupted.

7.1.12 Audit Descriptor

The Audit Descriptor specifies what information is to be audited. The Audit Descriptor specifies the list of descriptors and/or individual properties to be returned. Audit may be used in any command to force the return of any descriptor containing the current values of its properties, events, signals and statistics even if that descriptor was not present in the command, or had no underspecified parameters. Possible items in the Audit Descriptor are:

Modem (Deprecated, see clause 7.1.2)	
Mux	
Events	
Media	
Signals	
ObservedEvents	
DigitMap	
Statistics	
Packages	
EventBuffer	
Individual Audit Items:	
Media Properties	DigitMaps
Events	Statistics
EventBuffer	Packages
Signals, SignalLists	ContextAttribute

The Audit Descriptor may be empty, in which case, no descriptors are returned. This is useful in the Subtract Command to inhibit return of statistics, especially when using wildcard.

7.1.13 ServiceChange Descriptor

The ServiceChange Descriptor contains the following parameters:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeAddress;
- ServiceChangeDelay;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMGCID;
- TimeStamp;
- Extension;
- ServiceChangeInfo;
- ServiceChangeIncompleteFlag.

See clause 7.2.8.

7.1.14 DigitMap Descriptor

7.1.14.1 DigitMap definition, creation, modification and deletion

A DigitMap is a dialling plan resident in the Media Gateway used for detecting and reporting digit events received on a termination. The DigitMap Descriptor contains a DigitMap name and the DigitMap to be assigned. A DigitMap may be preloaded into the MG by management action and referenced by name in an Events Descriptor, may be defined dynamically and subsequently referenced by name, or the actual digitmap itself may be specified in the Events Descriptor. It is permissible for a DigitMap completion event within an Events Descriptor to refer by name to a DigitMap which is defined by a DigitMap Descriptor within the same command, regardless of the transmitted order of the respective descriptors.

DigitMaps defined in a DigitMap Descriptor can occur in any of the standard termination manipulation commands of the protocol. A DigitMap, once defined, can be used on all terminations specified by the (possibly wildcarded) TerminationID in such a command. DigitMaps defined on the Root Termination are global and can be used on every termination in the MG, provided that a DigitMap with the same name has not been defined on the given termination. When a DigitMap is defined dynamically in a DigitMap Descriptor:

- A new DigitMap is created by specifying a name that is not yet defined. The value shall be present.
- A DigitMap value is updated by supplying a new value for a name that is already defined. Terminations presently using the DigitMap shall continue to use the old definition; subsequent Events Descriptors specifying the name, including any Events Descriptor in the command containing the DigitMap Descriptor, shall use the new one.
- A DigitMap is deleted by supplying an empty value for a name that is already defined. Terminations presently using the DigitMap shall continue to use the old definition.

7.1.14.2 DigitMap timers

The collection of digits according to a DigitMap may be protected by three timers, viz. a start timer (T), short timer (S) and long timer (L).

- 1) The start timer (T) is used prior to any digits being available for processing against the DigitMap. If the start timer is overridden with the value set to zero ($T = 0$), then the start timer shall be disabled. This implies that the MG will wait indefinitely for digits.
- 2) If the Media Gateway can determine that at least one more digit is needed for a digit string to match any of the allowed patterns in the DigitMap, then the interdigit timer value should be set to a long (L) duration (e.g., 16 seconds).
- 3) If the digit string has matched one of the patterns in a DigitMap, but it is possible that more digits could be received which would cause a match with a different pattern, then instead of reporting the match immediately, the MG must apply the short timer (S) and wait for more digits.

The timers are configurable parameters to a DigitMap. Default values of these timers should be provisioned on the MG, but can be overridden by values specified within the DigitMap.

7.1.14.3 DigitMap syntax

The formal syntax of the DigitMap is described by the DigitMap rule in the formal syntax description of the protocol (see Annexes A and B). A DigitMap, according to this syntax, is defined either by a string or by a list of strings. Each string in the list is an alternative event sequence, specified either as a sequence of DigitMap symbols or as a regular expression of DigitMap symbols. These DigitMap symbols, the digits "0" through "9" and letters "A" through a maximum value depending on the signalling system concerned, but never exceeding "K", correspond to specified events within a package which has been designated in the Events Descriptor on the termination to

which the DigitMap is being applied. (The mapping between events and DigitMap symbols is defined in the documentation for packages associated with channel-associated signalling systems such as DTMF, MF, or R2. Digits "0" through "9" must be mapped to the corresponding digit events within the signalling system concerned. Letters should be allocated in logical fashion, facilitating the use of range notation for alternative events.)

The letter "x" is used as a wildcard, designating any event corresponding to symbols in the range "0"-"9". The string may also contain explicit ranges and, more generally, explicit sets of symbols, designating alternative events any one of which satisfies that position of the DigitMap. Finally, the dot symbol "." stands for zero or more repetitions of the event selector (event, range of events, set of alternative events, or wildcard) that precedes it. As a consequence of the third timing rule above, inter-event timing while matching a terminal dot symbol uses the short timer by default.

In addition to these event symbols, the string may contain "S" and "L" inter-event timing specifiers and the "Z" duration modifier. "S" and "L" respectively indicate that the MG should use the short (S) timer or the long (L) timer for subsequent events, overriding the timing rules described above. If an explicit timing specifier is in effect in one alternative event sequence, but none is given in any other candidate alternative, the timer value set by the explicit timing specifier must be used. If all sequences with explicit timing controls are dropped from the candidate set, timing reverts to the default rules given above. If used inside a range notation, the S and L specifiers shall be ignored. Finally, if conflicting timing specifiers are in effect in different alternative sequences, the long timer shall be used.

A "Z" designates a long duration event: placed in front of the symbol(s) designating the event(s) which satisfy a given digit position, it indicates that that position is satisfied only if the duration of the event exceeds the long-duration threshold. The value of this threshold is assumed to be provisioned in the MG, but, like the T, L, and S timers, can be overridden by specification within the DigitMap. If the Z specifier is not followed by a digit (0-9 or A-K), then the MG shall reject the digitmap as invalid procedure. When used in a range notation, the Z specifier applies solely to the immediately following digit. When used immediately prior to a range, the Z modifier applies to all digits in the range (thereby requiring a match in the range to be long duration).

7.1.14.4 DigitMap completion event

A DigitMap is active while the Events Descriptor which invoked it is active and it has not completed. A DigitMap completes when:

- a timer has expired; or
- an alternative event sequence has been matched and no other alternative event sequence in the DigitMap could be matched through detection of an additional event (unambiguous match); or
- an event has been detected such that a match to a complete alternative event sequence of the DigitMap will be impossible no matter what additional events are received.

Upon completion, a DigitMap completion event, as defined in the package providing the events being mapped into the DigitMap, shall be generated. At that point, the DigitMap is deactivated. Subsequent events in the package are processed as follows:

- If EventBufferControl is "LockStep", subsequent digit events are processed in the same way as any other events;
- If EventBufferControl is "OFF", if the active Events Descriptor did not change, and if individual digit events are not enabled within that descriptor, then digit buffering will be initiated. Buffering will continue until the buffering time specified in the original DigitMap completion event has expired or until the active Events Descriptor is replaced.

The digit buffer shall take the logical form of a dial string which includes the digit characters as represented in the DigitMap, possibly preceded by 'Z'. The threshold value of tone duration used to identify long events shall be the same as that used with the most recently completed DigitMap.

Buffering time defaults to zero (no buffering) unless explicitly set otherwise within the DigitMap completion event. If buffering ceases due to buffering timer expiry, the contents of the digit buffer are discarded.

If buffering was stopped by a new Events Descriptor and that Events Descriptor contains a new DigitMap completion event from the same package as the previous one, any buffered digits are processed against the DigitMap as described below. Buffered digits not consumed by the new DigitMap are handled as if they had been observed after that map completed.

If, instead, the new Events Descriptor enables the reporting of individual digit events, the entire set of buffered digits shall immediately be processed, the applicable events reported, and the buffer cleared.

Finally, if the new Events Descriptor enables neither a DigitMap completion event nor the reporting of individual digit events from the package concerned, the buffer contents are discarded and buffering is terminated.

7.1.14.5 DigitMap procedures

Pending completion, successive events shall be processed according to the following rules:

- 1) The "current dial string", an internal variable, is initially empty. The set of candidate alternative event sequences includes all of the alternatives specified in the DigitMap.
- 2) At each step, if buffered digits are available, the oldest one (with possible accompanying long duration (Z) qualifier) is removed from the buffer and processing moves to the next step as if the digit event had just been observed. Otherwise a timer is set to wait for the next event, based either on the default timing rules given above, or on explicit timing specified in one or more alternative event sequences. If the timer expires and a member of the candidate set of alternatives is fully satisfied, a timeout completion with full match is reported. If the timer expires and part or none of any candidate alternative is satisfied, a timeout completion with partial match is reported. In either case, if the DigitMap completion event allows for detailed timeout reporting, the reported dial string will end with 'L', 'S', or 'T' as appropriate.
- 3) If an event is detected before the timer expires, it is mapped to a digit string symbol and provisionally added to the end of the current dial string. The duration of the event (long or not long) is noted if, and only if, this is relevant in the current symbol position (because at least one of the candidate alternative event sequences includes the "Z" modifier at this position in the sequence).
- 4) The current dial string is compared to the candidate alternative event sequences. If, and only if, a sequence expecting a long-duration event at this position is matched (i.e., the event had long duration and met the specification for this position), then any alternative event sequences not specifying a long duration event at this position are discarded, and the current dial string is modified by inserting a "Z" in front of the symbol representing the latest event. Any sequence expecting a long-duration event at this position, but not matching the observed event, is discarded from the candidate set. If alternative event sequences not specifying a long duration event in the given position remain in the candidate set after application of the above rules, the observed event duration is treated as irrelevant in assessing matches to them.

- 5) If exactly one candidate remains and it has been fully matched, a completion event is generated indicating an unambiguous match. If no candidates remain, the latest event is removed from the current dial string and a completion event is generated indicating full match if one of the candidates from the previous step was fully satisfied before the latest event was detected or partial match otherwise. The event removed from the current dial string will then be reported as a separate event, buffered, or discarded according to the rules described in the previous clause. This statement is qualified as follows:
- a) The DigitMap completion event may specify that the removed event be reported as a parameter of the completion event. This occurs independently of subsequent processing of the digit event.
 - b) The DigitMap completion event may specify that the extra digit should be discarded. In this case, it is discarded immediately. Any buffering or other processing applies only to subsequent events.
- 6) If no completion event is reported out of step 5), processing returns to step 2).

7.1.14.6 DigitMap activation

A DigitMap is activated whenever a new Events Descriptor is applied to the termination or embedded Events Descriptor is activated, and that Events Descriptor contains a DigitMap completion event. The DigitMap completion event contains an EventDM field in the requested actions field. Each new activation of a DigitMap begins at step 1 of the above procedure, with a clear current dial string. Any previous contents of the current dial string from an earlier activation are lost.

A DigitMap completion event that does not contain an EventDM field in its requested actions field is considered an error. Upon receipt of such an event in an Events Descriptor, an MG shall respond with an error response, including Error Code 457 ("Missing parameter in signal or event").

7.1.14.7 Interaction of DigitMap and event processing

While the DigitMap is activated, detection is enabled for all events defined in the package containing the specified DigitMap completion event. Normal event behaviour (e.g., stopping of signals unless the digit completion event has the KeepActive flag enabled) continues to apply for each such event detected, except that:

- the events in the package containing the specified DigitMap completion event other than the completion event itself are not individually notified and have no side-effects unless separately enabled; and
- an event that triggers a partial match completion event is not recognized and therefore has no side effects until reprocessed following the recognition of the DigitMap completion event. Similarly buffered digit events are not recognized and have no side effects until processed.

7.1.14.8 Wildcards

Note that if a package contains a DigitMap completion event, then an event specification consisting of the PackageID with a wildcarded EventID will activate a DigitMap; to that end, the event specification must include an EventDM field according to clause 7.1.14.6. If the package also contains the digit events themselves, this form of event specification will cause the individual events to be reported to the MGC as they are detected.

7.1.14.9 Example

As an example, consider the following dial plan:

0	Local operator
00	Long-distance operator
xxxx	Local extension number (starts with 1-7)
8xxxxxxx	Local number
#xxxxxxx	Off-site extension
*xx	Star services
91xxxxxxxxxx	Long-distance number
9011 + up to 15 digits	International number

If the DTMF detection package described in clause E.6 is used to collect the dialled digits, then the dialling plan shown above results in the following DigitMap:

```
(0 | 00 | [1-7]xxx | 8xxxxxxx | Fxxxxxxx | Exx | 91xxxxxxxxxx | 9011x.)
```

7.1.15 Statistics Descriptor

The Statistics Descriptor provides information describing the status and usage of a termination during its existence (ephemeral) or while it is outside the NULL context (physical). Statistics may be at a termination level or associated with a particular Stream Descriptor. By default, statistics occur at a termination level. If a stream is unable to support a statistic, Error Code 460 ("Unable to set statistic on stream") shall be returned. There is a set of standard statistics kept for each termination where appropriate (number of octets sent and received for example). By default, the particular statistics that are reported for a given termination are determined by the Packages realized by the termination. It is also possible, using the descriptor, to indicate what statistics are to be collected. The setting of a Statistics Descriptor overrides a previously set Statistics Descriptor. Thus, to maintain already set statistics, these shall be included in the new Statistics Descriptor and the statistic values shall not be reset. Statistics that have been removed from a Statistics Descriptor shall maintain their values until the termination is subtracted. However, if the particular statistic is reactivated by a subsequent Statistics Descriptor, the value shall be reset. An MGC may reactivate all statistics on a termination/stream by issuing a Statistics Descriptor with a single statistic with the PackageID and StatisticID wildcarded with "ALL". To reactivate all statistics in a given package on a termination/stream, a Statistics Descriptor with a single statistic with the PackageID specified and the StatisticID wildcarded with "ALL" is issued. The receipt of an empty descriptor means that no statistics shall be collected for the specified termination.

By default, unless an empty Statistics Descriptor had earlier been used to indicate that no statistics are to be collected, termination and stream level statistics are reported when the termination ceases to exist or is returned to the NULL context due to a Subtract command. This default behaviour can also be overridden by including an empty Audit Descriptor in the Subtract Command. If a stream is removed according to clause 7.1.6, by default stream level statistics are not reported. An audit of the Statistics Descriptor in the stream must be performed before the stream is removed to collect the statistics. Statistics may also be returned from the AuditValue Command, or any Add/Move/Modify Command using the Audit Descriptor.

Statistics are cumulative; reporting statistics does not reset them. The value of a Statistic at a termination level is the result of a meaningful superior function (like, for instance, sum or average) of the values as if it had been placed on all the streams in the termination. Such a superior function is dependent on the particular statistic type. Unless specified otherwise in the package that defines a particular statistic, the default behaviour is a sum of the values. Statistics are reset when a termination ceases to exist or is returned to the NULL Context due to a Subtract Command.

For terminations not in the NULL Context, the MGC may send a Modify Command with a Statistics Descriptor to disable one or more statistics followed by a second Modify Command including a Statistics Descriptor to re-enable those statistics. As explained above, this has the effect of resetting the included statistics. By bundling the two commands together into the same action or transaction, the MGC can minimize the time during which statistics are not collected by the MG. An audit of the Statistics Descriptor via the Audit Descriptor in the Modify Command or a separate AuditValue Command must be performed before the included statistics are reactivated to collect their values.

7.1.16 Packages Descriptor

Used only with the AuditValue command, the Packages Descriptor returns a list of packages realized by the termination.

7.1.17 ObservedEvents Descriptor

ObservedEvents is supplied with the Notify Command to inform the MGC of which event(s) were detected. Used with the AuditValue Command, the ObservedEvents Descriptor returns events in the event buffer which have not been notified. ObservedEvents contains the RequestID of the Events Descriptor that triggered the notification, the event(s) detected, optionally the detection time(s) and any parameters of the observed event. Detection times are reported with a precision of hundredths of a second.

The ObservedEvents Descriptor may contain the StreamID associated with the detected Event. If the MGC has set the Event on a termination level, the MG may or may not report the StreamID depending on the semantic of the event. If the MGC has set the Event on a particular stream then this StreamID shall be returned.

7.1.18 Topology Descriptor

7.1.18.1 General

A Topology Descriptor is used to specify flow directions between terminations in a context. Contrary to the descriptors in previous clauses, the Topology Descriptor applies to a context instead of a termination. The default topology of a context is that each termination's transmission is received by all other terminations. When a Termination is added to a Context, its default Topology is "bothway" to the other Terminations in the Context. If another Topology is required this shall be indicated in any updated Topology Descriptor. The Topology Descriptor is optional to implement. An MG that does not support Topology Descriptors, but receives a command containing one, returns Error Code 444 ("Unsupported or unknown descriptor"), and optionally includes a string containing the name of the unsupported descriptor ("Topology") in the error text in the Error Descriptor.

The Topology Descriptor occurs before the commands in an action. It is possible to have an action containing only a Topology Descriptor, provided that the context to which the action applies already exists.

7.1.18.2 Structure (syntax) of the Topology Descriptor

A Topology Descriptor consists of a sequence of associated terminations of the form (*T1*, *T2*, *association*[*StreamID*]). *T1* and *T2* specify terminations within the context, possibly using the ALL or CHOOSE wildcard. If the optional StreamID field is used, the association applies only to the particular stream between *T1* and *T2* labelled by the StreamID. If the StreamID field is omitted, the topology applies to all streams in the termination.

7.1.18.3 Descriptor element "association"

The *association* specifies how media flows between these two terminations as follows:

- (*T1*, *T2*, Isolate) means that the terminations matching *T2* do not receive media from the terminations matching *T1*, nor vice versa.
- (*T1*, *T2*, Oneway) means that the terminations that match *T2* receive media from the terminations matching *T1*, but not vice versa. In this case, use of the ALL wildcard such that there are terminations that match either *T1* or *T2* *but not both* is allowed.
- (*T1*, *T2*, OnewayExternal) means the terminations that match *T2*, receive media sent externally by terminations matching *T1*, but not vice versa. In this case, use of the ALL wildcard for *T1* is not allowed.
- (*T1*, *T2*, OnewayBoth) means the terminations that match *T2*, receive media sent and received externally by terminations matching *T1*, but not vice versa. In this case, use of the ALL wildcard for *T1* and/or *T2* is not allowed.
- (*T1*, *T2*, Bothway) means that the terminations matching *T2* receive media from the terminations matching *T1*, and vice versa. In this case it is allowed to use wildcards such that there are terminations that match both *T1* and *T2*. However, if there is a termination that matches both, no loopback is introduced.

7.1.18.4 Wildcarding of TerminationID elements

7.1.18.4.1 Wildcard CHOOSE

CHOOSE wildcards may be used in *T1* and *T2* as well, under the following restrictions:

- the action (see clause 8) of which the Topology Descriptor is part contains an Add Command in which a CHOOSE wildcard is used;
- if a CHOOSE wildcard occurs in *T1* or *T2*, then a partial name (underspecified Termination ID) shall not be specified.

The CHOOSE wildcard in a Topology Descriptor matches the TerminationID that the MG assigns in the first Add Command that uses a CHOOSE wildcard in the same action. An existing termination that matches *T1* or *T2* in the context to which a termination is added is connected to the newly added termination as specified by the Topology Descriptor. If a termination is not mentioned within a Topology Descriptor, any topology associated with it remains unchanged. If, however, a new termination is added into a context, its association with the other terminations within the context defaults to Bothway, unless a Topology Descriptor is given to change this (e.g., if *T3* is added to a context with *T1* and *T2* with topology (*T3*, *T1*, Oneway) it will be connected bothway to *T2*).

7.1.18.4.2 Wildcard ALL

ALL wildcard may be used in *T1* or *T2* as well, under the following restrictions:

- "all" excludes the other termination in the Topology Descriptor (this semantic is in line with clause 6.2.2).

ALL wildcard may not be used for both *T1* and *T2* in a single descriptor with a one-way association.

7.1.18.4.3 Combination of Wildcard ALL and CHOOSE in a single Topology Descriptor

ALL wildcard may be used in *T1* and CHOOSE wildcard may be used in *T2*, or vice versa. The chosen termination is then excluded from the "all" set of the remaining terminations.