



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T H.248.1 Version 1 Implementors' Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(13 April 2006)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS
Infrastructure of audiovisual services – Communication
procedures

**Implementors' Guide for Recommendation
H.248.1 Version 1 (03/2002) ("Media Gateway
Control Protocol")**

Summary

This document is a compilation of reported defects identified in ITU-T Recommendation H.248.1 Version 1 (03/2002). It must be read in conjunction with the Recommendation to serve as an additional authoritative source of information for implementors.

This revision contains all updates submitted up to and including those at the Study Group 16 meeting in April 2006 and supersedes the earlier version approved 26 November 2004.

This document was approved by ITU-T Study Group 16 on [13 April 2006].

NOTE: Changes relative to H.248.1 Version 2 (05/2002) and onwards are found in other documents.

Change Log

(All changes that were included in H.248.1 v1 (03/2002) are omitted here.)

V10 (Bruges, June 2002)

Changed references to H.248 Amendment 1 to H.248.1.

New:

- 6.1 Specify types for rtp/jit and rtp/delay in Annex E.12.4
- 6.2 Define the '#' symbol in INEQUAL in text encoding
- 6.3 Empty Descriptor Syntax
- 6.4 Define the symbol for NULL Context in text encoding
- 6.5 Corrections to Appendix A example statistics
- 6.6 Corrections to Package Guidelines for Statistics in 12.1.5
- 6.7 Specification of the meaning of automatic in E.13 tdm package

V11 (Geneva, October 2002)

Modification:

- 6.7 Added additional changes to gain

New:

- 6.8 Protocol Version Negotiation
- 6.9 Statistics and Move
- 6.10 Additional Codepoint for Annex C
- 6.11 Wildcarding Principles

V12 (San Jose, February 2003)

New:

- 6.12 Wildcarding in the Topology Descriptor
- 6.13 Binary Value for Packetization Time (Annex C)

V13 (Geneva, May 2003)

New:

- 6.14 Modification of Terminations by MGCs
- 6.15 Optional Command in an Action
- 6.16 Ordering of Transactions
- 6.17 Replies to Actions with no Commands

V14 (Paris, September 2003)

Made several editorial changes to the text.

Modification:

- 6.3 Ambiguous Audit and Individual Audit Return (Changed title and description – no technical change)

New:

- 6.18 Network Package can apply to TDM

V15 (Geneva, January 2004) [TD 55R1/Plen]

Modified:

- 6.17 This item is deprecated in favor of 6.23

New:

- 6.19 Precedence of LocalControl Mode property versus SDP mode
- 6.20 Digit processing clarification
- 6.21 Usage of DigitMap timer symbols with range notation
- 6.22 Clarification of the use of StreamID = 0
- 6.23 Correction of Context Audit Return
- 6.24 Clarification of return value for AuditCapabilities of strings

V16 (Beijing, May 2004) [TD 45]

New:

- 6.25 Error response when processing a ContextID
- 6.26 Support of packages
- 6.27 Mismatch between RFC2377 support and one “m=” line restriction
- 6.28 Annex C codepoints for RTCP
- 6.29 Clarification of PackageID and name for Annex C

V17 (San Jose, September 2004) [TD 35]

New:

- 6.30 Clarification of ReserveGroup and ReserveValue Properties
- 6.31 Clarification of Provisional Response Timer Values
- 6.32 Clarification of NULL Context Usage

V18 (Geneva, November 2004)

New:

- 6.33 Commands in ServiceChange on Root Transaction
- 6.34 Loopback Usage Clarification

V19 (Melbourne, February 2005)

New:

- 6.35 Annex C and SDP parameters
- 6.36 Case sensitivity of Profile Names
- 6.37 Profile Negotiation

V20 (Geneva, August 2005)

New:

6.38 Media Type Mismatch

6.39 Notify Avalanche

6.40 Topology Reply

V21 (Geneva, November 2005)

New:

6.41 Protocol version negotiation

V22 (Geneva, April 2006)

Modification:

6.11 Wildcarding Principles

New:

6.42 ServiceStates clarification for continuity testing

6.43 Clarification of termination service state upon restart of MG

6.44 Alignment of text among events in the Tone Detection Package

6.45 Clarification of package definition requirements for enumerations

6.46 Clarification of use of ABNF encodings of octet strings

6.47 Clarification of encoding for packet loss statistic in Annex E.12

Contact Information

ITU-T Study Group 16 / Christian Groves
Question 3 Rapporteur Australia

Tel: +61 3 9301 6116
E-mail: Christian.Groves@nteczone.com

H.248 Sub-series Kevin Boyle II
Implementors' Guide USA
Editor

Tel: +1 919 991 2690
E-mail: kboyle@nortel.com

Table of Contents

INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
1 SCOPE	1
2 INTRODUCTION	1
3 DEFECT RESOLUTION PROCEDURE	1
4 REFERENCES	1
5 NOMENCLATURE	2
6 TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.1 VERSION 1 (03/2002)	2
6.1 SPECIFY TYPES FOR RTP/JIT AND RTP/DELAY IN ANNEX E.12.4.....	2
6.2 DEFINE THE '#' SYMBOL IN INEQUAL IN TEXT ENCODING	3
6.3 AMBIGUOUS AUDIT AND INDIVIDUAL AUDIT RETURN	3
6.4 DEFINE THE SYMBOL FOR NULL CONTEXT IN TEXT ENCODING	4
6.5 CORRECTIONS TO APPENDIX A EXAMPLE STATISTICS.....	5
6.6 CORRECTIONS TO PACKAGE GUIDELINES FOR STATISTICS IN 12.1.5	6
6.7 SPECIFICATION OF THE MEANING OF AUTOMATIC IN E.13 TDM PACKAGE	6
6.8 PROTOCOL VERSION NEGOTIATION.....	8
6.9 STATISTICS AND MOVE.....	9
6.10 ADDITIONAL CODEPOINT FOR ANNEX C.....	10
6.11 WILD CARDING PRINCIPLES	11
6.12 WILDCARDING IN THE TOPOLOGY DESCRIPTOR	14
6.13 BINARY VALUE FOR PACKETIZATION TIME (ANNEX C)	14
6.14 MODIFICATION OF TERMINATIONS BY MGCs	15
6.15 OPTIONAL COMMANDS IN AN ACTION.....	15
6.16 ORDERING OF TRANSACTIONS	16
6.17 REPLIES TO ACTIONS WITH NO COMMANDS	16
6.18 NETWORK PACKAGE CAN APPLY TO TDM	17
6.19 PRECEDENCE OF LOCALCONTROL MODE PROPERTY VERSUS SDP MODE	17
6.20 DIGIT PROCESSING CLARIFICATION.....	17
6.21 USAGE OF DIGITMAP TIMER SYMBOLS WITH RANGE NOTATION	18
6.22 CLARIFICATION OF THE USE OF STREAMID = 0.....	19
6.23 CORRECTION OF CONTEXT AUDIT RETURN.....	19
6.24 CLARIFICATION OF RETURN VALUE FOR AUDITCAPABILITIES OF STRINGS	20
6.25 ERROR RESPONSE WHEN PROCESSING A CONTEXTID	20
6.26 SUPPORT OF PACKAGES	21
6.27 MISMATCH BETWEEN RFC2377 SUPPORT AND ONE "M=" LINE RESTRICTION.....	21
6.28 ANNEX C CODEPOINTS FOR RTCP.....	22
6.29 CLARIFICATION OF PACKAGEID AND NAME FOR ANNEX C	23
6.30 CLARIFICATION OF RESERVEGROUP AND RESERVEVALUE PROPERTIES	23
6.31 CLARIFICATION OF PROVISIONAL RESPONSE TIMER VALUES.....	24
6.32 CLARIFICATION OF NULL CONTEXT USAGE.....	25
6.33 COMMANDS IN SERVICECHANGE ON ROOT TRANSACTION.....	27
6.34 LOOPBACK USAGE CLARIFICATION	27
6.35 ANNEX C AND SDP PARAMETERS	28
6.36 CASE SENSITIVITY OF PROFILE NAMES	29
6.37 PROFILE NEGOTIATION.....	30
6.38 MEDIA TYPE MISMATCH	30
6.39 NOTIFY AVALANCHE.....	31
6.40 TOPOLOGY REPLY	32
6.41 PROTOCOL VERSION NEGOTIATIONS	32
6.42 SERVICESTATES CLARIFICATION FOR CONTINUITY TESTING	33
6.43 CLARIFICATION OF TERMINATION SERVICE STATE UPON RESTART OF MG.....	33
6.44 ALIGNMENT OF TEXT AMONG EVENTS IN THE TONE DETECTION PACKAGE	34
6.45 CLARIFICATION OF PACKAGE DEFINITION REQUIREMENTS FOR ENUMERATIONS.....	35
6.46 CLARIFICATION OF USE OF ABNF ENCODINGS OF OCTET STRINGS	36
6.47 CLARIFICATION OF ENCODING FOR PACKET LOSS STATISTIC IN ANNEX E.12.....	36

Implementors' Guide for Recommendation H.248.1 Version 1 (03/2002)

1 Scope

This guide resolves defects in the following categories:

- editorial errors
- technical errors, such as omissions and inconsistencies
- ambiguities

In addition, the Implementors' Guide may include explanatory text found necessary as a result of interpretation difficulties apparent from the defect reports.

This Guide will not address proposed additions, deletions, or modifications to the Recommendation that are not strictly related to implementation difficulties in the above categories. Proposals for new features should be made through contributions to the ITU-T.

2 Introduction

The H.248.1 Version 1 Implementors' Guide is a compilation of reported defects for version 1 of Recommendation H.248.1 (03/2002). This edition of the Guide contains reported defects identified as of 4/2006.

The Guide must be read in conjunction with Recommendation H.248.1 version 1 to serve as an additional source of information for implementors. For changes to version 2 of H.248.1, please reference the H.248.1 Version 2 Implementors' Guide. For other versions of H.248.1 or for other Recommendations in the H.248.x sub-series, please reference the H.248 Sub-series Implementors' Guide.

3 Defect Resolution Procedure

~~Upon discovering technical defects with Recommendation H.248.1 Version 1, please provide a written description directly to the editor with a copy to the Q.3/16 Rapporteur. The template for a defect report is located at the end of the Guide. Contact information for these parties is included at the front of the document. Return contact information should also be supplied so a dialogue can be established to resolve the matter and an appropriate reply to the defect report can be conveyed. This defect resolution process is open to any interested party. Formal membership in the ITU is not required to participate in this process. Question 3/16 is no longer supporting Version 1 of the H.248.1 specification. Implementors are encouraged to work with one of the more recent versions of the H.248.1 specification.~~

4 References

This document refers to the following Recommendation:

- ITU-T Recommendation H.248.1 Version 1 (03/2002), *Gateway control protocol: Version 1*

5 Nomenclature

In addition to traditional revision marks, the following marks and symbols are used to indicate to the reader how changes to the text of a Recommendation should be applied:

Symbol	Description
<u><i>[Begin Correction]</i></u>	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
<u><i>[End Correction]</i></u>	Identifies the end of revision marked text based on extractions from the published Recommendations affected by the correction being described.
...	Indicates that the portion of the Recommendation between the text appearing before and after this symbol has remained unaffected by the correction being described and has been omitted for brevity.
--- <i>SPECIAL INSTRUCTIONS</i> --- {instructions}	Indicates a set of special editing instructions to be followed.

6 Technical and Editorial Corrections to H.248.1 Version 1 (03/2002)

6.1 Specify types for rtp/jit and rtp/delay in Annex E.12.4

Description:	In Regard to rtp/jit and rtp/delay: These package elements do not have types (Integer, Double, etc.) in the 6/00 doc, the IG, or the corrigendum.
Reference:	Subject: [Megaco] rtp/jit, rtp/delay Date: 6/18/2002 7:22 PM From: Troy Cauble <troy@bell-labs.com

E.12.4 Statistics

[Begin Correction]

Jitter

StatisticID: jit (0x0007)

Requests the current value of the interarrival jitter on an RTP stream as defined in IETF RFC 1889. Jitter measures the variation in interarrival time for RTP data packets.

Type: double

Possible Values: any 64 bit integer

Delay

StatisticID:delay (0x0008)

Requests the current value of packet propagation delay expressed in timestamp units. Same as average latency.

Type: double

Possible Values: any 64 bit integer

[End Correction]

6.2 Define the '#' symbol in INEQUAL in text encoding

Description:	<p>INEQUAL = LWSP (">" / "<" / "#") LWSP</p> <p>The symbol "#" is not explained. From the ASN.1, it appears that it means "not equal". I found a comment in the archives describing it as "quantity of" (??)</p> <p>Which is correct?</p>
Reference:	<p>Subject: [Megaco] INEQUAL, parmValue</p> <p>Date: 6/2/2002 4:21 AM</p> <p>From: Raphael Tryster Raphael@tdsoft.com</p>

B.2 ABNF Specification

[Begin Correction]

INEQUAL = LWSP (">" / "<" / "#") LWSP; "#" means 'not equal'

[End Correction]

6.3 Ambiguous Audit and Individual Audit Return

Description:	<p>The use of the audititem information element in the <i>auditreturnparameter</i> structure leads to ambiguities in audit and individual audit replies. With the current ABNF formulation a reply could be for example:</p> <ul style="list-style-type: none"> - an empty eventsDescriptor or an auditItem - an individual audit response or a descriptor audit response. <p>This ambiguity was introduced because <i>auditreturnparameter</i> is used in both command requests and returns. It was not intended that <i>audititem</i> be used in command replies</p>
Reference:	<p>Subject: RE: [Megaco] Descriptor grammar issue</p> <p>Date: 5/30/2002 1:08 PM</p> <p>From: Aleksandr Ryabin <kengr@winphoria.com></p>

B.2 ABNF Specification

[Begin Correction]

```
signalsDescriptor = SignalsToken [ _LBRKT † signalParm
                               *(COMMA signalParm) † RBRKT_ ]
```

[End Correction]

[Begin Correction]

```
auditReturnParameter = (mediaDescriptor / modemDescriptor /
                        muxDescriptor / eventsDescriptor /
                        signalsDescriptor / digitMapDescriptor /
                        observedEventsDescriptor /
                        eventBufferDescriptor /
                        statisticsDescriptor / packagesDescriptor /
                        errorDescriptor / auditReturnItem)
auditReturnItem = (MuxToken / ModemToken / MediaToken /
                  DigitMapToken / StatsToken /
                  ObservedEventsToken / PackagesToken )
```

[End Correction]

[Begin Correction]

;at-most-once, and DigitMapToken and PackagesToken are not allowed
;in AuditCapabilities command

```
auditItem = ( auditReturnItem / SignalsToken /
              EventBufferToken / EventsToken )
auditItem = ( MuxToken / ModemToken / MediaToken /
              SignalsToken / EventBufferToken /
              DigitMapToken / StatsToken / EventsToken /
              ObservedEventsToken / PackagesToken )
```

[End Correction]

6.4 Define the symbol for NULL context in text encoding

Description:	In answering a recent question on the list, I was surprised to not be able to find where we define NULL as being encoded as '-' in text encoding. We define "*" and "\$" in B.1. "ROOT" appears in the syntax for TerminationID. Similarly '-' appears in the syntax for ContextID but nothing states that it stands for NULL. It seems obvious "ROOT" stands for ROOT but less so for '-'. Am I missing something or should be add something to B.2 to state this?
---------------------	---

B.2 ABNF Specification

[Begin Correction]

```
;The values 0x0, 0xFFFFFFFFE and 0xFFFFFFFFF are reserved,
; "-" is used for NULL context.
ContextID          = (UINT32 / "*" / "-" / "$")
```

[End Correction]

6.5 Corrections to Appendix A example statistics

Description:	The statistics returned in step 22 of the example in Appendix A.1.1 omits some of the statistics in the packages implemented on the terminations and indicates the wrong units for nt/dur. While this example is not normative, errors are confusing to readers.
Reference:	Private discussion during Jun 2002 meeting in Bruges.

Appendix A.1.1 step 22

[Begin Correction]

```
From MG2 to MGC:

MEGACO/1 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
    Subtract = A5555 {
      Statistics {
        nt/os=45123, ; Octets Sent
        nt/or=45123, ; Octets SentReceived
        nt/dur=40000 ; in milliseconds
      }
    },
    Subtract = A5556 {
      Statistics {
        rtp/ps=1245, ; packets sent
        nt/os=62345, ; octets sent
        rtp/pr=780, ; packets received
        nt/or=45123, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=27,
        rtp/delay=48 ; average latency
        nt/dur=38000 ; in millisec
      }
    }
  }
}
```

[End Correction]

6.6 Corrections to Package Guidelines for Statistics in 12.1.5

Description:	The guidelines for defining statistics for packages only suggests indicating the units of the statistic but not its type or range. The packages in Annex E that define statistics uses sections similar to those for package parameters which include type and possible values. It would seem preferable to change the guidelines to recommend this format.
Reference:	Private discussion during Jun 2002 meeting in Bruges.

12.1.5 Statistics

[Begin Correction]

Statistics defined by the package, specifying:

Statistic name: only descriptive.

StatisticID: Is an identifier

StatisticID is used in a StatisticsDescriptor

DescriptionUnits: unit of measure, e.g. milliseconds, packets

Type: One of:

_____ Boolean

_____ String: UTF-8 string

_____ Octet String: A number of octets. See Annex A and Annex B.3 for encoding

_____ Integer: 4 byte signed integer

_____ Double: 8 byte signed integer

_____ Character: Unicode UTF-8 encoding of a single letter.

_____ _____ Could be more than one octet.

_____ Enumeration: One of a list of possible unique values (See 12.3)

_____ Sub-list: A list of several values from a list. The type of sub-list SHALL also be specified. The type shall be chosen from the types specified in this section (with the exception of sub-list). For example, Type: sub-list of enumeration. The encoding of sub-lists is specified in Annexes A and B.3.

_____ Boolean

Possible Values:

A package must indicate the unit of measure, e.g. milliseconds, packets, either here or along with the type above, as well as indicating any restriction on the range.

[End Correction]

6.7 Specification of the meaning of automatic in E.13 tdm package

Description:	The meaning of “automatic” in the gain parameter of E.13 tdm package is not well defined.
---------------------	---

	<p>and</p> <p>There have been several issues raised over the last month or so about gain in the TDM package (Annex E.13) as well as in Annex C. I will summarize the issues and propose changes to address them.</p> <p>1. tdm/gain is defined as integer which in 12.1.2 is clearly signed but the description and choice of value for "automatic" seem to have assumed that the value was unsigned. [consensus was that gain should be signed, negative values have useful meaning and that the reserved value for automatic should be changed. Note this last issue is definitely not backward compatible but no objections were raised.]</p> <p>2. tdm/gain does not specify if it applies to outbound signal level, inbound or both. [consensus was that it should be for outbound signal level.]</p> <p>3. Nigel Williams suggested that the value in text encoding be restricted to decimal (non-hex) representation for easier parsing. [there was no discussion on this, but since the comment the in B.2 specifies that either decimal and hexadecimal can be used for positive values of any integer property, I think we should NOT make this a special case. Note that the specification does require decimal for negative values.]</p> <p>4. There is also a Gain in C.1 (100C) for binary encoding. It is not clear what it applies to and it is defined as unsigned integer and 0..65535 (evidently 2 bytes). [Consensus was that this should be deprecated similar to what we did for echo control.]</p>
Reference:	<p>AVD-2191 a liaison from SG15 received at the Jun 2002, Bruges meeting and further changes from:</p> <p>Subject: [Megaco] Gain in TDM package and in Annex C</p> <p>Date: 8/1/2002 12:24 PM</p> <p>From: Terry L Anderson tla@lucent.com</p>

C.1 General media attributes

[Begin Correction]

Gain	100C	Unsigned integer	Gain in dB: 0..65535 Not Used. See H.248.1 Annex E.13 for an available gain property.
------	------	-----------------------------	--

[End Correction]

E.13.1 Properties

[Begin Correction]

Gain Control

PropertyID: gain (0x000a)

Gain control, or usage of ~~ef~~ signal level adaptation and noise level reduction is used to adapt the level of the outbound signal. However, it is necessary, for example for modem calls, to turn off this function. When the value is set to "automatic", the termination serves as an automatic level control (ALC) with a target level provisioned on the MG and the direction being outward.

Type: integer

Possible values:

The gain control specifies the gain in decibels (positive or negative), with the maximum positive integer, 214748646 (0x7fffffff), reserved to represent "automatic". ~~parameter may either be specified as "automatic" (0xffffffff), or as an explicit number of decibels of gain (any other integer value).~~ The default value is provisioned in the MG.

Defined in: LocalControlDescriptor

Characteristics: read/write

[End Correction]

6.8 Protocol Version Negotiation

Description:	<p>Section 11.3 on protocol negotiation, describes the behavior of MGC specifying explicitly that it return version in the response for two cases:</p> <ol style="list-style-type: none"> 1) MGC supports only a lower version than that proposed by MG 2) MGC supports a higher version but can support the version proposed by MG. <p>It does not explicitly state behavior for the case:</p> <ol style="list-style-type: none"> 3) MGC supports ONLY the same version proposed by MG but this is not actually stated. MG of course could not know the difference between #2 and #3. I suppose that an MG receiving a response with no version should assume that the version proposed is being accepted but of course this would have worked for #2 as well. The question is, is omission equivalent to returning the equal value. <p>Description should be changed to cover this case and to indicate that return of version is optional if MG's choice is used.</p>
Reference:	<p>Subject: [Megaco] Protocol version negotiation Date: 7/23/2002 10:27 PM From: Terry L Anderson tla@lucent.com</p>

11.3 Negotiation of protocol version

[Begin Correction]

A ServiceChange command from a MG that registers with an MGC shall contain the version number of the protocol supported by the MG in the ServiceChangeVersion parameter. Regardless of the version placed in the ServiceChangeVersion parameter the message containing the command shall be encoded as a version 1 message. Upon receiving such a message, if the MGC supports only a lower version, then the MGC shall send a ServiceChangeReply with the lower version and thereafter all the messages between MG and MGC shall conform to the lower version of the protocol. If the MG is unable to comply and it has established a transport connection to the MGC, it should close that connection. In any event, it should reject all subsequent requests from the MGC with Error 406 – Version Not Supported.

If the MGC only supports higher version(s) than the MG, it shall reject the association with Error 406 Version Not Supported.

If the MGC supports the version indicated by the MG, it shall conform to that version in all subsequent messages. In this case it is optional for the MGC to return a version in the ServiceChangeReply.

~~If the MGC supports a higher version than the MG but is able to support the lower version proposed by the MG, it shall send a ServiceChangeReply with the lower version and thereafter all the messages between MG and MGC shall conform to the lower version of the protocol. If the MGC is unable to comply, it shall reject the association, with Error 406 – Version Not Supported.~~

[End Correction]

6.9 Statistics and Move

Description:	<p>Statistics on a termination are defined in two ways:</p> <ol style="list-style-type: none">6.2 says: "Statistics are reported...when the termination is taken out of the call it is in."7.1.15 says: "The Statistics parameter provides information describing the status and usage of a Termination during its existence with a specific Context." <p>I don't think a "call" is a well defined concept in H.248, but perhaps these two statements are consistent if we associate a "call" with a context. The issue is, if a termination is Moved to a different termination, is it in the same call (one party is probably the same but the other may not be)? Or, more specifically are its statistics zeroed.</p> <p>Subtract clearly zeros them since ephemerals disappear and physicals get default values when returned to NULL context, and we state this as the last sentence in 7.1.15. The issue is for a Move. We never clearly state this for Move, but from the current language defining what statistics mean, I assume that they ARE zeroed. It would help if that last sentence in 7.1.15 included Move (or if they are NOT cleared, we state that).</p> <p>We report statistics, by default, on a Subtract, but NOT on a Move. So if statistics are zeroed by the move, then one must remember to add a Statistics Descriptor to Move to get them where this is not needed for a Subtract. It would seem to be consistent we would have reported statistics by default on all commands that zero them rather than only on Subtract.</p>
Reference:	<p>Subject: [Megaco] Statistics and Move Date: 3/7/2002 4:23 PM From: Terry L Anderson tla@lucent.com</p>

6.2 Terminations [H.248.1v1 (03/02)]

[Begin Correction]

Terminations may have signals applied to them (see 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 7.2.5) and when the Termination ceases to exist or is returned to the null context due to a Subtract command~~is subtracted from a context.~~

[End Correction]

7.1.15 Statistics descriptor

[Begin Correction]

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)~~within a specific Context~~. There is a set of standard statistics kept for each Termination where appropriate (number of octets sent and received for example). The particular statistical properties that are reported for a given Termination are determined by the Packages realized by the Termination. By default, statistics are reported when the Termination ceases to exist or is returned to the null context due to a Subtract command~~is subtracted from the Context~~. This behaviour can be overridden by including an empty AuditDescriptor in the Subtract

command. 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. Statistics are reset when a ceases to exist or is returned to the null context due to a Subtract command ~~Termination is Subtracted from a Context.~~

[End Correction]

E.11.4 Statistics

[Begin Correction]

Duration

StatisticsID: dur (0x0001)

Description: provides duration of time the termination has existed or been out of the null context ~~been in the Context.~~

Type: double, in milliseconds

[End Correction]

6.10 Additional Codepoint for Annex C

Description:	H.248.20 needs an additional code point in Annex C and should be available in both v1 and v2 of H.248.1
Reference:	D-280 from Geneva 10/2002

C.12 H.245

[Begin Correction]

PropertyID	Property tag	Type	Value
OLC	C001	Octet string	The value of H.245 OpenLogicalChannel structure. Ref.: ITU-T H.245
OLCack	C002	Octet string	The value of H.245 OpenLogicalChannelAck structure. Ref.: ITU-T H.245
OLCcnf	C003	Octet string	The value of H.245 OpenLogicalChannelConfirm structure. Ref.: ITU-T H.245
OLCrej	C004	Octet string	The value of H.245 OpenLogicalChannelReject structure. Ref.: ITU-T H.245
CLC	C005	Octet string	The value of H.245 CloseLogicalChannel structure. Ref.: ITU-T H.245
CLCack	C006	Octet string	The value of H.245 CloseLogicalChannelAck

PropertyID	Property tag	Type	Value
			structure. Ref.: ITU-T H.245
<u>LCN</u>	<u>C007</u>	<u>Integer</u>	<u>The value of H.245 Local Channel Number</u> <u>0 - 65535.</u> <u>Ref.: ITU-T H.245</u>

[End Correction]

6.11 Wild Carding Principles

Description:	<p>Over the last several months on the IETF Megaco list the issue of wild carded context command handling has been raised several times. There has been a significant amount of discussion surrounding this topic and it is apparent that the H.248.1 recommendation is unclear with regards to the handling of wildcarded commands.</p> <p>From the discussions on the list the main issues that need to be clarified are:</p> <ul style="list-style-type: none"> • All commands should have the same behavior with respect to wildcarding • Context=ALL does NOT cover the NULL context. • The reply to commands using a wildcarded Context=ALL and partial Termination ID should not contain an error if at least one of the Full TerminationIDs referenced by the partial TerminationID appear in a context. <p>The intent is to clarify the use of wildcard not change the specification, but since the current text has many ambiguities the clarification will not match all current implementation.</p> <p>April 2006: This item has been updated to correct an error in the selection of error code for a particular wildcarding situation.</p>
Reference:	<p>D-283 to the Geneva 10/2002 meeting and discussions at the meeting.</p> <p>D-223 to the Geneva 04/2006 meeting and discussions at the meeting.</p>

[Begin Correction]

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 a MGC instructs a MG to choose a circuit within a trunk group.

~~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. 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 a 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 response may be particularly useful in the Audit commands.

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

[End Correction]

[Begin Correction]

6.63 Wildcarding Principles

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

1. Context Wildcarding
2. Termination Wildcarding

When executing a transaction that contains wildcarded contexts and optionally 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 an error code 431 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.

6.36.1 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 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 4 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}}

6.36.2 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. 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}}

6.36.3 ContextID wildcarded (ALL) with TerminationID wildcarded

In the case where the ContextID is wildcarded (i.e. Context ID = 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 431 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 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.

6.36.4 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 a 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 response 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 termination/s 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 transactionID 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.

[End Correction]

6.12 Wildcarding in the Topology Descriptor

Description:	During discussions on the Megaco Mailing list an ambiguity was recognized in the text describing how wildcarding works in the case of a one way topology descriptor. It was thought that a "*" wildcard was not allowed to be used in the topology triple (T1,T2,one-way). This is incorrect as a "*" wildcard may be used in one termination ID but not both.
Reference:	Subject: RE: [Megaco] Question on Section 7.1.18 Megaco V2 Date: Tue, 14 Jan 2003 13:31:54 -0500 From: "Kevin Boyle" <kboyle@nortelnetworks.com>

[Begin Correction]

7.1.18 Topology Descriptor

- (T1, T2, one-way) 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 both T1 and T2 but not both is not allowed.

[End Correction]

6.13 Binary Value for Packetization Time (Annex C)

Description:	During discussion on the Megaco Mailing list it was highlighted that there was no equivalent binary codepoint for the SDP a=ptime. It was felt that a new Annex C property should be defined.
Reference:	Subject: [Megaco] Binary equivalent for SDP ptime attribute Date: Mon, 20 Jan 2003 15:16:25 -0600 From: John_Poplett@3com.com

[Begin Correction]

C.1 General media attribute

RTPpayload	100F	Integer	Payload type in RTP Profile for Audio and Video Conferences with Minimal Control
------------	------	---------	--

			Ref.: RFC 1890
<u>Ptime</u>	<u>1010</u>	<u>Integer</u>	<packet Packetization Time This gives the length of time in milliseconds represented by the media in a packet. Ref.: RFC 2327

[End Correction]

6.14 Modification of Terminations by MGCs

Description:	During discussion on the Megaco Mailing list, it was highlighted that the spec prohibited modification of terms in the NULL context. This clearly was not the intent of the passage, and many implementations allow this very function.
Reference:	Subject: [Megaco] Restriction on Modification Of Terminations Date: Tue, 15 Apr 2003 22:00:09 -0400 From: Christian.Groves@ericsson.com.au

[Begin Correction]

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 it has have been previously seized via, e.g. the Add command.

[End Correction]

6.15 Optional Commands in an Action

Description:	There is ambiguity in section 8 on whether or not commands are optional in transaction requests. A command is optional if a context property modification or audit is included.
Reference:	D-347 to the 05/03 Geneva meeting and discussions at the meeting

[Begin Correction]

8.2.1 TransactionRequest

The TransactionRequest is invoked by the sender. There is one Transaction per request invocation. A request contains one or more Actions, each of which specifies its target Context ~~and one or more Commands per Context.~~

...

[End Correction]

6.16 Ordering of Transactions

Description:	Transaction processing order when multiple transactions are contained in a message is not guaranteed. This behavior can generate errors in scenarios where there was successful command execution on the MG. For example: the MGC may receive a message with a transaction reply containing the results of a command ADD(ContextID=?, TerminationID=?) and a transaction request containing a notification from the contextID and TerminationID in the transaction reply. If the MGC chooses to execute the second transaction request first it will generate an error 410 "Incorrect identifier" although the transaction reply contained the ContextID and Termination ID. This error can be avoided if transaction replies are processed before transaction requests.
Reference:	D-347 to the 05/03 Geneva meeting and discussions at the meeting

[Begin Correction]

8 Transactions

...

Transactions guarantee ordered Command processing. That is, Commands within a Transaction are executed sequentially. Ordering of Transactions is NOT guaranteed – transactions may be executed in any order, or simultaneously however transaction replies should be executed before transaction requests when both are contained in a message.

...

[End Correction]

6.17 Replies to Actions with no Commands

NOTE – This item has been deprecated in favor of item 6.23.

Description:	The ABNF commandReply construct requires replies to actions without commands to include the contextProperties specified in the action. This is different from the ASN.1, which does not require this and different from commands in that commandReplies are not required to return fully specified descriptors set by the command. This is because the ABNF puts the contextProperties construct in the commandReply construct, rather in the actionReply construct. When H.248.1 V3 is produced, then consideration should be given to moving the contextProperties construct up to the actionReply level.
Reference:	Discussions at the 05/03 meeting

[Begin Correction]

7.2.10 Generic Command Syntax

...

A complete ABNF of the text encoding of the protocol per IETF RFC 2234 is given in Annex B. SDP is used as the encoding of the Local and Remote descriptors for use with the text encoding as modified in 7.1.8.

Note: The ABNF is different from the ASN.1 in the case of a reply to an action with no commands. The ABNF requires the contextProperties to be included in the reply even when fully specified in the request, while the ASN.1 does not.

[End Correction]

6.18 Network Package can apply to TDM

Description:	There has been some confusion about whether the network package can apply to TDM. The intent is that the package can apply to any network. Some clarifying text to this end is needed.
Reference:	Discussions at the 09/03 meeting

[Begin Correction]

E.11 Network Package

PackageID: nt (0x000b)

Version: 1

Extends: None

This package defines properties of network terminations independent of network type. This includes, but is not limited to, TDM, IP and ATM.

[End Correction]

6.19 Precedence of LocalControl Mode property versus SDP mode

Description:	Discussions on the Megaco mailing list highlighted that there is no text indicating how the LocalControl Mode property interacts with mode as specified in SDP.
Reference:	Subject: Re: [Megaco] SDP mode and Megaco mode question Date: Mon, 08 Dec 2003 15:55:41 +1100 From: Christian Groves <Christian.Groves@ericsson.com>

7.1.7 LocalControl descriptor

[Begin Correction]

...

The allowed values for the mode property are send-only, receive-only, send/receive, inactive and loop-back. "Send" and "receive" 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. The default value for the mode property is "Inactive". Signals and Events are not affected by mode. The LocalControl Mode property takes precedence over any mode specified in the Local and Remote descriptors.

[End Correction]

6.20 Digit processing clarification

Description:	Discussion on the Megaco mailing list revealed that the wording in both H.248.1 and H.248.16 regarding "first digit" processing is misleading.
Reference:	Subject: [Megaco] DTMF detection and buffering

Date: Wed, 27 Aug 2003 19:18:53 +0200
From: "CHATRAS Bruno FTRD/DAC/ISS" <bruno.chatras@francetelecom.com>

[Begin Correction]

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 digit map~~~~having been dialed~~. 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.

[End Correction]

6.21 Usage of DigitMap timer symbols with range notation

Description:	Discussion on the Megaco mailing list has concluded that clarification is needed on the usage of timer symbols when building digitmap strings utilizing the range notation.
Reference:	Subject: RE: [Megaco] Digit Maps: Long Duration Modifier(Z), Long Timers (L), & Short Timers (S) Date: Fri, 5 Dec 2003 13:09:11 -0500 From: Steve Cipolli <SCipolli@radvision.com>

7.1.14.3 DigitMap Syntax

[Begin Correction]

...

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. If the Z specifier is not followed by a digit (0-9 or A-K), then the MG shall reject the digitmap as invalid protocol. 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).

[End Correction]

6.22 Clarification of the use of StreamID = 0

Description:	The ASN.1 and ABNF version of the H.248.1 syntax allow StreamIDs 0 to 65535. The convention in the procedure section of H.248 is that Stream ID starts with stream 1. This needs to be further clarified.
Reference:	COM 16 D-376

7.1.4 Media descriptor

[Begin Correction]

...

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 three subsidiary descriptors: LocalControl, Local, and Remote. The relationship between these descriptors is thus:

[End Correction]

6.23 Correction of Context Audit return

Description:	The ABNF commandReply construct requires replies to actions without commands to include the contextProperties specified in the action. This is different from the ASN.1, which does not require this and different from commands in that commandReplies are not required to return fully-specified descriptors set by the command. This is because the ABNF puts the contextProperties construct in the commandReply construct, rather in the actionReply construct.
Reference:	COM 16 D-374
NOTE – This item supersedes item 6.17	

B.2 ABNF specification

[Begin Correction]

...

```
; at-most-once
; EmergencyOffToken to be used in MG to MGC direction only in H.248.1 v1
; either EmergencyToken or EmergencyOffToken, but not both
contextProperty      = (topologyDescriptor / priority / EmergencyToken /
EmergencyOffToken)
```

...

```
EmergencyToken      = ("Emergency"           / "EG")
EmergencyOffToken = ("EmergencyOff"         / "EGO")
ErrorToken          = ("Error"               / "ER")
```

...

[End Correction]

6.24 Clarification of return value for AuditCapabilities of strings

Description:	H.248.1 does not specify what should be returned by the MG if the MGC requests an AuditCapabilities of a property that is a string. Since the possibilities are infinite, H.248.1 should specify a value to be returned.
Reference:	Discussions at the 01/2004 Geneva meeting

7.2.6 AuditCapabilities

[Begin Correction]

...

Interpretation of what capabilities are requested for various values of ContextID and TerminationID is the same as in AuditValue.

For property and parameter values of type string, character or octet string, the MG shall return an empty value. For the text encoding, strings and characters return an empty quotedString construct, while octet strings return NUL (0x00). This behavior may be overridden by the package definition.

...

[End Correction]

6.25 Error response when processing a ContextID

Description:	H.248.1 § 8.2.2 discusses the action to be taken when a receiver encounters an error parsing a ContextID. It specifies that the ContextID is returned in an Action response with error code 422 "Syntax Error in action". It is very likely however that if a receiver cannot parse the context ID it will not be able to return the ContextID back to the sender. It is proposed below to allow sending of a response at a transaction level without including the ContextID.
Reference:	AVD-2467

[Begin Correction]

8.2.2 TransactionReply

...

If the receiver encounters an error in processing a ContextID but can parse the ContextID, the requested Action response will consist of the Context ID and a single error descriptor, 422 (Syntax Error in Action). If the receiver cannot parse the ContextID, it shall return a TransactionReply consisting of the TransactionID and a single error descriptor, 422 (Syntax Error in Action).

...

[End Correction]

6.26 Support of packages

Description:	There has been some confusion of what is meant by the term “To support a particular package the MG must support all properties, signals, events and statistics defined in a package. It must also support all Signal and Event parameters.” The term "Must Support" is different from "it is mandatory to implement all the functionality in the package." This term was added so that the MG had to support the reception of properties, signal, events and statistics. The MG must not return a syntax error or unknown ID for any of these elements. However the MG could give a meaningful response such as 501 not implemented. This shows that the MG has at least considered the complete package. It is encouraged to support the complete functionality of a package however it is better to return error code 501 and using an existing package rather than creating new packages with the same functionality as contained in existing packages. It is proposed to clarify this.
Reference:	Subject: RE: [Megaco] Support of a H.248 package Date: Tue, 23 Mar 2004 14:39:43 -0500 From: Kevin Boyle <kboyle@nortelnetworks.com> and AVD-2467

[Begin Correction]

6.2.3 Packages

...

To support a particular package the MG must ~~support~~ recognize all properties, signals, events and statistics defined in a 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”.

...

[End Correction]

6.27 Mismatch between RFC2377 support and one “m=” line restriction

Description:	H.248.1 § 7.1.8 Local and Remote descriptor mandates that there shall only be one m= line per SDP. However further in the section it states that “Implementations shall accept session descriptions that are fully conformant to RFC 2327.” There could be some confusion over whether or not multiple m=lines should be accepted if they conform to RFC2327. Multiple m=lines per SDP should not be accepted by the MG. This should be clarified.
Reference:	AVD-2467

[Begin Correction]

7.1.8 Local and Remote descriptors

...

A Stream Descriptor specifies a single bidirectional media stream and so a single session description MUST NOT include more than one media description ("m=" line). 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. 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.

...

[End Correction]

6.28 Annex C codepoints for RTCP

Description:	H.248.30 introduces a package for the support of RTCP XR. As part of the functioning of RTCP SDP bandwidth modifiers may be sent. This is currently supported in SDP but not in the Binary version of H.248.1. Thus the bandwidth modifiers RTCP(b=RS:xx, b=RR:xx) as defined in RFC3556 need to be added to H.248.1 Annex C.
Reference:	Subject: [Megaco] RTCP bandwidth modifiers Date: Thursday, March 25, 2004 5:41 AM From: Kalleitner Franz [mailto:franz.kalleitner@siemens.com]

[Begin Correction]

6.28.1.1 C.6 IP

PropertyID	Property tag	Type	Value
IPv4	6001	32 bits Ipv4Address	Ipv4Address Ref.: IETF RFC 791
IPv6	6002	128 bits	IPv6 Address Ref.: IETF RFC 2460
Port	6003	Unsigned integer	0..65535
Porttype	6004	Enumerated	TCP(0), UDP(1), SCTP(2)
<u>RtcpbwRS</u>	<u>6005</u>	<u>Integer</u>	<u>RS RTCP bandwidth modifier indicates the RTCP bandwidth allocated to active data senders (as defined by the RTP spec)</u> Ref.: IETF RFC 3556
<u>RtcpbwRR</u>	<u>6006</u>	<u>Integer</u>	<u>RR RTCP bandwidth modifier indicates the RTCP bandwidth allocated to other participants in the RTP session (i.e., receivers)</u> Ref.: IETF RFC 3556

[End Correction]

6.29 Clarification of PackageID and name for Annex C

Description:	The binary encoding of H.248.1 uses Annex C for a number of local and remote properties. The identities of the Annex C properties use the normal package name/property id construct. It is not clear from reading Annex C what the Package name is. The Annex A ASN.1 syntax must be read to find out that the binary package name is H'0000. No text package name has been defined for Annex C for use with the H.248 property SDP element. Annex C should be modified to make the package name/id clear.
Reference:	----- Original Message ----- Subject: Re: Signaling of UDI rsp Clearmode Bearer Service; Re: TDM Hairpinning; Re: [Megaco] Hairpin case A/u Law conversion Date: Tue, 27 Apr 2004 13:55:12 +0200 From: Carsten Waitzmann <cwaitzmann@alcatel.de>

[Begin Correction]

Annex C Tags for media stream properties

Parameters for Local, Remote and LocalControl descriptors are specified as tag-value pairs if binary encoding is used for the protocol. This annex contains the property names (PropertyID), the tags (Property tag), type of the property (Type) and the values (Value). Values presented in the Value field when the field contains references shall be regarded as "information". The reference contains the normative values. If a value field does not contain a reference, then the values in that field can be considered as "normative".

The referencing of Annex C properties follows the PackageID/PropertyID structure; however Annex C is not in itself a package. Annex C is considered to have PackageID 0x0000 for binary encoding and "anxc" for text encoding. For text encoding of H.248.1, Annex C shall only be used in the case that the required property is not already defined by a package or represented by SDP. The nesting of one Annex C property inside another is forbidden.

Tags are given as hexadecimal numbers in this annex. When setting the value of a property, a MGC may underspecify the value according to one of the mechanisms specified in 7.1.1.

...

[End Correction]

6.30 Clarification of ReserveGroup and ReserveValue Properties

Description:	The use of the Reserve property is described in H.248.1 sections 7.1.7 and 7.1.8. Currently there is a mismatch between the stated behaviour of Reserve Group. Section 7.1.7 states: "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." however section 7.1.8 states "If ReserveGroup is True, the MG reserves the resources required to support any of the requested property group alternatives that it can currently support. If ReserveValue is True, the MG reserves the resources required to support any of the requested property value alternatives that it can currently support." It is proposed to update 7.1.8 to align with the definition in 7.1.7.
Reference:	AVD-2569

[Begin Correction]

7.1.8 Local and remote descriptors

...

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

...

[End Correction]

6.31 Clarification of Provisional Response Timer Values

Description:	<p>H.248.1 has 2 sections that discuss the setting of the (MG)provisionalResponseTimer value. Section D.1.4 states: “The root Termination has a property (ProvisionalResponseTimerValue), which can be set to the requested maximum number of milliseconds between receipt of a command and transmission of the TransactionPending response.”</p> <p>Section E.2.1 states: “Initially set to normalMGExecutionTime plus network delay, but may be lowered.”\</p> <p>These two sections give conflicting advice of what to set the timer to, D.1.4 does not take into account network delay which may result in messages being repeated unnecessary. It is proposed to remove the text from D.1.4 and rely on the definition in E.2.1.</p> <p>Furthermore section 8.2.3 discusses procedures for the sending of Transaction Pending. It however has omitted the (MG/MGC)ProvisionalResponseTimerValue the use of which should be documented.</p>
Reference:	AVD-2569

[Begin Correction]

8.2.3 TransactionPending

...

TransactionPending(TransactionID { })

The TransactionID parameter must be the same as that of the corresponding TransactionRequest. A property of root (normalMGExecutionTime) is settable by the MGC to indicate the interval within which the MGC expects a response to any transaction from the MG (exclusive of network delay). Another property (normalMGCEExecutionTime) is settable by the MGC to indicate the interval within which the MG should expect a response to any transaction from the MGC (exclusive of network delay). MGProvisionalResponseTimerValue indicates the time within which the MGC should expect a Pending Response from the MG if a Transaction cannot be completed (initially set to normalMGExecutionTime plus network delay, but may be lowered). MGCProvisionalResponseTimerValue has the corresponding meaning to the MG. Senders may receive more than one TransactionPending for a command. If a duplicate request is received when pending, the responder may send a duplicate pending immediately, or continue waiting for its timer to trigger another TransactionPending.

...

[End Correction]

[Begin Correction]

D.1.4 Provisional responses

Executing some transactions may require a long time. Long execution times may interact with the timer-based retransmission procedure. This may result either in an inordinate number of retransmissions, or in timer values that become too long to be efficient. Entities that can predict that a transaction will require a long execution time may send a provisional response, "Transaction Pending". They SHOULD send this response if they receive a repetition of a transaction that is still being executed.

Entities that receive a Transaction Pending shall switch to a different repetition timer for repeating requests. ~~The root Termination has a property (ProvisionalResponseTimerValue), which can be set to the requested maximum number of milliseconds between receipt of a command and transmission of the TransactionPending response.~~ Upon receipt of a final response following receipt of provisional responses, an immediate confirmation shall be sent, and normal repetition timers shall be used thereafter. An entity that sends a provisional response, SHALL include the immAckRequired field in the ensuing final response, indicating that an immediate confirmation is expected. Receipt of a Transaction Pending after receipt of a reply shall be ignored.

...

[End Correction]

6.32 Clarification of NULL Context Usage

Description:	<p>There has been some confusion over what terminations in the NULL context actually represent. H.248.1 § 6. Connection Model gives some guidance:</p> <p>“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 associated to any other Termination. For instance, in a decomposed access gateway, all idle lines are represented by Terminations in the null Context.”</p> <p>Terminations in the NULL context typically have default descriptor values associated with them. For example: the termination may have an event detecting “off-hook”. Therefore according to the above text it could be stated that terminations with default descriptor values could be considered as IDLE lines.</p> <p>H.248.1 § 7.2.3 Subtract gives further guidance on what happens to property values when a termination is SUBTRACTED back to the NULL context:</p> <p>“When a provisioned Termination is Subtracted from a Context, its property values shall revert to:</p> <ul style="list-style-type: none">• the default value, if specified for the property and not overridden by provisioning;• otherwise, the provisioned value.” <p>It is the contributor’s contention that the term “property values” is a generic term and what is really meant is “descriptor values”. Therefore the conclusion that when a termination is subtracted back to the NULL context this represents an IDLE line.</p>
---------------------	--

Reference:	AVD-2570
-------------------	----------

[Begin Correction]

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 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 Termination state 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 PackageName 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.

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 behavior. For more details, see 7.1 dealing with the individual Descriptors.

The above behavior applies equally to Signals, Events and their parameters.

...

[End Correction]

[Begin Correction]

7.2.3 Subtract

...

When a provisioned Termination is Subtracted from a Context, its ~~property-descriptor~~ values shall revert to:

- the default value, if specified for the ~~property-descriptor~~ and not overridden by provisioning;
- otherwise, the provisioned value.

...

[End Correction]

6.33 Commands in ServiceChange on root transaction

Description:	H.248.1 allows multiple commands to be grouped per Transaction. An exception to this is a transaction containing a ServiceChange command specifying the "Root" terminationID and ServiceChangeMethod equal to Restart or Failover. This is because subsequent transactions shall use any ServiceChangeAddress supplied in the transaction response and the negotiated protocol version. This is already implied in the H.248.1 text however the prevention of multiple commands in this scenario is not explicit and may lead to confusion. It is proposed to make this explicit in H.248.1.
Reference:	COM 16 D-44

[Begin Correction]

7.2.8 ServiceChange

...

A ServiceChange Command specifying the "Root" for the TerminationID and ServiceChangeMethod equal to Restart is a registration command by which a Media Gateway announces its existence to the Media Gateway Controller. The Media Gateway may also announce a registration command by specifying the "Root" for the TerminationID and ServiceChangeMethod equal to Failover when the MG detects MGC failures. The Media Gateway is expected to be provisioned with the name of one primary and optionally some number of alternate Media Gateway Controllers. Acknowledgement of the ServiceChange Command completes the registration process, except when the MGC has returned an alternative ServiceChangeMgcId as described in the following paragraph. The MG may specify the transport ServiceChangeAddress to be used by the MGC for sending messages in the ServiceChangeAddress parameter in the input ServiceChangeDescriptor. The MG may specify an address in the ServiceChangeAddress parameter of the ServiceChange request, and the MGC may also do so in the ServiceChange reply. In either case, the recipient must use the supplied address as the destination for all subsequent transaction requests within the association. At the same time, as indicated in clause 9, transaction replies and pending indications must be sent to the address from which the corresponding requests originated. This must be done even if it implies extra messaging because commands and responses cannot be packed together. The TimeStamp parameter shall be sent with a registration command and its response. A message containing a ServiceChange Command specifying "Root" for the TerminationID and a ServiceChangeMethod equal to Restart or Failover shall not contain other commands as these commands should use the new ServiceChangeAddress and negotiated protocol version.

...

[End Correction]

6.34 Loopback usage clarification

Description:	There are 4 stream mode properties in H.248.1: send, receive, inactive and loopback. Send, receive and inactive are described by section 7.1.7 however H.248.1 is largely silent on the operation of loopback. The use of loopback should be clarified.
Reference:	COM 16 D-44

[Begin Correction]

7.1.7 LocalControl Descriptor

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 underspecified as in 7.1.1.

The allowed values for the mode property are send-only, receive-only, send/receive, inactive and loop-back. "Send", ~~and~~ "receive" 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 "loop-back" on a termination, media received (local descriptor) on that 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 mode.

[End Correction]

6.35 Annex C and SDP parameters

Description:	<p>H.248.1 Annex C.11 allows the use of SDP equivalents for use in binary implementations of the protocol. One issue that is not clear is whether the SDP is 100% RFC2327 compliant or is subject to the exceptions of H.248.1 section 7.1.8 "Local and Remote Descriptor".</p> <p>3GPP in their technical specification TS29.332 "Media Gateway Control Function (MGCF) – IM Media Gateway; Mn Interface", V6.0.0 (2004-09) make use of SDP equivalents. The note to Table 10.1: "required parameters" makes the assumption that the exceptions of H.248.1 apply.</p> <p>It is believed that 3GPP2 also make this assumption.</p> <p>It is proposed to clarify the text in H.248.1 Annex C.1 that the SDP is subject to the behavior of H.248.1 section 7.1.8.</p>
Reference:	AVD-2663

[Begin Correction]

C.11 SDP equivalents

The SDP equivalents are subject to the SDP exceptions of 7.1.8 described for text encoding of the protocol. Also the CHOOSE wildcard is used as in text encoding of the protocol.

PropertyID	Property tag	Type	Value
SDP_V	B001	String	Protocol Version Ref.: RFC 2327
SDP_O	B002	String	Owner/creator and session ID Ref.: RFC 2327
SDP_S	B003	String	Session name Ref.: RFC 2327
SDP_I	B004	String	Session identifier Ref.: RFC 2327
SDP_U	B005	String	URI of descriptor Ref.: RFC 2327

PropertyID	Property tag	Type	Value
SDC_E	B006	String	email address Ref.: RFC 2327
SDP_P	B007	String	phone number Ref.: RFC 2327
SDP_C	B008	String	Connection information Ref.: RFC 2327
SDP_B	B009	String	Bandwidth Information Ref.: RFC 2327
SDP_Z	B00A	String	Time zone adjustment Ref.: RFC 2327
SDP_K	B00B	String	Encryption Key Ref.: RFC 2327
SDP_A	B00C	String	Zero or more session attributes Ref.: RFC 2327
SDP_T	B00D	String	Active Session Time Ref.: RFC 2327
SDP_R	B00E	String	Zero or more repeat times Reference: RFC 2327
SDP_M	B00F	String	Media type, port, transport and format Ref.: RFC 2327

[End Correction]

6.36 Case Sensitivity of Profile Names

Description:	<p>Profiles of H.248 are becoming more widely used. One of the important features of the registration of profiles is that the profile name is unique and able to be agreed upon by a MGC and MG. There is a potential interoperability problem in that some implementers may assume that the "Profile Name" is case sensitive, where others may assume that the name is case insensitive. This could cause failure at profile negotiation.</p> <p>H.248.1 is largely silent of the case sensitivity of profile names. However as other as package names are case-insensitive it is believed that profile names are also case insensitive. Thus H.248.1 § 13 should be updated to reflect this.</p>
Reference:	AVD-2663

[Begin Correction]

13 Profile Definition

Profiles may be specified to further define how the H.248.1 protocol is used and what functionality is supported by a MG. It only describes the capabilities of the MGC/MG H.248 interface. The profile itself specifies what options associated with H.248.1 have been used. For example: transport and packages used for an application.

A profile is identified by a name (IANA registered) and a Version. A name shall be a case-insensitive string up to 64 characters long. Version shall be 1 to 99.

...

[End Correction]

6.37 Profile Negotiation

Description:	<p>Profile negotiation is ambiguous in the situation where the MG issues a ServiceChange request without the ServiceChangeProfile parameter (ie. NoProfile) and the MGC requires a profile to work.</p> <p>It is assumed that the MGC responds with the profile that it wants to support in this situation.</p> <p>However the current text in 7.2.8.1.11 indicates that the profile is “only” returned when the MGC cannot support the profiles in the ServiceChangeRequest. The contributors contend that if no ServiceChangeProfile is added to the request this is the MG indicated “NoProfile”. This should be clarified.</p> <p>Furthermore on reception of the Profile in the ServiceChangeResponse it is unclear whether the MG should:</p> <ul style="list-style-type: none">a) reissue a ServiceChange registration with the MGC indicated profile to accept the profile, or, <p>simply await commands from the MGC.</p>
Reference:	AVD-2663

[Begin Correction]

7.2.8.1.11 ServiceChange Command and Response

...

- ServiceChangeProfile, if the responder wishes to negotiate the profile to be used for the association. The profile (name and version) is ~~only~~ returned in reply in the case that the MGC cannot support the specified profiles in the ServiceChangeRequest. The profile “NoProfile” is assumed if no ServiceChangeProfile is included in the ServiceChangeRequest. The returned reply shall indicate the profile and version supported or "NoProfile" if no profile is supported. Upon reception of a profile in the reply the MG may continue the relationship with the current MGC by issuing a subsequent ServiceChangeRequest with the appropriate profile or contact secondary MGCs and establish a relationship with them. If the profile is not returned the MGC will use the capabilities specified by the Profile indicated in the service change request;

...

[End Correction]

6.38 Media Type Mismatch

Description:	<p>H.248.1 allows media descriptor parameters to be set independently on terminations in different commands. The H.248 connection model also makes use of this fact to be able to describe different functions on an MG. For example if Codec=GSM AMR is specified on TerminationA and Codec=G.711 is specified on TerminationB then the MG can assume that transcoding will take place.</p>
---------------------	--

	<p>The problem comes when the MGC tries to change the codec on one or more of the terminations. For example: the MGC wants to change Terminations A and B to both to G.729. It issues a MOD.req on Termination A Codec=G729. What does the MG do? Does it try to insert another transcoder to transcode G.729 to G.711? This would be a waste of resources as no transcoding will be required when TerminationB is set. Does the MG reject the command as it can't support G.729 to G.711 transcoding?</p> <p>The MG must have unambiguous knowledge of when to apply a function eg. transcoding. Therefore it is proposed that the MG only applies the function when the streammode associated with the function is NOT inactive.</p>
Reference:	COM 16 D-119

[Begin Correction]

6.2.4 Termination properties and descriptors

...

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.

By setting properties on different terminations in the same context the MG can be instructed to perform certain functions. For example: if a G.711 codec is set on Termination A and a G.729 codec is set on Termination B then the MG performs a transcoding function. This transcoding function is activated when the Mode Property (see Local Control Descriptor 7.1.7) on the termination/streams affected by the function are not set to "Inactive".

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 behavior. For more details, see 7.1 dealing with the individual Descriptors.

...

[End Correction]

6.39 Notify Avalanche

Description:	<p>During a failure of the MGC-MG control link a large number of events could occur on the MG. During this period the MG may still continue to generate Notify requests that need to be sent to the MGC. However as the control link has failed these Notify reqs would be queued. Once the link is re-established (ServiceChangeMethod = Disconnected) then these Notify requests would be sent. This could result in a potential signaling avalanche problem. An avalanche problem has previously been identified in H.248.1 section "9.2 Protection against restart avalanche" however this was restricted to ServiceChange commands.</p> <p>It is proposed to add some advisory text to minimize the chances of Notify avalanches.</p>
Reference:	COM 16 D-119

[Begin Correction]

9.3 Protection against Notify avalanche

In the event that a control association goes down the MG may continue to generate notify messages. These Notify messages must be buffered until the control association comes back into service (ServiceChangeMethod = Disconnected). When the control association comes back into service the rapid sending of the Notifications may result in a notification avalanche. To prevent this from occurring the MG should send the Notifications in a restricted manner until the buffer is cleared.

...

[End Correction]

6.40 Topology Reply

Description:	<p>H.248.1 section 7.1.18 states that "It is possible to have an action containing only a Topology descriptor". In the text encoding, what should the response be to such an action? The ABNF requires something in the actionReply - at least one context property, command reply or error descriptor.</p> <p>The behaviour for v1 and v2 needs to be described. E.g. the MG should echo back what was sent. If CHOOSE \$ was indicated you should send back the chosen TerminationID.</p> <p><i>Editor's Note: It is not permissible to have \$ in a TP descriptor without an Add command in the same action.</i></p>
Reference:	<p>Subject: RE: [Megaco] Topology Questions Date: Wed, 29 Jun 2005 05:35:39 +1000 From: Kevin Boyle <kboyle@nortel.com> To: Christian Groves (BR/EPA) <christian.groves@ericsson.com>, Frank Reno <hmegaco@yahoo.com> CC: megaco@ietf.org</p>

[Begin Correction]

7.1.18 Topology 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. In the instance where there are no commands in an action containing a Topology Descriptor, the Topology Descriptor is echoed back to the MGC.

...

[End Correction]

6.41 Protocol version negotiations

Description:	H.248.1 is silent on what to do when either the MGC or the MG fails to abide by the
---------------------	---

	<p>negotiated protocol version within a control association. Consider the following:</p> <p>The MG offers Version 2, which the MGC accepts. The MG then starts sending all messages as Version 1.</p> <p>This is clearly not what was intended in the version negotiation procedures. H.248.1 should allow the receiver of the “off-version” messaging to reject it as not in line with the negotiated version. The most appropriate error code is 406, “Version not supported”.</p>
Reference:	AVD-2820

[Begin Correction]

11.3 Negotiation of protocol version

...

If the MGC supports the version indicated by the MG, ~~both the MGC and MG~~ shall conform to that version in all subsequent messages. In this case it is optional for the MGC to return a version in the ServiceChange Reply. Any subsequent messaging that does not conform to the negotiated version shall be rejected with Error Code 406 (“Version Not Supported”).

...

[End Correction]

6.42 ServiceStates clarification for continuity testing

Description:	The continuity package does not specify whether or not a termination must be placed in the Test state prior to conducting a continuity test.
Reference:	COM16 D-224

[Begin Correction]

E.10.5 Procedures

...

When a continuity test is performed on a Termination, no echo devices or codecs shall be active on that Termination. The termination under test does not need to have its ServiceStates Property set to Test.

...

[End Correction]

6.43 Clarification of termination service state upon restart of MG

Description:	During discussion on the 3GPP and IETF Megaco mailing lists it became apparent that there is a source of confusion on the default states of all terminations after a ServiceChange restart. It is widely agreed that all terminations including physical and ephemeral terminations are default “InService” after the ServiceChange. However H.248.1 doesn’t
---------------------	--

	explicitly make this statement.
Reference:	COM16 D-274

[Begin Correction]

7.2.8 ServiceChange

...

- 3) Restart – indicates that service will be restored on the specified tTerminations after expiration of the ServiceChangeDelay. The SserviceStates Property should be set to "~~inService~~InService" upon expiry of ServiceChangeDelay. Upon receipt of a ServiceChange Command on Root with ServiceChangeMethod Restart all terminations are assumed to be "InService". This includes physical and ephemeral terminations. Those terminations which are "OutOfService" may be reported by subsequent ServiceChange Commands with ServiceChangeMethod Forced.

...

[End Correction]

6.44 Alignment of text among events in the Tone Detection Package

Description:	<p>The Tone Detection Package specifies three different events for tone detection: 'Start Tone Detected', 'End Tone Detected' and 'Long Tone Detected'. While the '*' wildcard in the EventsDescriptor parameter 'tl' is allowed for the 'Start Tone Detected' and 'Long Tone Detected' events, H.248 currently doesn't allow it in the 'End Tone Detected' event.</p> <p>There is no reason why the wildcard should not be allowed in the 'End Tone Detected' event. In fact, a very common use of this event, as of the other two, is the detection of DTMF tones. For DTMF it is common to order the MGW to detect any DTMF digit, as it is not known in advance which DTMF digit will be received in the line.</p>
Reference:	COM16 D-303

[Begin Correction]

E.4.2 Events

...

End tone detected

EventID: etd, 0x0002

Detects the end of a tone.

EventDescriptor parameters:

Tone id list

ParameterID: tl (0x0001)

Type: enumeration or list of enumerated types

Possible values: The only tone id defined in this package is "wild card" which is "*" in text encoding and 0x0000 in binary. No possible values are specified in this package. Extensions to this package would add possible values for tone id. If tl is "wild card", any tone id is detected.

...

[End Correction]

6.45 Clarification of package definition requirements for enumerations

Description:	Packages may define properties, statistics and parameters for signals and events of enumeration type. As stated in the guideline for package definition in H.248.1, the possible values for these parameters must be also specified in the package. Once the values are specified, the binary encoding is unambiguous, as with ASN.1 each of the values of an enumeration type is associated to an integer. However, with text encoding, the encoded values may use any character or character string, not only integers. Therefore it is important that the package specifies not only the possible values that a property, statistic or parameter of type enumeration may take, but also the strings to be used to encode each of the values if ABNF, Annex B/H.248.1 is used.
Reference:	COM16 D-303

[Begin Correction]

12.1.2 Properties

Properties defined by the package, specifying:

...

Type: One of:

Boolean

String: UTF-8 string

Octet String: A number of octets. See Annexes A and B.3 for encoding.

Integer: 4 byte signed integer

Double: 8 byte signed integer

Character: ~~unicode~~Unicode UTF-8 encoding of a single letter. Could be more than one octet.

Enumeration: one of a list of possible unique values (see 12.3). Packages MUST define the text and binary encodings for each value in the enumeration.

Sub-list: a list of several values from a list. The type of sub-list SHALL also be specified. The type shall be chosen from the types specified in this clause (with the exception of sub-list). For example, Type: sublist of enumeration. The encoding of sub-lists is specified in Annexes A and B.3.

...

12.2 Guidelines to defining Parameters to Events and Signals

...

Type: One of:

Boolean

String: UTF-8 octet string

Octet String: A number of octets. See Annexes A and B.3 for encoding.

Integer: 4-octet signed integer

Double: 8-octet signed integer

Character: ~~unicode~~Unicode UTF-8 encoding of a single letter. Could be more than one octet.

Enumeration: one of a list of possible unique values (see 12.3). Packages MUST define the text and binary encodings for each value in the enumeration.

Sub-list: a list of several values from a list (not supported for statistics). The type of sub-list SHALL also be specified. The type shall be chosen from the types specified in this clause (with the exception of sub-list). For example, Type: sub-list of enumeration. The encoding of sub-lists is specified in Annexes A and B.3.

...

[End Correction]

6.46 Clarification of use of ABNF encodings of octet strings

Description:	Properties, statistics and signal and events parameters can be defined as of type Octet String, among other types. This is described in H.248.1 12.1.2, 12.1.5 and 12.2. These chapters refer to Annex B.3 for how the actual encoding of the Octet String shall be done. Annex B.3 does indeed describe a method for the encoding of strings, but fails to make a precise reference to the type Octet String, as the object of the method it is describing. Instead, it talks about “representing a string of octets” or “encoding octet strings”. As ABNF defines still another type called “octetString” to describe SDP lines, and which is different to the type Octet String defined above (is not compatible), there is a risk to misinterpret the applicability of B.3.
Reference:	COM16 D-303

[Begin Correction]

B.3 Hexadecimal octet coding

Hexadecimal octet coding is a means of representing ~~a string of octets~~ package elements of type Octet String as a string of hexadecimal digits, with two digits representing each octet. This octet encoding should be used when encoding ~~octet strings~~ values of type Octet String in the text version of the protocol.

For each octet, the 8-bit sequence is encoded as two hexadecimal digits. Bit 0 is the first transmitted; bit 7 is the last.

Bits 7-4 are encoded as the first hexadecimal digit, with Bit 7 as MSB and Bit 4 as LSB. Bits 3-0 are encoded as the second hexadecimal digit, with Bit 3 as MSB and Bit 0 as LSB.

Examples:

Octet bit pattern	Hexadecimal coding
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

This encoding is not applicable to the octetString construct defined in section B.2

[End Correction]

6.47 Clarification of encoding for packet loss statistic in Annex E.12

Description:	H.248.1 E.12.4 defines the statistics packet loss rtp/pl to describe the packet loss rate, as a
---------------------	---

	percentage. Although this statistics element is defined as type double, it is meant to hold both the whole part and the fractional part of the percentage. The ASN.1 “double” encoding of this element entails multiplying the percentage by 2^{32} in order to obtain an integer and then use 4 octets to encode the resulting integer. In ABNF is questionable if the same applies, as that would lead to a long string. This seems unnecessary, especially considering that RFC 3550, to which E.12.4 refers to when defining rtp/pl, defines the packet fraction lost with only 8 bits. Therefore it is proposed to clarify that the notation x.y is allowed when encoding rtp/pl with ABNF.
Reference:	COM16 D-303

[Begin Correction]

E.12.4 Statistics

...

Packet Loss

StatisticID: pl (0x0006)

Describes the current rate of packet loss on an RTP stream, as defined in IETF RFC 1889. Packet loss is expressed as percentage value: number of packets lost in the interval between two reception reports, divided by the number of packets expected during that interval.

Type: double

Possible values: a 32-bit whole number and a 32-bit fraction. The value shall be encoded in ABNF as “x.y” where x is the whole part and y the fractional part of the percentage.

...

[End Correction]
