INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

H.248 Series Implementors' Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(08/06/2001)

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

Media Gateway Control Protocol
Implementors' Guide

Contact Information

ITU-T Study Group 16 /
Question 3 Rapporteur

Glen Freundlich
Avaya Communications
1300 W. 120th Avenue
Westminster, Colorado 80234
USA

Tel: +1 303 538 2899
Fax: +1 303 538 3907
E-mail: ggf@avaya.com

Implementors' Guide ITU-T
Recommendation H.248
Editor

Christian Groves
Ericsson Australia
37/360 Elizabeth St
Melbourne, Victoria 3000
Australia

Tel: +61 3 9301 6116
Fax: +61 3 9301 1499
E-mail: Christian.Groves@ericsson.com

1 Introduction

This document is a compilation of reported defects identified as at 08/06/2001 with the following recommendations:

- 2000 decided edition of ITU-T Recommendation H.248,
- 2000 decided edition of ITU-T Recommendation H.248 Annex F
- 2000 decided edition of ITU-T Recommendation H.248 Annex H
- 2000 decided edition of ITU-T Recommendation H.248 Annex K
- RFC3015

It must be read in conjunction with the Recommendation to serve as an additional authoritative source of information for implementors'. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.248 Recommendations.

2 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 Recommendations that are not strictly related to implementation difficulties in the above categories. Proposals for new features should be made in through contributions to the ITU-T.

3 Defect Resolution Procedure

Upon discovering technical defects with any components of the H.248 Recommendation, please provide a written description directly to the editors of the affected Recommendations 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 anyone interested in H.248 Recommendation. Formal membership in the ITU is not required to participate in this process.

4 References

ITU-T Recommendation H.248 (2000), *Media Gateway Control Protocol*

ITU-T Recommendation H.248 (2000) Annex F, *Facsimile, Text Conversation and Call Discrimination packages*

ITU-T Recommendation H.248 (2000) Annex H, *Transport over Stream Control Transmission Protocol (SCTP)*

ITU-T Recommendation H.248 (2000) Annex K, *Generic Announcement Package*

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**
[Begin Correction]	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
[End Correction]	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.
--- *SPECIAL INSTRUCTIONS* --- *{instructions}*	Indicates a set of special editing instructions to be followed.

6 Technical and Editorial Corrections to ITU-T Recommendation H.248 (2000)

6.1 Correction in bibliographic reference

Description:	Section 2.1/H.248 contains a bibliographic reference to Q.765. The Q.765 series consists of a number of recommendations. The correct reference is Q.765.5, Application transport mechanism – Bearer independent call control (BICC), instead of the entire series.

[Begin Addition]

2.1 Normative references

…

~~ITU-T Recommendation Q.765, "Signalling System No. 7 – Application transport mechanism".~~

ITU-T Recommendation Q.765.5, "Application transport mechanism – Bearer independent call control (BICC)".

…

[End Addition]

The reference to Q.765 in section C.1/H.248 should be corrected too:

[Begin Correction]

ACodec	1006	Octet String	Audio Codec Type: Reference: ITU-T Rec. Q.765.5 - ~~Application Transport Mechanism~~ Non-ITU codecs are defined with the appropriate standards organisation under a defined Organizational identifier

[End Correction]

6.2 Valid parameters to Add, Modify and Move commands

Description:	The EventBufferDescriptor parameter was inadvertently omitted as a valid parameter to Add, Modify and Move commands in sections 7.2.1, 7.2.2, 7.2.4/H.248.

[Begin Addition]

7.2.1 Add

The Add command adds a Termination to a Context.

TerminationID
[, MediaDescriptor]
[, ModemDescriptor]
[, MuxDescriptor]
[, EventsDescriptor]
[, SignalsDescriptor]
[, DigitMapDescriptor}
[, ObservedEventsDescriptor]
[, EventBufferDescriptor]

```
[, StatisticsDescriptor]
[, PackagesDescriptor]
          Add(     TerminationID
                   [, MediaDescriptor]
                   [, ModemDescriptor]
                   [, MuxDescriptor]
                   [, EventsDescriptor]
                   [, EventBufferDescriptor]
                   [, SignalsDescriptor]
                   [, DigitMapDescriptor]
                   [, AuditDescriptor]
          )
```

[End Addition]

[Begin Addition]

…

The EventsDescriptor parameter is optional. If present, it provides the list of events that should be detected on the Termination.
The EventBufferDescriptor parameter is optional. If present, it provides the list of events that the MG is requested to detect and buffer when EventBufferControl equals LockStep.

…

[End Addition]

[Begin Addition]

7.2.2 Modify

The Modify command modifies the properties of a Termination.

```
TerminationID
[, MediaDescriptor]
[, ModemDescriptor]
[, MuxDescriptor]
[, EventsDescriptor]
[, SignalsDescriptor]
[, DigitMapDescriptor}
[, ObservedEventsDescriptor]
[, EventBufferDescriptor]
[, StatisticsDescriptor]
[, PackagesDescriptor]
          Modify( TerminationID
                   [, MediaDescriptor]
                   [, ModemDescriptor]
                   [, MuxDescriptor]
                   [, EventsDescriptor]
                   [, EventBufferDescriptor]
                   [, SignalsDescriptor]
                   [, DigitMapDescriptor]
                   [, AuditDescriptor]
          )
```

[End Addition]

[Begin Addition]

7.2.4 Move

…

```
TerminationID
[, MediaDescriptor]
[, ModemDescriptor]
[, MuxDescriptor]
[, EventsDescriptor]
[, SignalsDescriptor]
[, DigitMapDescriptor}
[, ObservedEventsDescriptor]
[, EventBufferDescriptor]
[, StatisticsDescriptor]
[, PackagesDescriptor]
        Move(   TerminationID
                [, MediaDescriptor]
                [, ModemDescriptor]
                [, MuxDescriptor]
                [, EventsDescriptor]
                [, EventBufferDescriptor]
                [, SignalsDescriptor]
                [, DigitMapDescriptor]
                [, AuditDescriptor]
        )
```

[End Addition]

6.3 Cold Start

Description:	Section 11.2/H.248 contains a leftover from old terminology: Transaction Accept instead of Transaction Reply.

[Begin Correction]

11.2 Cold Start

A MG is pre-provisioned by a management mechanism outside the scope of this protocol with a Primary and (optionally) an ordered list of Secondary MGCs. Upon a cold start of the MG, it will issue a ServiceChange command with a "Restart" method, on the Root Termination to its primary MGC. If the MGC accepts the MG, it will send a Transaction ~~Accept~~ Reply, with the ServiceChangeMgcId set to itself.

…

[End Correction]

6.4 Digit map syntax

Description:	Annex A.3/H.248 provides a copy of the syntax of digit maps, stating that the definition given in Annex B/H.248 takes precedence in case of discrepancies. A discrepancy occurs in the production rule for `digitStringList`, and it is proposed to correct this discrepancy: the quoted forward slash should be replaced by a quoted vertical bar.

[Begin Correction]

A.3 Digit maps and path names

…

digitStringList = digitString *(LWSP "~~/~~|" LWSP digitString)

[End Correction]

6.5 Omission in specification of text encoding

Description:	The specification of the text encoding of H.248 messages currently allows multiple occurrences of the same servicechange parameter, while the intention is that every such parameter should occur only once. The proposed resolution is to add a comment to the ABNF indicating this restriction.

[Begin Correction]

B.2 ABNF specification

…

```
; each parameter at-most-once
serviceChangeParm    = (serviceChangeMethod / serviceChangeReason /
                        serviceChangeDelay / serviceChangeAddress /
                        serviceChangeProfile / extension / TimeStamp /
                        serviceChangeMgcId / serviceChangeVersion )
```

[End Correction]

6.6 Ambiguity in text encoding

Description:	The text encoding as specified in Annex B/H.248 contains an ambiguity because the token "EB" was inadvertently used twice, in the production rules for EmbedToken and EventBufferToken. It is proposed to change the short tokens in the production rules for EmbedToken and EmergencyToken, and leave the production rule for EventBufferToken unchanged.

[Begin Correction]

B.2 ABNF specification

…

```
EmbedToken         = ("Embed"        / "EM~~EB~~")
EmergencyToken     = ("Emergency"    / "EG~~EM~~")
```

[End Correction]

6.7 Use of ServiceChange for MG registration

Description:	There is an inconsistency between section 7.2.8 and section 11.2 on when the ServiceChangeMgcID is used.

[Begin Correction]

11.2 Cold Start

…

A MG is pre-provisioned by a management mechanism outside the scope of this protocol with a Primary and (optionally) an ordered list of Secondary MGCs. Upon a cold start of the MG, it will issue a ServiceChange command with a "Restart" method, on the Root Termination to its primary MGC. If the MGC accepts the MG's registration, ~~it will send a Transaction Accept, with the ServiceChangeMgcId set to itself. If the MG receives an ServiceChangeMgcId not equal to~~

~~the MGC it contacted, it sends a ServiceChange to the MGC specified in the ServiceChangeMgcId.~~ it sends a Transaction Reply not including a ServiceChangeMgcId parameter. If the MGC does not accept the MG's registration, it sends a Transaction Reply, providing the address of an alternate MGC to be contacted by including a ServiceChangeMgcId parameter.

If the MG receives a Transaction Reply that includes a ServiceChangeMgcId parameter, it sends a ServiceChange to the MGC specified in the ServiceChangeMgcId. It continues this process until it gets a controlling MGC to accept its registration, or it fails to get a reply. Upon failure to obtain a reply, either from the Primary MGC, or a designated successor, the MG tries its pre-provisioned Secondary MGCs, in order. If the MG is unable to establish a control relationship with any MGC, it shall wait a random amount of time as described in section 9.2 and then start contacting its primary, and if necessary, its secondary MGCs again.

It is possible that the reply to a ServiceChange with Restart will be lost, and a command will be received by the MG prior to the receipt of the ServiceChange response. The MG shall issue error 505 – Command Received before Restart Response.

[End Correction]

6.8 Echo cancellation parameters

Description:	Appendix E.13.1/H.248 contains a property to turn echo cancellation off or on. In addition, C.1 and C.9 contain codepoints dealing with echo cancellation. The codepoints are • Echocanc (100B), with allowed values Off, G.165 and G.168; • ECHOCI (9021), with allowed values Off, incoming echo canceler on, outgoing echo canceler on, and incoming and outgoing echo canceler on. The codepoints in Annex C/H.248 are for use with binary encoding only, while packages define properties for use with both text and binary encodings. In addition it is expected that SG 11 will complete work on their SPNE package, allowing more advanced control of echo cancellers than the basic control offered by the TDM circuit package of Annex E.13.1. Therefore it is proposed that the codepoints of Annex C dealing with echo cancellation be deprecated, and that the entries in the tables in C.1 and C.9 be listed as Reserved.

[Begin Correction]

C.1 General Media Attributes

…

Echocanc	100B	~~Enumeration~~	~~Echo Canceller: Off(0), G.165(1), G168(2)~~ Not used. See H.248 E.13 for an example of possible Echo Control properties.

[End Correction]

[Begin Correction]

C.9 Bearer Capabilities

…

ECHOCI	9021	~~Enumeration~~	~~Echo Control Information~~ ~~echo canceler off (0), incoming echo canceler on (1), outgoing echo canceler on (2), incoming and outgoing echo canceler on (3)~~ Not used. See H.248 E.13 for an example of possible Echo Control properties.

[End Correction]

6.9 Topology Triples in ABNF

Description:	In the ABNF (Annex B), the term TopologyDescriptor allows the specification of only one triple. The ASN.1 permits a sequence of such triples.

[Begin Correction]

B.2 ABNF Specification

…

topologyDescriptor = TopologyToken LBRKT topologyTriple
*(COMMA topologyTriple) RBRKT
topologyTriple = terminationA COMMA terminationB COMMA topologyDirection ~~RBRKT~~

[End Correction]

6.10 Local Control for Annex C and optionality

Description:	Currently the introduction of Annex C specifies that native tags are applicable to Local and Remote descriptors. This introduction should also say that native tags are applicable to the Local Control descriptor as the ASN1 encoding makes use of native tags in the Local Control descriptor.
Reference:	Subject: Re: MEGACO: More Annex C Questions Date: Wed, 06 Dec 2000 11:36:38 -0500 From: Troy Cauble <troy@lucent.com> To: Christian Groves <Christian.Groves@ericsson.com>

[Begin Correction]

ANNEX C TAGS FOR MEDIA STREAM PROPERTIES (NORMATIVE)

Parameters for Local ~~descriptors,and~~ Remote and Local Control 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".

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 section 7.1.1.

It is optional to support the properties in this Annex or any of its sub-sections. For example 3 properties from C.3 and five properties from C.8 may be implemented only.

[End Correction]

6.11 Echo Canceller Default

Description:	As the Echo Cancellation properties in Annex C have been deprecated in 6.8 of this implementors' guide the default of the Echo Canceller property should be provisioned to allow for a wider change of applications.

[Begin Correction]

E.13.1 Properties

Echo Cancellation
PropertyID: ec (0x0008)
~~By default, the telephony gateways always perform echo cancellation. However, it is necessary, for some calls, to turn off these operations.~~
Type: boolean
Possible Values:
"on" (when the echo cancellation is requested) and
"off" (when it is turned off.)
The default is ~~"on~~provisioned~~"~~.
Defined In: LocalControlDescriptor
Characteristics: read/write

[End Correction]

6.12 Error in text on interim AH header

Description:	The UDP destination port should be encoded as 20 hex digits, representing 10 bytes (4 source, 4 dest, 2 port). The Interim AH scheme should apply to all transactions in a message. The current text in H.248 section 10.2 indicates that the ICV calculation should be performed on one transaction. This is incorrect.
Reference:	Subject: RE: Interim AH Scheme Date: Wed, 28 Mar 2001 09:53:25 -0500 From: "Tom-PT Taylor" <taylor@nortelnetworks.com> To: "'Christian Groves'" <Christian.Groves@ericsson.com>, girirs@netlab.hcltech.com CC: megaco@fore.com

[Begin Correction]

10.2 Interim AH Scheme

…

As an interim solution, an optional AH header is defined within the H.248 protocol header. The header fields are exactly those of the SPI, SEQUENCE NUMBER and DATA fields as defined in [RFC2402]. The semantics of the header fields are the same as the "transport mode" of [RFC2402], except for the calculation of the Integrity Check value (ICV). In IPsec, the ICV is calculated over the entire IP packet including the IP header. This prevents spoofing of the IP addresses. To retain the same functionality, the ICV calculation should be performed across all the transactions (concatenated) in the message ~~the entire transaction~~ prepended by a synthesized IP header consisting of a 32 bit source IP address, a 32 bit destination address and a 16 bit UDP destination port encoded as ~~10~~20 hex digits. When the interim AH mechanism is employed when TCP is the transport Layer, the UDP Port above becomes the TCP port, and all other operations are the same.

[End Correction]

6.13 Termination Subtract from NULL context

Description:	A subtraction of a termination from a NULL context is not allowed however this is not clear in the recommendation. This should be stated.

[Begin Correction]

7.2.3 Subtract

…

ALL may be used as the ContextID as well as the TerminationId in a Subtract, which would have the effect of deleting all contexts, deleting all ephemeral terminations, and returning all physical terminations to Null context. Subtract of termination from the NULL context is not allowed.

[End Correction]

6.14 Missing M= Line in Annex SDP

Description:	Section C.11 SDP Equivalents lists various SDP encoding lines. However the Media Line (m=) is missing from this table. The Media line should occur in this table.

[Begin Addition]

C.11 SDP Equivalents

…

SDP_M	B00F	STRING	Media name and transport address Reference: IETF RFC2327

[End Correction]

6.15 Missing Optional on Keepactive Flag

Description:	In section 7.1.9 EventsDescriptor is states " Each event in the descriptor contains the Event name, an optional streamID, an optional KeepActive flag, and optional parameters." Clearly the KeepActive flag is meant to be optional however in the ASN.1 this flag is mandatory. The ASN.1 should be updated indicating OPTIONAL.

[Begin Correction]

A.2 ASN.1 syntax specification

…

```
      RequestedActions ::= SEQUENCE
      {
            keepActive              BOOLEAN OPTIONAL,
            eventDM                 EventDM OPTIONAL,
            secondEvent             SecondEventsDescriptor OPTIONAL,
            signalsDescriptor       SignalsDescriptor OPTIONAL,
            ...
      }
```

…

```
      SecondRequestedActions ::= SEQUENCE
      {
            keepActive              BOOLEAN OPTIONAL,
            eventDM                 EventDM OPTIONAL,
            signalsDescriptor       SignalsDescriptor OPTIONAL,
            ...
```

} [End Correction]

6.16 Syntax Problem in Appendix A

Description:	According to the definition of digitMapRange : digitMapRange = ("x" / LWSP "[" LWSP digitLetter LWSP "]" LWSP) "x" must be followed by "[" .

[Begin Correction]

B.2 ABNF syntax specification

```
                                    ...
digitMapRange         = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))
```

[End Correction]

6.17 Retaining Descriptors on MOVE

Description:	When a MOVE command is issued on a termination the descriptors currently residing on that termination are retained. This is current ambiguous in the recommendation text.

[Begin Correction]

7.2.4 Move

~~The remaining descriptors are processed as in the Modify Command.~~ The Move command does not affect the properties of the Termination on which it operates, except those properties explicitly modified by descriptors included in the Move command. The AuditDescriptor with the Statistics option, for example, would return statistics on the Termination just prior to the Move. Possible descriptors returned from Move are the same as for Add.

[End Correction]

6.18 Clarification on extending packages

Description:	An extended package shall not redefine or overload an identifier defined in the original package. For example: if package "aa" has a signal "ab", then if package "bb" extends aa it cannot define signal "ab". This is also valid for not redefining an id in "earlier" packages, when multiple levels of extension occur. Several packages in H.248 Annex E have made this error. Corrections are below.

[Begin Correction]

12.1.1 Package

A package may extend an existing package. The version of the original package must be specified. When a package extends another package it shall only add additional Properties, Events, Signals, Statistics and new possible values for an existing parameter described in the original package. An extended package shall not redefine or overload an identifier~~name~~ defined in the original package and~~.~~ packages it may have extended (multiple levels of extension).

Hence, if package B version 1 extends package A version 1, version 2 of B will not be able to extend the A version 2 if A version 2 defines a name already in B version 1.

[End Correction]

[Begin Correction]

Section E.6.2 Events

…

DigitMap Completion Event

EventID: ce, 0x0004~~1~~

Generated when a digit map completes as described in section 7.1.14.

[End Correction]

[Begin Correction]

Section E.7.3

Signal Name	**Signal ID/tone id**
Dial Tone	dt (0x0030)
Ringing Tone	rt (0x0031)
Busy Tone	bt (0x0032)
Congestion Tone	ct (0x0033)
Special Information Tone	sit(0x0034)
Warning Tone	wt (0x0035)
Payphone Recognition Tone	prt (0x0036)
Call Waiting Tone	cw (0x0037)
Caller Waiting Tone	cr (0x0038)

[End Correction]

6.19 Context = ALL in Transaction Reply

Description:	The ASN1 and ABNF allows for the return of Context ID = ALL. This is used in the Wildcard response case. However the text in 8.2.2 Transaction Reply states that only a Specific or NULL is valid for the Context ID. This should be updated to allow ALL.

[Begin Correction]

8.2.2 TransactionReply

…

The ContextID parameter must specify a value to pertain to all Responses for the action. The ContextID may be specific, all or null.

…

[End Correction]

6.20 Number of Events on a Termination

Description:	The text in H.248 Sect 9.1 says that "On a given Termination, there should normally be at most one outstanding Notify command at any time." This is only a consideration when using UDP. A termination typically can realise multiple events on a terminations.

[Begin Correction]

9.1 Ordering of Commands

…

3. For transport that do not guarantee in sequence delivery of messages (ie. UDP), o~~O~~n a given Termination, there should normally be at most one outstanding Notify command at any time.

…

[End Correction]

6.21 Error Code for Number of Terminations in a Context exceeded

Description:	H.248 provides a means of setting the maximum number of terminations in a context. However no mechanism is provided to allow an error when the maximum number of terminations is requested to be exceeded.

[Begin Correction]

Command error code "434 - Max number of Terminations in a Context exceeded." has been added to H.248 Annex L – Error Codes.

[End Correction]

6.22 Optional Statistics Parameter Value

Description:	In section 7.2.6 for Audit capability its stated that "StatisticsDescriptor returns the names of the statistics being kept on the termination." But the ABNF grammar states the descriptor as "statisticsDescriptor = StatsToken LBRKT statisticsParameter *(COMMA statisticsParameter) RBRKT ;at-most-once per item statisticsParameter = pkgdName EQUAL VALUE" which doesn't allow specification of stats parameter always with value. If only names needs to be sent from MG, the value field needs to be made optional.

[Begin Correction]

B.2 ABNF Syntax Specification

…

statisticsParameter = pkgdName [EQUAL VALUE]

…

[End Correction]

[Begin Correction]

A.2 – ASN.1 Syntax Specification

…

```
StatisticsParameter ::= SEQUENCE
{
      statName          PkgdName,
      statValue         Value OPTIONAL
}
```

…

[End Correction]

6.23 Event Permanancy

Description:	The current Event Descriptor text is unclear on whether or not a event continues to trigger notifications after the first event is detected. The intention is that the event shall do this.

[Begin Correction]

7.1.9 Event Descriptor

…

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 MG. Notification may be deferred if the event is absorbed into the current dial string of an active digit map (see section 7.1.14). Any other action is for further study. 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 section. Unless the events descriptor is replaced by another events descriptor, it remains active after an event has been recognized.

…

[End Correction]

6.24 Wildcard Response Alignment between ASN1 and ABNF

Description:	Chapter 6.2.2 allows wildcard response on all commands. It mentions that it is useful for audit. The ASN1 allows you to request a wildcard response on all commands [w-]. The ABNF only has it at: auditRequest = ["W-"] (AuditValueToken / AuditCapToken) EQUAL TerminationID LBRKT auditDescriptor RBRKT Although they might be deduced, the meanings of the "O-" and "W-" command prefixes are not explained.
Reference:	Subject: Command prefixes Date: Sun, 31 Dec 2000 13:18:04 -0600 Reply-To: plong@ipdialog.com From: Paul Long <plong@packetizer.com> Organization: ipDialog, Inc. To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

Section B.2 ABNF Specification

…

```
; "O-" indicates an optional command
; "W-" indicates an wildcarded response to a command
commandRequestList= ["O-"]["W-"] commandRequest *
                             (COMMA ["O-"]["W-"]commandRequest)
                                          …

subtractRequest        =  ~~["W-"]~~SubtractToken EQUAL TerminationID
                          [ LBRKT auditDescriptor RBRKT]
                                          …

auditRequest           = ~~["W-"]~~ (AuditValueToken / AuditCapToken ) EQUAL
                          TerminationID LBRKT auditDescriptor RBRKT
                                          …
```

[End Correction]

6.25 MTP addressing for non ITU variants

Description:	The MTP MID needed to be updated to allow for American and Japanese variants as it only currently allows ITU defined MTP addresses to be used.

[Begin Correction]

A.2 ASN.1 Syntax Specification

```
MId ::= CHOICE
{
      ip4Address                    IP4Address,
      ip6Address                    IP6Address,
      domainName                    DomainName,
      deviceName                    PathName,
      mtpAddress                    OCTET STRING(SIZE(2..4)),
       -- Addressing structure of mtpAddress:
       --        ~~15~~               0
       --        |  PC         | NI |
       --        24 - ~~14~~ bits    2 bits
      -- Note: 14 bits is defined for international use.
       -- Two national options exist where the point code is 16 or 24 bits.
       -- To octet align the mtpAddress the MSBs shall be encoded as 0s.
                                              ...
}
```

B.2 ABNF Syntax Specification

```
                                              …
; Addressing structure of mtpAddress:
;                           0
;     |  PC          | NI |
;     24 - 14 bits    2 bits
; Note: 14 bits is defined for international use.
; Two national options exist where the point code is 16 or 24 bits.
; To octet align the mtpAddress the MSBs shall be encoded as 0s.
; An octet shall be represented by 2 hex digits.
~~; An mtp address is two octets long~~
mtpAddress             = MTPToken LBRKT 4*8(HEXDIG)~~octetString~~ RBRKT
```

[End Correction]

6.26 Audit Descriptor and Subtract and Statistics

Description:	The protocol document mentions: 7.1.1 Specifying Parameters ... "A missing Audit descriptor is equivalent to an empty Audit Descriptor." and also 7.1.15 Statistics Descriptor ... "By default, statistics are reported when the Termination is Subtracted from the Context. This behavior can be overridden by including an empty AuditDescriptor in the Subtract command." According to this text, if Subtract command does not have an AuditDescriptor it would mean that there is an empty audit descriptor and no statistics would be reported. And, if the MGC needs termination statistics, it must send AuditDescriptor with Statistics token in the Subtract command. But this seems to change the definition of "By default".

[Begin Correction]

7.1.1 Specifying Parameters

…

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~~A~~ missing Audit descriptor is equivalent to an empty Audit Descriptor. The behavior of the MG with respect to unspecified parameters within a descriptor varies with the descriptor concerened, as indicated in succeeding sections. Whenever a parameter is underspecified or overspecified, the descriptor containing the value chosen by the responder is included as output from the command.

[End Correction]

6.27 Signal Lists

Description:	There are several inconsistency in the way the signal has been documented they are: Section 7.1.11 states that Signal Lists have type. This is incorrect. Section E.1.2 Doesn't not allow for the specification of which list instance of a signal contained in several lists should generate a notify completion.

[Begin Correction]

7.1.11 Signal Descriptor

A sequential signal list consists of a signal list identifier and~~,~~ a sequence of signals to be played sequentially~~, and a signal type~~. Only the trailing element of the sequence of signals in a sequential signal list may be an on/off signal.~~. If the trailing element of the sequence is an on/off signal, the signal type of the sequential signal list shall be on/off as well. If the sequence of signals in a sequential signal list contains signals of type timeout and the trailing element is not of type on/off, the type of the sequential signal list SHALL be set to timeout.~~ The duration of a sequential signal list with type timeout is the sum of the durations of the signals it contains. ~~If the sequence of signals in a sequential signal list contains only signals of type brief, the type of the sequential signal list SHALL be set to brief. A signal list is treated as a single signal of the specified type when played out.~~

A sequential signal list consists of a signal list identifier 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.

[End Correction]

[Begin Correction]

Section E.1.2 Events

…

Termination Method

ParameterID: Meth (0x0002)

Indicates the means by which the signal terminated.

Type: enumeration

Possible values:

"TO" (0x0001) Duration expired
"EV" (0x0002) Interrupted by event
"SD" (0x0003) Halted by new Signals Descriptor
"NC" (0x0004) Not completed, other cause

Signal List ID

ParameterID: SLID (0x0003)

Indicates to which signal list a signal belongs. The SignalList ID is only returned in cases where the signal resides in a signal list.

Type: integer

Possible values: Any integer

[End Correction]

6.28 Topology

Description:	Topology specifications are cumulative over the life of the context. This is ambiguous in the text.

[Begin Correction]

7.1.18 Topology Descriptor

…

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. ~~The default association when a termination is not mentioned in the Topology descriptor is bothway (if T3 is added to a context with T1 and T2 with topology (T3,T1,oneway) it will be connected bothway to T2).~~ 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 (eg. if T3 is added to a context with T1 and T2 with topology (T3,T1,oneway) it will be connected bothway to T2).

[End Correction]

6.29 Value optionality in Packages

Description:	When supporting packages you must support all properties, signals, events and statistics. It is unclear in the specification on whether or not you must support all values of properties and parameter. The intention is that a subset of values may be supported.

[Begin Correction]

6.2.3 Packages

…

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, StatisticsIds 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 support all Properties, Signals, Events and Statistics defined in a package. It must also support all Signal and Event parameters. The MG may support a subset of the values listed in a package for a particular Property or Parameter.

…

[End Correction]

6.30 RequestID in AuditCapReply

Description:	Section 7.2.6 says "... The EventsDescriptor returns the list of possible events on the Termination together with the list of all possible values for the EventsDescriptor Parameters. ..." What is the value of requestId sent in the events Descriptor from MG to MGC for a AuditCap reply ? ALL should be returned in this case.

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
-- For an AuditCapReply with all events, the RequestID SHALL be ALL.
-- ALL is represented by 0xffffffff.

RequestID ::= INTEGER(0..4294967295)   -- 32 bit unsigned integer
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
; For an AuditCapReply with all events, the RequestID should be ALL.
RequestID            = ( UINT32 / "*" )
```

…

[End Correction]

6.31 Context ID Audit

Description:	H.248 allows you the audit the Context ID of where a termination currently belongs. This is not represented in the table in section 7.2.5 Audit Value. It should be added.

[Begin Correction]

Section 7.2.5 Audit Value

…

All	wildcard	Audit of all matching Terminations and the Context to which they are associated
All	Root	List of all ContextIds
All	Specific	(Non-null) Context Id in which the Termination currently exists

[End Correction]

6.32 Context Priorities

Description:	It is unclear in the recommendation on what the values of the priorities represent.

[Begin Clarification]

6.1.1 Context Attributes and Descriptors

…

- The priority 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 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.

…

[End Clarification]

6.33 Error and topology descriptors

Description:	Section 7.3/H.248 correctly states that when a MG reports an error to a MGC, it does so in an error descriptor. However, the preceding sections listing (supposedly) all descriptors and detailing their contents do not contain any reference to error descriptors. The command descriptions in Section 7.2 also seems to disallow error descriptors to be returned by MGs in reponse to requests from MGCs. We propose to add clarifying text as shown below. The topology descriptor was also accidentally omitted from the table in Section 6.2.4; its use is, however, explained in Section 7.1.18
Reference:	Subject: RE: Wherefore out thou Error Descriptor? Date: Tue, 24 Apr 2001 15:50:40 -0500 From: Paul Long <plong@ipdialog.com> To: <megaco@fore.com>

[Begin Correction]

6.2.4 Termination Properties and Descriptors

…

Statistics	In Subtract and Audit, Report of Statistics kept on a Termination
Topology	Specifies flow directions between Terminations in a Context

Error	Contains and error code and optionally error text; it may occur in command replies and in Notify requests

…

[End Correction]

[Begin Correction]

7.1.19 Error descriptor

If a command responder encounters an error when processing a transaction request, it must include an error descriptor in its response. A Notify request may contain an error descriptor as well.
An error descriptor consists of an error code, optionally accompanied by an error text. Section 7.3 contains a list of valid error codes.

An error descriptor shall be specified at the "deepest level" that is semantically appropriate for the error being described and that is possible given any parsing problems with the original request. An error descriptor may refer to a syntactical construct other than where it appears. For example, error descriptor, 422 – Syntax Error in Action, could appear within a command even though it refers to the larger construct--the action--and not the particular command within which it appears.

[End Correction]

[Begin Correction]

7.2 Command Application Programming Interface

Following is an Application Programming Interface (API) describing the Commands of the protocol. This API is shown to illustrate the Commands and their parameters and is not intended to specify implementation (e.g. via use of blocking function calls). It describes the input parameters in parentheses after the command name and the return values in front of the Command. This is only for descriptive purposes; the actual Command syntax and encoding are specified in later subsections. The order of parameters to commands is not fixed. Descriptors may appear as parameters to commands in any order. The descriptors SHALL be processed in the order in which they appear.

An error descriptor is a possible reply to any command, the API does not specifically show this.

All parameters enclosed by square brackets ([. . .]) are considered optional.

[End Correction]

6.34 Error in digit map activation

Description:	The current text in section 7.1.14.6 specifies that a digitmap is activated by means of an (possibly embedded) events descriptor that includes a digit map completion event, which itself contains a digit map parameter. A digit map completion event, however, cannot contain a digit map parameter. Section E.6.2 also specifies that a digit map parameter has to be present. It is more accurate to say that the events descriptor that requests detection of the digitmap completion event must contain an eventDM parameter.

[Begin Correction]

7.1.14.6 DigitMap Activation

A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, ~~and~~ that event descriptor contains a digit map completion event and the digit map completion event ~~which itself~~ contains an eventDM field in the requested actions field. ~~digit map parameter.~~ Each new activation of a digit map 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 digit map 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 EventsDescriptor, a MG shall respond with an error reponse, including error 457 – Missing parameter in signal or event.

[End Correction]

[Begin Correction]

Error code 457 – Missing parameter in signal or event has been added to H.248 Annex L Error Codes and Service Change reasons.

[End Correction]

[Begin Correction]

E.6.2 Events

…

EventsDescriptor parameters: None~~digit map processing is activated only if a digit map parameter is present, specifying a digit map by name or by value. Other parameters such as a KeepActive flag or embedded Events or Signals Descriptors may be present.~~

…

E.6.5 Procedures

~~None~~ Digit map processing is activated only if an events descriptor is activated that contains a digit map completion event as defined in Section E.6.2 and that digit map completion event contains an eventDM field in the requested actions as defined in Section 7.1.9. Other parameters such as KeepActive or embedded events of signals descriptors may also be present in the events descriptor and do not affect the activation of digit map processing.

[End Correction]

6.35 Use of wildcarded TerminationIDs in Add command

Description:	The text in Section 7.2.1/H.248 implies that a CHOOSE wildcard is used only in Add commands that create ephemeral terminations, and cannot be used to allow a MG to choose a particular physical Termination. Moreover, the text implies CHOOSE must be used to create ephemeral Terminations. Neither restriction is valid.

[Begin Correction]

7.2.1 Add

…

The TerminationID specifies the termination to be added to the Context. The Termination is either created, or taken from the null Context. ~~For an existing Termination, the TerminationID would be specific. For a Termination that does not yet exist, the TerminationID is specified as CHOOSE in the command.~~ If a CHOOSE wildcard is used in the TermintationID, the selected ~~The new~~TerminationID will be returned. Wildcards may be used in an Add, but such usage would be unusual. If the wildcard matches more than one

TerminationID, all possible matches are attempted, with results reported for each one. The order of attempts when multiple TerminationIDs match is not specified.

…

[End Correction]

6.36 Meaning of Transaction replies

Description:	It is unclear when Transaction replies are sent, in particular in the presence of commands that activate signals. Is the reply sent when • the signals have finished, • the signals have been activated, or • when the signals have been queued for activation? The intention is that the reply is sent when the signals have been queued for activation, implying that the signals descriptor was syntactically correct and only supported signals were requested.

[Begin Correction]

Section 8.2.2 TransactionReply

The TransactionReply is invoked by the receiver. There is one reply invocation per transaction. A reply contains one or more Actions, each of which must specify its target Context and one or more Responses per Context. The TransactionReply is invoked by the command responder when it has processed the TransactionRequest.

A TransactionRequest has been processed

- when all commands in that TransactionRequest have been processed, or
- when an error is encountered in processing a non-optional command in that TransactionRequest.

A command has been processed when all descriptors in that command have been processed.

A SignalsDescriptor is considered to have been processed when it has been established that the descriptor is syntactically valid, the requested signals are supported and they have been queued to be played out.

An EventsDescriptor or EventBufferDescriptor is considered to have been processed when it has been established that the descriptor is syntactically valid, the requested events can be observed, any embedded signals can be generated, any embedded events can be detected, and the MG has been brought in a state in which the events will be detected.

…

[End Correction]

6.37 Empty Action requests

Description:	The syntax specification in Annex B/H.248 forbids actions that are completely empty. In particular, an Action has to contain at least a command, a context modification request or a context audit request. The text in Section 8 does not reflect this.

[Begin Correction]

8. Transactions

Commands between the Media Gateway Controller and the Media Gateway are grouped into Transactions, each of which is identified by a TransactionID. Transactions consist of one or more Actions. An Action consists of a non-empty series of Commands, Context property modifications, or Context property audits that are limited to operating within a single Context.

…

[End Correction]

6.38 Auditing list of TerminationIDs

Description:	The protocol contains syntax to allow a MGC to audit which Terminations are in a Context. The relevant clauses in the binary and text encodings are `contextAuditResult` and `contextTerminationAudit`. The intention that the binary and text versions have the same functionality is not met in this case.

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
AuditReply ::= ~~SEQUENCE~~CHOICE
{
      contextAuditResult       TerminationIDList,
      error                    ErrorDescriptor,
      ~~terminationID            TerminationID,~~
      auditResult              AuditResult,
      ...
}

AuditResult ::= SEQUENCE~~CHOICE~~
{
      ~~contextAuditResult       TerminationIDList,~~
      terminationID            TerminationID,
      terminationAuditResult   TerminationAudit
}

TerminationAudit ::= SEQUENCE OF AuditReturnParameter
```

…

[End Correction]

6.39 Handoff in case of MGC failure

Description:	Section 11.5/H.248 contains procedures to be followed by MGs in case of MGC failure. The scenario addressed in the second paragraph of this section states that a MG that does not receive an Audit request after having established a control relationship with a backup MGC, acts as if this backup MGC failed. This imposes restrictions on MGC behavior that are unnecessary. For instance, the backup MGC could be a hot standby that does not need to audit the MG when it takes over control. Therefore we propose striking the clause stating this. Furthermore, the text states that the MG *should* follow its controlling MGC's Handoff request.

[Begin Correction]

11.5 Failure of an MG

…

In partial failure, or manual maintenance reasons, an MGC may wish to direct its controlled MGs to use a different MGC. To do so, it sends a ServiceChange method to the MG with a "HandOff" method, and its designated replacement in ServiceChangeMgcId. If "Handoff" is supported t~~T~~he MG ~~should~~shall send a

ServiceChange message with a "Handoff" method and a "MGC directed change" reason to the designated MGC. If it fails to get a reply from the designated MGC, ~~or fails to see an Audit command subsequently, it~~ the MG shall~~should~~ behave as if its MGC failed, and start contacting secondary MGCs as specified in the previous paragraph. If the MG is unable to establish a control relationship with any MGC, it shall wait a random amount of time as described in section 9.2 and then start contacting its primary, and if necessary, its secondary MGCs again.

[End Correction]

6.40 Syntax for signal and event parameters in Annex A.2

Description:	Section A.2 contains a clause defining EventParameters as a SEQUENCE consisting of an eventParameterName followed by a value. The type of the value is defined as OCTET STRING. In order to support lists of values mentioned in Section 12.2/H.248, the type of the value field has to be changed to SEQUENCE OF OCTET STRING.

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
TimeNotation ::= SEQUENCE
{
     date        IA5String(SIZE(8)),  -- yyyymmdd format
     time        IA5String(SIZE(8))   -- hhmmssss format
}

Value ::= SEQUENCE OF OCTET STRING


END
```

[End Correction]

6.41 Definition of PathName in Annex A.3

Description:	Annex A states that it reproduces the definition of PathName of Annex B. However, the definition presented there is not the same as that provided in Annex B. The appropriate text from Annex B should be copied to Annex A, replacing the current definition given in A.3.

[Begin Correction]

A.3 Digit maps and path names

…

A path name is also a string with syntactic restrictions imposed upon it. The ABNF production defining it is copied from Annex B.

~~PathName = NAME *(["/"] ["*"] ["@"] (ALPHA / DIGIT)) ["*"]~~

```
; Total length of pathNAME must not exceed 64 chars.
pathNAME                 = ["*"] NAME *("/" / "*"/ ALPHA / DIGIT /"_" / "$" )
                           ["@" pathDomainName ]
```

```
; ABNF allows two or more consecutive "." although it is meaningless
; in a path domain name.
pathDomainName        = (ALPHA / DIGIT / "*" )
                        *63(ALPHA / DIGIT / "-" / "*" / ".")

NAME                  = ALPHA *63(ALPHA / DIGIT / "_" )
```

[End Correction]

6.42 Packaged Name in Modem Descriptor in ABNF

Description:	The ASN.1 Modem Descriptor contains a sequence of Modem parameters of format Packaged Name. The ABNF only contains NAME which does not allow for package definition. Package definition should be allowed.

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
modemDescriptor       = ModemToken (( EQUAL modemType) /
                        (LSBRKT modemType *(COMMA modemType) RSBRKT))
                        [ LBRKT propertyParm~~NAME parmvalue~~
                       *(COMMA propertyParm~~NAME parmvalue~~) RBRKT ]
```

[End Correction]

6.43 Error descriptor in Notify request

Description:	Recommendation H.248 allows Notify requests to contain error descriptors. The recommendation does not specify under which circumstances error descriptors are to be included in Notify requests. One case where this is useful is in a Notify request that contains the generic error event defined in Annex E.1.2. This is used in the case when a general error event is triggered.

[Begin Correction]

7.2.7 Notify

…

The RequestID returns the RequestID parameter of the EventsDescriptor that triggered the Notify Command. It is used to correlate the notification with the request that triggered it. The events in the list must have been requested via the triggering EventsDescriptor or embedded events descriptor unless the RequestID is 0 (which is for further study).

The ErrorDescriptor may be sent in the Notify as a result of Error 518 - Event buffer full.

[End Correction]

6.44 Octet strings in text encoding

Description:	Sometimes it is desirable to transfer octet strings between MG and MGC. The definition of octet string in Annex B is not general enough because it is essentially a text string. Not all characters are allowed in text strings. The null character (0x00) is an example of a character that is not allowed in a text string.

	A solution to this problem is to use a standard way of encoding the octet string into a text string. Prescribing one way to be used in all cases facilitates uniform encoding and decoding routines. Another problem with the current definition of octetString in Annex B/H.248 is the fact that opening and closing braces must be escaped (i.e. preceded by a backslash). This contradicts the provision in section 7.1.8 that SDP session descriptions conformant to RFC 2327 must be accepted.

[Begin Correction]

Section Annex B Text encoding of the protocol (Normative)

…

B.3 Hexadecimal octet coding:

Hexadecimal octet coding is a means for representing a string of octets as a string of hexadecimal digits, with two digits representing each octet. This octet encoding should be used when encoding octet strings 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

B.4 Hexadecimal octet sequence:

A hexadecimal octet sequence is an even number of hexadecimal digits, terminated by a <CR> character.

[End Correction]

6.45 Annex C USI Correction

Description:	H.248 Annex C lists tag 9008 as being for the USI. However the values only represent the User Information Layer 1. The whole USI value should be supported and the layer 1 protocol.

[Begin Correction]

Annex C.9 Bearer Capabilities

…

layer1prot~~USI~~	9008	5 BITS	User Information Layer 1 Protocol Reference: ITU Recommendation Q.931 Bits 5 4 3 2 1 00001 – CCITT standardized rate adaption V.110 and X.30. 00010 - Recommendation G.711 u-law 00011 - Recommendation G.711 A-law 00100 – Recommendation G.721 32 kbit/s ADPCM and Recommendation I.460.

			00101 - Recommendations H.221 and H.242 00110 – Recommendations H.223 and H.245 00111 – Non-ITU-T standardized rate adaption. 01000 – ITU-T standardized rate adaption V.120. 01001 – CCITT standardized rate adaption X.31 HDLC flag stuffing. All other values are reserved.

...

USI	9023	OCTET STRING	User Service Information Reference ITU Recommendation Q.763 Section 3.57

[End Correction]

6.46 Encoding of ASN.1 Octet String

Description:	It is ambigious how Integer, boolean etc is encoded in the ASN.1 when OCTET STRING is used to encode properties.
Reference:	Subject: Re: OCTET STRINGs in H.248 Date: Wed, 8 Nov 2000 14:25:29 –0500 From: Troy Cauble <troy@LUCENT.COM> To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

A.2 ASN.1 Syntax Specification

...

NOTE – The ASN.1 specification below contains a clause defining `TerminationIDList` as a sequence of TerminationIDs. The length of this sequence SHALL be one, except possibly when used in contextAuditResult.

NOTE – The ASN.1 in this section uses OCTET STRING to encode values for property parameter, signal parameter and event parameter values and statistics. The actual types of these values vary and are specified in Annex C or the relevant package definition.
A value is first ASN.1 BER encoded based on it's type using the table below. The result of this ASN.1 BER encoding is then encoded as an ASN.1 BER OCTET STRING, "double wrapping" the value. The format specified in Annex C or the package relates to ASN.1 BER encoding according to the following table:

Type Specified in Package	**ASN.1 BER Type**
String (UTF-8)	IA5String
Integer (4 Octet)	INTEGER
Double (8 octet signed int)	INTEGER (*Note 3*)
Character (UTF-8, *Note 1*)	IA5String
Enumeration	ENUM
Boolean	BOOLEAN
Unsigned Integer (*Note 2*)	INTEGER (*Note 3*)
Note 1 : Can be more than one byte	
Note 2 : Unsigned integer is referenced in Annex C	
Note 3 : ASN.1 BER encoding of INTEGER does not imply the use of 4 bytes.	

See A.7 X.690 for definition of encoding of Octet String value.

```
MEDIA-GATEWAY-CONTROL DEFINITIONS AUTOMATIC TAGS::=
BEGIN
```

...

[End Correction]

6.47 Property and descriptor values of subtracted physical Terminations

Description:	The text of H.248 states that only property values in TerminationState and LocalControl descriptors remain visible when a physical Termination is subtracted, unless provisioned otherwise. More specifically, Section 7.2.3/H.248 states that the property values revert to provisioned values or, if no value is provisioned, the default value specified for that property. Section 6.2.4 states that all descriptors except TerminationState and LocalControl revert to empty/"no value" when a physical Termination is returned to the NULL context or when it is first created, except when provisioned otherwise. H.248 packages (should) include default values for properties they define, or specify that the default value is provisioned. Hence, a MGC that supports a particular package has knowledge of the property values that result from subtracting a termination. However, an MGC does not know what the provisioned values of properties in descriptors other than TerminationState and LocalControl are. This may lead to interoperability problems, unless the MGC audits all physical Terminations after cold boot and finds out about the provisioned property values.
Reference:	Editor

[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 standard or in a package (see Section 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. The default contents of the two exceptions are described in sections 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 standard or 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.

…

[End Correction]

6.48 Problem with syntax of auditOther

Description:	In the text of section 7.2.5 on AuditValue, the following can be found: "Specifying an empty Audit Descriptor results in only the TerminationID being returned." In the syntax in Annex B, this would be achieved by using the auditOther construction. However, the syntax is: auditOther = EQUAL TerminationID LBRKT terminationAudit RBRKT In order to return only the TerminationID, the syntax should be: auditOther = EQUAL TerminationID [LBRKT terminationAudit RBRKT] The optionality is shown correctly in the ASN.1 syntax.
Reference:	Subject: Problem with syntax of auditOther Date: Fri, 17 Nov 2000 17:43:28 -0500 From: "Brown, Michael" <C.Michael.Brown@AMERICASM10.NT.COM> Organization: Nortel Networks

	To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

B.2 ABNF Syntax Specification

...

```
      auditOther            = EQUAL TerminationID [LBRKT
                                 terminationAudit RBRKT]
```

...

[End Correction]

6.49 Stream Mode default

Description:	When adding a termination to a context the stream mode should be set to inactive unless specified overwise. This is not mentioned in the text. It is also unclear
Reference:	Subject: Re: H.248 Mode Parameter Date: Fri, 8 Dec 2000 13:23:09 –0500 From: "Taylor, Tom-PT [NORSE:B901:EXCH]" <taylor@AMERICASM01.NT.COM> To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[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 section 7.1.1.

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.

…

[End Correction]

6.50 Optional Stream in ASN.1

Description:	The specification of a stream in the media descriptor is optional. The ASN.1 does not allow for this optionality and should be updated.
Reference:	Subject: MEGACO: MediaDescriptor question Date: Thu, 14 Dec 2000 22:42:47 -0500 From: Troy Cauble <troy@LUCENT.COM> To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
MediaDescriptor ::= SEQUENCE
{

      termStateDescr    TerminationStateDescriptor OPTIONAL,
      streams            CHOICE
                         {
                               oneStream   StreamParms,
                               multiStream SEQUENCE OF StreamDescriptor
                         } OPTIONAL,
      ...
}
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
; at-most-once per item
; and may include either streamParm or streamDescriptor but not both
mediaParm            = (streamParm / streamDescriptor /
                        terminationStateDescriptor)
```

…

[End Correction]

6.51 embedSig/EmbedSig

Description:	Although correct, "EmbedSig" should be replaced with "embedSig" in order to be stylistically consistent and therefore avoid confusion.
Reference:	Subject: Re: ABNF semantics Date: Fri, 12 Jan 2001 15:48:34 –0500 From: "Rosen, Brian" <Brian.Rosen@MARCONI.COM> To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
secondEventParameter = ( e~~E~~mbedSig / KeepActiveToken / eventDM /
                          eventStream / eventOther )
```

…

[End Correction]

6.52 ABNF syntax error of Local Control Descriptor, Modem and Termination State Descriptor

Description:	According to the ASN.1 syntax the property parameter is SEQUENCE, it can be only one as per the ABNF syntax. A comment "at-most-once per item, except for propertyParm" can be added to solve this issue.
Reference:	Subject: RE: ABNF syntax error of LocalControlDescriptor and TerminationStateDescriptor Date: Thu, 4 Jan 2001 21:13:25 -0500 From: "Tom-PT Taylor" <taylor@nortelnetworks.com>

	To: Richie Wu <wuyongji@YAHOO.COM>, megaco@standards.nortelnetworks.com

[Begin Correction]

B.2 ABNF Syntax Specification

```
                                …
localControlDescriptor = LocalControlToken LBRKT localParm
                         *(COMMA localParm) RBRKT

; at-most-once per item except for propertyParm
localParm             = ( streamMode / propertyParm / reservedValueMode
                       / reservedGroupMode )
                                …
terminationStateDescriptor = TerminationStateToken LBRKT
            terminationStateParm *( COMMA terminationStateParm ) RBRKT

; at-most-once per item except for propertyParm
terminationStateParm =(propertyParm / serviceStates / eventBufferControl )
                                …
; at-most-once except for extensionParameter
modemType             = (V32bisToken / V22bisToken / V18Token /
                        V22Token / V32Token / V34Token / V90Token /
                        V91Token / SynchISDNToken / extensionParameter)
                                …
```

[End Correction]

6.53 Wilcarding Context IDs

Description:	Wildcarding ContextIDs is allowed however it is poorly specified in the current specification.
Reference:	Subject: Re: Wildcard in ContextId Date: Fri, 12 Jan 2001 15:14:46 –0500 From: Michael Brown C.Michael.Brown@NORTELNETWORKS.COM Organization: Nortel Networks To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

6.1.1 Context Attributes and Descriptors

The attributes of Contexts are:

- ContextID. A wildcarding mechanism using two types of wildcards can be used with ContextIDs. The two wildcards are ALL and CHOOSE. The former is used to address ALL (except the NULL context) Contexts at once in a command request and/or reply, while the latter is used to indicate to a media gateway that it must create a Context.

…

[End Correction]

6.54 ABNF is Context dependent

Description:	It is not clear whether the ABNF is context dependent.
Reference:	Subject: context-free? Date: Wed, 13 Dec 2000 17:54:32 –0600 Reply-To: plong@ipdialog.com From: Paul Long <plong@packetizer.com> Organization: ipDIalog, Inc. To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

B.2 ABNF Syntax Specification

The protocol syntax is presented in ABNF according to RFC2234.

Note - The syntax is context-dependent. For example, "Add" can be the AddToken or a NAME depending on the context in which it occurs.

…

[End Correction]

6.55 Double quote not allowed in quotedString and Empty Quote allowed.

Description:	Reiterate that the double-quote character is not allowed in a quotedString. Empty quotedString is allowed as, section E.6.2 says under parameter "ds", "string of digit map symbols (possibly empty) returned as a quoted string".
Reference:	Subject: Re: Double quote not allowed Date: Tue, 19 Dec 2000 16:54:21 -0600 Reply-To: plong@ipdialog.com From: Paul Long <plong@packetizer.com> Organization: ipDialog, Inc. To: MEGACO@STANDARDS.NORTELNETWORKS.COM Subject: RE: empty quotedString ? Date: Thu, 10 May 2001 16:30:42 -0500 From: "Paul Long" plong@ipdialog.com To: "MEGACO list" <megaco@fore.com>

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
;Note - The double-quote character is not allowed in quotedString.
quotedString           = DQUOTE ~~1~~*(SafeChar / RestChar/ WSP) DQUOTE
```

…

[End Correction]

6.56 Ranges and Multiple Values for Signal and Event Parameter Values

Description:	The ABNF specification allows the specification of multiples values and ranges for a signal and event parameter value. The ASN.1 currently does not and needs to be aligned.

Reference:	Subject: Re: ASN.1 - Values in Event and Signal Parameter Date: Mon, 15 Jan 2001 12:42:25 +0530 From: Sandeep Gautam <sgautam@HSS.HNS.COM> To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
EventParameter ::= SEQUENCE
{
      eventParameterName       Name,
      value                    Value,
-- For use of extraInfo see the comment related to PropertyParm
      extraInfo CHOICE
      {
            relation    Relation,
            range       BOOLEAN,
            sublist     BOOLEAN
      } OPTIONAL,
      ...
}
```

…

```
SigParameter ::= SEQUENCE
{
      sigParameterName         Name,
      value                    Value,
-- For use of extraInfo see the comment related to PropertyParm
      extraInfo CHOICE
      {
            relation    Relation,
            range       BOOLEAN,
            sublist     BOOLEAN
      } OPTIONAL,
      ...
}
```

…

[End Correction]

6.57 TimeStamp in registration replies

Description:	There is a mismatch between the text in section 7.2.8 which states " The TimeStamp parameter shall be sent with a registration command and its response." and the ASN.1 and ABNF syntax which only allows it in the request. There is confusion regarding whether the timeStamps in registration requests and replies are relative or absolute.
Reference:	Subject: TimeStamp in registration replies Date: Mon, 15 Jan 2001 13:33:19 –0500 From: Terry L Anderson <tla@LUCENT.COM> Organization: Lucent Technologies To: MEGACO@STANDARDS.NORTELNETWORKS.COM Subject: RE: "Does anybody really know what time it is?" :-) Date: Wed, 2 May 2001 13:30:37 –0500

	From: "Paul Long" <plong@ipdialog.com> To: "'Megaco \(E-mail\)'" <megaco@fore.com>

[Begin Clarification]

7.2.8 Service Change

…

The optional TimeStamp parameter specifies the actual time as kept by the sender. As such, it is not necessarily absolute time according to, for example, a local time zone—it merely establishes an arbitrary starting time against which all future timestamps transmitted by a sender during this association shall be compared. It can be used by the responder to determine how its notion of time differs from that of its correspondent. TimeStamp is sent with a precision of hundredths of a second, and is expressed in UTC.

…

[End Clarification]

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
ServiceChangeResParm ::= SEQUENCE
{
      serviceChangeMgcId      MId OPTIONAL,
      serviceChangeAddress    ServiceChangeAddress OPTIONAL,
      serviceChangeVersion    INTEGER(0..99) OPTIONAL,
      serviceChangeProfile    ServiceChangeProfile OPTIONAL,
      timestamp               TimeNotation OPTIONAL,
      ...
}
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
servChgReplyParm     = (serviceChangeAddress / serviceChangeMgcId /
                       serviceChangeProfile / serviceChangeVersion /
                       TimeStamp)
```

…

[End Correction]

6.58 ABNF Token for Signals and Events overlap with packages

Description:	In the ABNF it is possible to define new tokens for Signal and Event information elements. Ie.
	`sigParameter = sigStream / sigSignalType / sigDuration`
	`               / sigOther / notifyCompletion / KeepActiveToken`
	The package identity can be contained in sigOther. The problem lies in the fact that in the

	future that the introduction of a new Token for a signal Parameter may cause overlap with an existing package identity. This would lead to an ambiguous interpretation. This problem relates to both signals and events. The solution below limits the any new sigParameter or eventParameter Tokens to start with a certain prefix. An update is also made to the package definition rules saying that packages identiities cannot start with this prefix.

[Begin Correction]

B.2 ABNF Syntax Specification

...

```
RestChar                 = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
                           "<" / ">" / "="

; New Tokens added to sigParameter must take the format of SPA*
; * may be of any form ie. SPAM
; New Tokens added to eventParameter must take the form of EPA*
; * may be of any form ie. EPAD

AddToken                       = ("Add"                   / "A")
```

[End Correction]

[Begin Correction]

12.2 Guidelines to defining Properties, Statistics and Parameters to Events and Signals

Parameter Name: only descriptive
ParameterID: Is an identifier. The textual ParameterID of parameters to Events and Signals shall not start with "EPA" and "SPA", respectively. The textual ParameterID shall also not be "ST", "Stream", "SY", "SignalType", "DR", "Duration", "NC", "NotifyCompletion", "KA", "Keepactive", "EB", "Embed", "DM" or "DigitMap".

[End Correction]

6.59 Aligning Error Code Description with Annex L

Description:	In the development of H.248 Annex L Service Change Reason and Error Code definition it was identified that the definition of several of the codes in 7.3 were incomplete. These should be aligned.
Reference:	Editor

[Begin Correction]

7.3 Command Error Codes

Errors consist of an IANA registered error code and an explanatory string. Sending the explanatory string is optional. Implementations are encouraged to append diagnostic information to the end of the string.

When a MG reports an error to a MGC, it does so in an error descriptor. An error descriptor consists of an error code and optionally the associated explanatory string.

H.248 Annex L contains the error codes supported by H.248.

~~The identified error codes are:~~

~~400 - Syntax error in message~~

~~...~~

~~403 - Syntax Error in Transaction~~

~~…~~

~~441 - Missing RemoteDescriptor~~

~~…~~

~~443 - Unknown Command~~

~~…~~

~~455 - Parameter illegal in this Descriptor~~

~~…~~

~~456 - Parameter or Property appears twice in this Descriptor~~

~~…~~

~~500 – Internal Gateway Error~~

~~…~~

~~505 – Command Received before Restart Response~~

~~…~~

~~520 - Media Gateway does not have a digit map~~

~~…~~

~~529 – Internal hardware failure~~

…

[End Correction]

6.60 Termination ID in Notify Reply

Description:	The API does not show a notify reply returning a termination ID. However the ASN.1 and ABNF return the terminationID. The API should be updated. The ASN.1 also shows that the return is optional however the ABNF shows the return as manadatory. The return should be mandatory in line with the other commands.
Reference:	Editor

[Begin Correction]

7.2.7 Notify

The Notify Command allows the Media Gateway to notify the Media Gateway Controller of events occurring within the Media Gateway.

TerminationID

```
        Notify(TerminationID,
              ObservedEventsDescriptor,
                        [ErrorDescriptor]
```

…

[End Correction]

[Begin Correction]

A.2 ASN.1 Syntax Specification

```
        NotifyReply ::= SEQUENCE
        {
                terminationID                   TerminationIDList ~~OPTIONAL~~,
                errorDescriptor                 ErrorDescriptor OPTIONAL,
                ...
        }
```

…

[End Correction]

6.61 missing Descriptor vs. missing properties in a Descriptor

Description:	The issue is that when a MGC sends a Command to the MG, it makes a big difference whether a Descriptor is completely omitted from the Command, or only some properties are omitted from an existing Descriptor. At the moment this is not well explained in section 6.2.4/H.248, as it does not clearly distinguish between a "Descriptor" and a "Property". In addition, sections 6.2.4 and 7.1.x are contradicting each other, viz. where the sections 7.1.x state explicitly that a new setting of a Descriptor completely replaces the previous setting of that Descriptor – see e.g. the last para of section 7.1.6/H.248. The matter can be resolved by changing the 4th paragraph of section 6.2.4/H.248 as follows:
Reference:	Editor

[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. ~~Properties not mentioned in the command retain their prior values.~~ Similarly, a property of a Termination in a Context may have its value changed by the Modify command. ~~Properties not mentioned in the Modify command retain their prior values.~~ 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) the Command acts on. On the other hand, if some properties are omitted from a Descriptor in a Command – i.e., the Descriptor is only partially specified – those properties will be removed/reset for the Termination(s) the Command acts on. For more details, see section 7.1 dealing with the individual Descriptors.

…

[End Correction]

6.62 Redundant MGs

Description:	
Reference:	Subject: Re: FW: Redundant MG's Date: Wed, 31 Jan 2001 08:04:04 -0500 From: "Rosen, Brian" <Brian.Rosen@MARCONI.COM>

[Begin Correction]

11.4 Failure of an MG

…

Allowing the MGC to send duplicate messages to both MGs accommodates pairs of MGs that are capable of redundant failover of one of the MGs. Only the Working MG shall accept or reject transactions. Upon failover, the Primary MG sends a ServiceChange command with a "Failover" method and a "MG Impending Failure" reason. The MGC then uses the secondary MG as the active MG. When the error condition is repaired, the Working MG can send a "ServiceChange" with a "Restart" method. Redundant failover MGs with the current protocol definition requires a reliable transport, and knowledge in the MGC of the redundancy at the MG.

[End Correction]

6.63 Use of MID

Description:	There is some confusion on the usage of MID and how it relates to the control association between a MGC and MG.
Reference:	Subject: RE: Use of MID: Implementors Guide Issue Date: Tue, 13 Feb 2001 16:09:38 -0800 From: "Kaul, Bharat" <bharat@trillium.com> To: "'Christian Groves'" <Christian.Groves@ericsson.com>, "Kaul, Bharat"

[Begin Correction]

8.3 Messages
Multiple Transactions can be concatenated into a Message. Messages have a header, which includes the identity of the sender. The Message Identifier (MID) of a message is set to a provisioned name (e.g. domain address/domain name/device name) of the entity transmitting the message. Domain name is a suggested default. An H.248 entity (MG/MGC) must consistently use the same MID in all messages it originates for the duration of control association with the peer (MGC/MG).

…

[End Correction]

6.64 Typographical error in ASN.1 choose

Description:	There is a typographical error in the numerical value for CHOOSE in the ASN.1 description.
Reference:	Subject: Typo in H.248 standard. Date: Wed, 21 Mar 2001 08:56:14 -0700 (MST) From: Nattapong Mongkolnavin nm59@avaya.com To: Christian.Groves@ericsson.com

[Begin Correction]

A.2 ASN.1 Syntax Specification

...

-- Context NULL Value: 0
-- Context CHOOSE Value: 4294967294 (0xFFFFFFFE)

```
-- Context ALL Value: 4294967295 (0xFFFFFFFF)
```

...

[End Correction]

6.65 Events desciptor and eventDM

Description:	H.248 Implementors' Guide 6.38 introduces an inconsistency in H.248. Namely, section 7.1.14.8/H.248 implies it is possible to specify an EventsDescriptor comprising a Digit Map Completion event, and still omit the eventDM parameter. Section 7.1.14.8/H.248 should be changed to align with the change in approved H.248 Implementors' Guide 6.38.
Reference:	Editor

[Begin Correction]

7.1.14.8 Wildcards
Note that if a package contains a digit map completion event, then an event specification consisting of the package name with a wildcarded ItemID (Property Name) will activate a digit map– to that end the event specification must include an

eventDM field according to section 7.1.14.6.~~ if the event includes a digit map parameter. Regardless of whether a digit map is activated, i~~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.

[End Correction]

6.66 MTP Address and ServiceChange

Description:	The definition of the MTP address when used with a ServiceChange is different from the MTP address used for the MID. The usage should be consistent.
Reference:	Editor

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
ServiceChangeAddress ::= CHOICE
{
      portNumber                INTEGER(0..65535), -- TCP/UDP port number
      ip4Address                IP4Address,
      ip6Address                IP6Address,
      domainName                DomainName,
      deviceName                PathName,
      mtpAddress                OCTET STRING(SIZE(2..4)),
      ...
}
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
mId                  = (( domainAddress / domainName )
                         [":" portNumber]) / mtpAddress / deviceName
```

…

serviceChangeAddress = ServiceChangeAddressToken EQUAL (mId / portNumber)~~VALUE~~

…

[End Correction]

6.67 Optional Reserve in ASN.1

Description:	The procedural text and the ABNF specification state that reserve group and reserve property have a default and thus do not need to be set. The ASN.1 does not have this possibity. OPTIONAL should be added.
Reference:	Subject: RE: [Fwd: RE: Adding Streams] Date: Thu, 15 Mar 2001 16:27:44 –0600 (CST) From: David Dykhuizen <eusdbd@exu.ericsson.se> To: Christian.Groves@ericsson.com, Brian.Rosen@marconi.com

[Begin Correction]

A.2 ASN.1 Syntax Specification

...

```
LocalControlDescriptor ::= SEQUENCE
{
      streamMode        StreamMode OPTIONAL,
      reserveValue      BOOLEAN OPTIONAL,
      reserveGroup      BOOLEAN OPTIONAL,
      propertyParms     SEQUENCE OF PropertyParm,
      ...
}
```

...

[End Correction]

6.68 Transaction Acknowledgement Lists

Description:	As per the ABNF definition we can acknowledge a list of unsequential transaction, for example:
	TransactionResponseAck = { 1234, 123-456, 9, 78-100 }.
	But, as per ASN.1 syntax, only one transaction or a sequential range of transaction can be acknowledged at one time, viz. Either
	TransactionResponseAck = { 123} or
	TransactionResponseAck = {123-456}.
	These methods should be aligned.

Reference:	Subject: Confilict definitions of transaction response acknowledgement Date: Tue, 13 Mar 2001 23:45:13 -0800 (PST) From: Richie Wu wuyongji@yahoo.com To: megaco@fore.com

[Begin Correction]

A.2 ASN.1 syntax specification

...

```
TransactionResponseAck ::= SEQUENCE OF TransactionAck
TransactionAck   ::= SEQUENCE
{
      firstAck TransactionId,
      lastAck  TransactionId OPTIONAL
}
```

...

[End Correction]

6.69 Private Package ID registration

Description:	The IANA resgistration at < <http://www.isi.edu/in-notes/iana/assignments/megaco-h248> Differs from the registration procedure in the H.248 recommendation for package Ids for private packages. The recommendation should align with the IANA procedure.
Reference:	Editor

[Begin Correction]

13.1 Packages

The following considerations SHALL be met to register a package with IANA:

1. A unique string name, unique serial number and version number is registered for each package. The string name is used with text encoding. The serial number shall be used with binary encoding. Serial Numbers 0x8000 to 0xffff~~60000-64565~~ are reserved for private use. Serial number 0 is reserved.

…

[End Correction]

6.70 Gain Control property type

Description:	The gain control parameter in the "tdmc" package specifies the type to be "enumeration" when clearly it should be "integer".
Reference:	Subject: Gain control parameter Date: Thu, 8 Mar 2001 16:05:34 –0800 From: "Kumar, Sanjay1" <sanjayk@trillium.com> To: "'megaco@fore.com'" <megaco@fore.com>

[Begin Correction]

E.13.1 Properties

Gain Control

PropertyID: gain (0x000a)

Gain control, or usage of of signal level adaptation and noise level reduction is used to adapt the level of the signal. However, it is necessary, for example for modem calls, to turn off this function.

Type: ~~enumeration (~~integer~~)~~

Possible Values:

…

[End Correction]

6.71 Missing Characteristics in the base root package

Description:	The characteristics of properties in the base root package are missing. The characteristics should be present in a package.
Reference:	Editor

[Begin Correction]

…

E.2.1 Properties

MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

The value of this property gives the maximum number of contexts that can exist at any time. The NULL context is not included in this number.

Type: Double

Possible values: 1 and up

Defined in: TerminationState

Characteristics: read only

MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

The maximum number of allowed terminations in a context, see section 6.1

Type: Integer

Possible Values: any integer

Defined In: TerminationState

Characteristics: read only

normalMGExecutionTime

PropertyId: normalMGExecutionTime (0x0003)

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)

Type: Integer

Possible Values: any integer, represents milliseconds

Defined in: TerminationState

Characteristics: read / write

normalMGCExecutionTime

PropertyId: normalMGCExecutionTime (0x0004)

Settable by the MGC to indicate the interval within which the MG should expects a response to any transaction from the MGC (exclusive of network delay)

Type: Integer

Possible Values: any integer, represents milliseconds

Defined in: TerminationState

Characteristics: read / write

…

[End Correction]

6.72 Service Change address / Service Change MGCid mutual exclusion

Description:	Section 7.2.8 mentions "ServiceChangeAddress and ServiceChangeMgcId Parameters must not both be present in the ServiceChangeDescriptor or the ServicechangeResultDescriptor" If I understand this right, a comment on this mutual exclusion should be included in Annex B just like there is one in the case of eventParameter for embedWithSig / embedNoSig
Reference:	Subject: ServiceChangeAddress / ServiceChangeMgcId Date: Wed, 21 Feb 2001 16:35:39 -0500 From: Preeti Sharma <preetis@lucent.com> To: megaco@fore.com

[Begin Correction]

B.2 ABNF syntax specification

```
                                ...
;at most one of either serviceChangeAddress or serviceChangeMgcId but not both
serviceChangeParm     = (serviceChangeMethod / serviceChangeReason /
                       serviceChangeDelay / serviceChangeAddress /
                       serviceChangeProfile / extension / TimeStamp /
                       serviceChangeMgcId / serviceChangeVersion )

serviceChangeReplyDescriptor = ServicesToken LBRKT
                       servChgReplyParm *(COMMA servChgReplyParm) RBRKT

;at-most-once. Version is REQUIRED on first ServiceChange response
;at most one of either serviceChangeAddress or serviceChangeMgcId but not both
servChgReplyParm      = (serviceChangeAddress / serviceChangeMgcId /
                       serviceChangeProfile / serviceChangeVersion )
                                ...
```

[End Correction]

6.73 ROOT termination not in ALL

Description:	When wildcarding a termination with ALL this does not address the root termination. This is not documented clearly in the specification.
Reference:	Subject: Re: ROOT not in ALL Date: Tue, 20 Feb 2001 15:49:07 -0500 From: "Kevin Boyle" <kboyle@nortelnetworks.com> Organization: Nortel Networks To: Madhu Babu Brahmanapally <madhubabu@kenetec.com> CC: "'Rosen, Brian'" <Brian.Rosen@marconi.com>, megaco <megaco@fore.com>

[Begin Correction]

6.2.2 TerminationIDs

…

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.

…

[End Correction]

6.74 Case sensitivity of ABNF and text encoding

Description:	It is unclear whether or the ABNF and its text encoding is case sensistive. The ABNF is not case sensistive.
Reference:	Subject: Case insensitivity Date: Tue, 20 Feb 2001 15:07:11 -0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "'megaco@fore.com'" <megaco@fore.com>

[Begin Correction]

B.2 ABNF Syntax Specification
The protocol syntax is presented in ABNF according to RFC2234.

EVERYTHING IN THE ABNF AND TEXT ENCODING IS CASE INSENSITIVE. This includes TerminationIDs, digitmap Ids etc. THE SDP IS CASE SENSITIVE AS PER RFC2327.

…

[End Correction]

6.75 Multiple Media Descriptor parameters

Description:	The ABNF says the following: ; at-most-once per item ; and either streamParm or streamDescriptor but not both mediaParm = (streamParm / streamDescriptor / Someone interpreted the two comments to imply that you could only have ONE streamParm. This is not the intention, you can have one each (one per item). So you can say Media{Local{..},LocalControl{...}) but you cannot say Media{Local{..},Stream=1{LocalControl{..
Reference:	Subject: Media Descriptor Parameters Date: Mon, 19 Feb 2001 16:41:17 –0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "'megaco@fore.com'" <megaco@fore.com>

[Begin Correction]

7.1.4 Media Descriptor

…

As a convenience a LocalControl, Local, or Remote descriptors may be included in the Media Descriptor without an enclosing Stream descriptor. In this case, the StreamID is assumed to be 1.

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
; at-most-once per item
; using ~~and ~~either streamParms or streamDescriptors but not both
mediaParm            = (streamParm / streamDescriptor /
                        terminationStateDescriptor)
```

…

[End Correction]

6.76 Provisional Timer Response Value

Description:	There is an inconsistency in package H.248 E.2 in that it has two values for execution time ie. normalMGExecutionTime and normalMGCExecutionTime. It however only has one value to represent the ProvisionalResponseTimerValue. This provisionalResponseTimerValue could in fact be set to two different times based upon the execution timers, one for the MGC and one for the MG. A second timer should be introducted for the provisionalResponseTime so that the MGC and MG can send timers at a different time.
Reference:	Subject: Re: Provisional Timer Response Date: Tue, 27 Mar 2001 02:48:27 -0800 (PST) From: Richie Wu <wuyongji@yahoo.com> To: Christian Groves <Christian.Groves@ericsson.com>, megaco ietf <megaco@fore.com>

[Begin Correction]

E.2.1 Properties

MGProvisionalResponseTimerValue

PropertyId: MGProvisionalResponseTimerValue (0x0005)

Indicates the time within which the MGC should~~to~~ expect a Pending Response from the MG if a Transaction cannot be completed. Initially set to normalMGExecutionTime~~ or normalMGCExecutionTime as appropriate~~, plus network delay, but may be lowered.

Type: Integer

Possible Values: any integer, represents milliseconds

Defined in: TerminationState

Characteristics: read / write

MGCProvisionalResponseTimerValue

PropertyId: MGCProvisionalResponseTimerValue (0x0006)

Indicates the time within which the MG should expect a Pending Response from the MGC if a Transaction cannot be completed. Initially set to normalMGCExecutionTime, plus network delay, but may be lowered.

Type: Integer

Possible Values: any integer, represents milliseconds

Defined in: TerminationState

Characteristics: read / write

…

[End Correction]

6.77 Typographical Error in Event Descriptor

Description:	There is a typographical error in section 7.1.9 a Notify is sent to the MGC not the MG.
Reference:	Editor

[Begin Correction]

7.1.9 Events Descriptor

…

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 digit map (see section 7.1.14). Any other action is for further study. 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 section.

…

[End Correction]

6.78 Clearing Contexts in ServiceChange

Description:	There is confusion as to the handling on subtraction of terminations from a context when a ServiceChange with "forced" indication is used. When a ServiceChange "forced" is issued on a non-Root termination the MGC is responsible for subsequently subtracting the termination from the applicable context. When a ServiceChange "forced" is issued on the Root termination it is assumed that all connection are lost on the MG and thus the MGC can consider that all the terminations are subtracted.
Reference:	Subject: Re: Context Id in Service Change Date: Wed, 04 Apr 2001 19:31:17 -0400 From: "Kevin Boyle" <kboyle@nortelnetworks.com> Organization: Nortel Networks To: "Rosen, Brian" <Brian.Rosen@marconi.com> CC: "'Paul Rheaume'" <paul.rheaume@alcatel.com>, "'megaco@fore.com'" <megaco@fore.com>

[Begin Correction]

7.2.8 ServiceChange

…

2) Forced – indicates that the specified Terminations were taken abruptly out of service and any established connections associated with them may be~~were~~ lost. For non-Root terminations t~~T~~he MGC is responsible for cleaning up the context (if any) with which the failed termination is associated. At a minimum the termination shall be subtracted from the context. The termination serviceState should be "out of service". For the root termination the MGC can assume that all connections are lost on the MG and thus can consider that all the terminations have been subtracted.

…

[End Correction]

6.79 Cancelling Event Detection

Description:	The ASN.1 can say that there are no events to be monitored. The ABNF needs changing to align. The request ID should be omitted if there are no events. This has the advantage of reducing message size. It also follows the precedent set in section 7.1, which uses the descriptor token alone to denote an empty descriptor in transaction replies. It should be spelt out in section 7.1.9 what an empty Events or EventBuffer Descriptor

	means, and what happens to buffered events collected in LockStep mode when the new Events Descriptor is empty.
Reference:	Subject: RE: [Fwd: RE: Does the First Line of SDP need a newline] Date: Fri, 30 Mar 2001 09:23:30 –0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "'David Stonehouse'" <stonehouse@nortelnetworks.com>, "'Christian Groves'" <Christian.Groves@ericsson.com> CC: megaco ietf <megaco@fore.com>, Tom-PT Taylor <taylor@nortelnetworks.com>

[Begin Correction]

A.2 ASN.1 Syntax Specification

```
EventsDescriptor ::= SEQUENCE
{
      requestID         RequestID OPTIONAL,
                        -- RequestID must be present if eventList
                        -- is non empty
      eventList         SEQUENCE OF RequestedEvent,
      ...
}
```

…

```
SecondEventsDescriptor ::= SEQUENCE
{
      requestID               RequestID OPTIONAL,
      eventList               SEQUENCE OF SecondRequestedEvent,
      ...
}
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
eventsDescriptor     = EventsToken [ EQUAL RequestID LBRKT
                       requestedEvent *( COMMA requestedEvent ) RBRKT ]
```

…

```
; at-most-once of each
embedFirst      = EventsToken [ EQUAL RequestID LBRKT
               secondRequestedEvent *(COMMA secondRequestedEvent) RBRKT ]
```

…

```
eventBufferDescriptor= EventBufferToken [ LBRKT eventSpec
                       *( COMMA eventSpec) RBRKT ]
```

…

[End Correction]

[Begin Correction]

7.1.9 Events Descriptor

The EventsDescriptor parameter contains a RequestIdentifier and a list of events that the Media Gateway is requested to detect and report. The RequestIdentifier 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 RequestIdentifier is omitted if the EventsDescriptor is empty (i.e. no events are specified).

…

An EventsDescriptor 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 EventsDescriptor executes, shall be processed according to the new EventsDescriptor.

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 events buffer is immediately cleared.

…

[End Correction]

6.80 Encoding of ABNF Value construct

Description:	It is unclear in the ABNF what may be included in the different ABNF values constructs. This correction gives guidance on the allowable fields.
Reference:	Subject: [FINAL;)] Re: encoding package values as ANBF VALUES Date: Thu, 26 Apr 2001 18:32:56 -0400 From: Troy Cauble <troy@lucent.com> Reply-To: Troy Cauble <troy@bell-labs.com> To: MEGACO list <megaco@fore.com>, Christian Groves <Christian.Groves@ericsson.com>

[Begin Correction]

B.2 ABNF Syntax Specification

The protocol syntax is presented in ABNF according to RFC2234.

~~; Boolean values, indicated in the text as True and False, are~~
~~; encoded as "On" and "Off", respectively, in the ABNF.~~

; NOTE -- The ABNF in this section uses the VALUE construct (or lists of
; VALUE constructs) to encode various package element values (properties,
; signal parameters, etc.). The types of these values vary and are
; specified the relevant package definition. Several such types are
; described in section 12.2.
;
; The ABNF specification for VALUE allows a quotedString form or a
; collection of SafeChars. The encoding of package element values into
; ABNF VALUES is specified below. If a type's encoding allows characters
; other than SafeChars, the quotedString form MUST be used for all values
; of that type, even for specific values that consist only of SafeChars.
;
; String: A string MUST use the quotedString form of VALUE and can
; contain anything allowable in the quotedString form.
;
; Integer, Double, and Unsigned Integer: Decimal values can be encoded
; using characters 0-9. Hexadecimal values must be prefixed with '0x'
; and can use characters 0-9,a-f,A-F. An octal format is not supported.
; Negative integers start with '-' and MUST be Decimal. The SafeChar
; form of VALUE MUST be used.
;
; Character: A UTF-8 encoding of a single letter surrounded by double
; quotes.
;
; Enumeration: An enumeration MUST use the SafeChar form of VALUE
; and can contain anything allowable in the SafeChar form.
;
; Boolean: Boolean values are encoded as "on" and "off" and are

; case insensitive. The SafeChar form of VALUE MUST be used.
;
; Future types: It is expected that packages will define types
; beyond these initial types. Any defined types MUST fit within
; the ANBF specification of VALUE. Specifically, if a type's encoding
; allows characters other than SafeChars, the quotedString form MUST
; be used for all values of that type, even for specific values that
; consist only of SafeChars.
;
; Note that there is no way to use the double quote character within
; a value.

…

[End Correction]

6.81 Alignment of Failover text between 7.2.8 and 11.5

Description:	There is discrepancy between the text with regards to Failover between sections 7.2.8 Service Change and 11.5 Failure of an MGC.
Reference:	Subject: RE: Reset of an MGC Date: Wed, 11 Apr 2001 15:02:31 –0400 From: "Tom-PT Taylor" <taylor@nortelnetworks.com> To: "'Ian Leighton'" <ian.leighton@alcatel.com>, megaco@fore.com

[Begin Correction]

11.5 Failure of an MGC
If the MG detects a failure of its controlling MGC, it attempts to contact the next MGC on its pre-provisioned list. It starts its attempts at the beginning (Primary MGC), unless that was the MGC that failed, in which case it starts at its first Secondary MGC. It sends a ServiceChange message with a "Failover" method and a " MGC Impending Failure" reason. If the MG is unable to establish a control relationship with any MGC, it shall wait a random amount of time as described in section 9.2 and then start contacting its primary, and if necessary, its secondary MGCs again.

…

[End Correction]

[Begin Correction]

7.2.8 Service Change
6) Failover – sent from MG to MGC to indicate the primary MG is out of service and a secondary MG is taking over. This serviceChange method is also sent from the MG to the MGC when the MG detects that MGC has failed.

…

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....

…

[End Correction]

6.82 ALF not defined

Description:	The ALF acronym needs to be defined.
Reference:	Subject: ALF? Date: Mon, 7 May 2001 09:08:44 -0400 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: megaco@fore.com

[Begin Correction]

4. ABBREVIATIONS

This recommendation defines the following terms.

```
ALF          Application Layer Framing
```

. . .

[End Correction]

[Begin Correction]

D.1 Transport over IP/UDP using Application Level Framing

Protocol messages defined in this document may be transmitted over UDP. When no port is provided by the peer (see section 7.2.8), commands should be sent to the default port number, 2944 for text-encoded operation or 2945 for binary-encoded operation. Responses must be sent to the address and port from which the corresponding commands were sent.

ALF is a set of techniques that allow an application, as opposed to a stack, to affect how messages are sent to the other side. A typical ALF technique is to allow an application to change the order of messages sent when there is a queue AFTER it has queued them. There is no formal specification for ALF. The procedures in Annex D.1 contain a minimum suggested set of ALF behaviors

. . .

[End Correction]

6.83 Source address

Description:	We need to be explicit about what a source address is.
Reference:	Subject: RE: From whence they came? Date: Thu, 10 May 2001 14:49:30 -0400 From: "Tom-PT Taylor" <taylor@nortelnetworks.com> To: megaco <megaco@fore.com>

[Begin Correction]

9. TRANSPORT

. . .

As described in section 7.2.8, either the MG or the MGC may supply an address in the ServiceChangeAddress parameter to which subsequent transaction requests must be addressed, but responses (including the response to the initial ServiceChange request) must always be sent back to the address which was the source of the corresponding request. For example: in IP networks, this is the source address in the IP header and the source port number in the TCP/UDP/STCP header.

. . .

[End Correction]

6.84 Command, not Action

Description:	Cut-and-paste mistake most likely resulted in typographical error where “Actions” is specified rather than “Commands.”
Reference:	Subject: RE: Question regarding formulating transaction replies. Date: Fri, 27 Apr 2001 20:32:46 -0500 From: Paul Long <plong@ipdialog.com> To: "Megaco Mailing List \(E-mail\)" <megaco@fore.com>

[Begin Correction]

8.2.2 TransactionReply

. . .

If the receiver encounters an error such that it cannot determine a legal Action, it will return a TransactionReply consisting of the TransactionID and a single error descriptor, 422 Syntax Error in Action. If the end of an action cannot be reliably determined but one or more ~~Actions~~Commands can be parsed, it will process them and then send 422 Syntax Error in Action as the last action for the transaction. If the receiver encounters an error such that is cannot determine a legal Transaction, it will return a TransactionReply with a null TransactionID and a single error descriptor (403 Syntax Error in Transaction).

. . .

[End Correction]

6.85 Message is a transport mechanism

Description:	Transactions are transported independently within messages.
Reference:	Subject: RE: Correspondence between requests and replies in messages Date: Mon, 30 Apr 2001 12:53:42 -0700 From: "Tulpule, Naren" <naren@trillium.com> To: <megaco@fore.com>

[Begin Clarification]

8.3 Messages

. . .

The transactions in a message are treated independently. There is no order implied; there is no application or protocol acknowledgement of a message. A message is essentially a transport mechanism. For example, message X containing transaction requests A, B, and C may be responded to with message Y containing replies to A and C and message Z containing the reply to B. Likewise, message L containing request D and message M containing request E may be responded to with message N containing replies to both D and E.

. . .

[End Clarification]

6.86 Interpretation of Value

Description:	The encoding of the information element "Value" is unclear in the ASN.1 and ABNF syntax specifications. This should be clarified.
Reference:	Subject: Re: [Fwd: RE: Interpretation of Value in ASN.1] Date: Wed, 09 May 2001 11:35:53 –0400 From: Troy Cauble <troy@lucent.com>

	Reply-To: Troy Cauble <troy@bell-labs.com> To: Christian Groves <Christian.Groves@ericsson.com>

[Begin Correction]

A.2 ASN.1 Syntax Specification

…

```
ServiceChangeParm ::= SEQUENCE
{
      serviceChangeMethod      ServiceChangeMethod,
      serviceChangeAddress     ServiceChangeAddress OPTIONAL,
      serviceChangeVersion     INTEGER(0..99) OPTIONAL,
      serviceChangeProfile     ServiceChangeProfile OPTIONAL,
      serviceChangeReason      Value,
            ; A serviceChangeReason consists of a numeric reason code
            ; and an optional text description.
            ; The serviceChangeReason SHALL be a string consisting of
            ; a decimal reason code, optionally followed by a single
            ; space character and a textual description string.
            ; This string is first ASN.1 BER encoded as an IA5String.
            ; The result of this ASN.1 BER encoding is then encoded as
            ; an ASN.1 BER OCTET STRING, "double wrapping" the value
            ; as was done for package elements.
      serviceChangeDelay       INTEGER(0..4294967295) OPTIONAL,
                                   -- 32 bit unsigned integer
      serviceChangeMgcId       MId OPTIONAL,
      timeStamp                TimeNotation OPTIONAL,
      nonStandardData          NonStandardData OPTIONAL,
      ...
}
```

…

[End Correction]

[Begin Correction]

B.2 ABNF Syntax Specification

…

```
serviceChangeMethod  = MethodToken EQUAL (FailoverToken /
                       ForcedToken / GracefulToken / RestartToken /
                       DisconnectedToken / HandOffToken /
                       extensionParameter)
; A serviceChangeReason consists of a numeric reason code
; and an optional text description.
; A serviceChangeReason MUST be encoded using the quotedString
; form of VALUE.
; The quotedString SHALL contain a decimal reason code,
; optionally followed by a single space character and a
; textual description string.
serviceChangeReason  = ReasonToken  EQUAL VALUE
serviceChangeDelay   = DelayToken   EQUAL UINT32
```

…

[End Correction]

6.87 Digit Map ABNF error

Description:	eventDM = DigitMapToken ((EQUAL digitMapName) / (LBRKT digitMapValue RBRKT))
	and a digit map descriptor is defined as:
	digitMapDescriptor = DigitMapToken EQUAL ((LBRKT digitMapValue RBRKT) / (digitMapName [LBRKT digitMapValue RBRKT]))
	
	As These structures are alike, the EQUAL in eventDM should be right after DigitMapToken as is the case in digitMapDescriptor.

Reference:	Subject: RE: ABNF Question: digit maps embedded in events Date: Wed, 30 May 2001 17:27:14 –0400 From: "Tom-PT Taylor" <taylor@nortelnetworks.com> To: "'Rosen, Brian'" <Brian.Rosen@marconi.com>, "'Steven Weisz'" <sweisz@mediatrix.com>, "Megaco Mailing List (E-mail)" <megaco@fore.com>

[Begin Correction]

B.2 ABNF Syntax Specification

…

eventDM = DigitMapToken EQUAL((~~EQUAL~~ digitMapName) /
(LBRKT digitMapValue RBRKT))

…

[End Correction]

7 Technical and Editorial Corrections to H.248 Annex F(2000)

7.1 Package ID of Text Telephone Package in Annex F shall be 0x0010

Description:	The numeric ID of the Text Telephone package in Section 7 of H.248 Annex F shall be changed to 0x0010 to match the IANA registration.

[Begin Correction]

F.7 Text Telephone package

PackageID: txp (0x0010~~6~~)

…

[End Correction]

7.2 Value of NAK

Description:	The numeric value of NAK shall be 0x000D, in the V8bistype parameter of the dtone event in the Call Type Discrimination package.

[Begin Correction]

F.8.2.1 Discriminating tone detected

EventID: dtone (0x0001)

...

ObservedEventDescriptor parameters:

....

DiscriminatingToneValue

ParameterID: dtvalue (0x0002)

....

V8bistype

ParameterID: v8bist (0x0004)

Type: enumeration

Possible values:

ESi	(0x0001)	V.8bis signal ESi
ESr	(0x0002)	V.8bis signal ESr
MRe	(0x0003)	V.8bis signal MRe
MRdi	(0x0004)	V.8bis signal MRd from initiator
MRdr	(0x0005)	V.8bis signal MRd from responder
CRe	(0x0006)	V.8bis signal CRe
CRdi	(0x0007)	V.8bis signal CRd from initiator
CRdr	(0x0008)	V.8bis signal CRd from responder
MS	(0x0009)	V.8bis message MS with contents in "dtvalue"
CL	(0x000A)	V.8bis message CL with contents in "dtvalue"
CLR	(0x000B)	V.8bis message CLR with contents in "dtvalue"
ACK	(0x000C)	V.8bis message ACK with contents in "dtvalue"
NAK	(0x000D~~E~~)	V.8bis message NAK with contents in "dtvalue"

…

[End Correction]

7.3 Correction in parameter values in Call Type Discrimination package in Annex F

Description:	Correction of conflicting parameter values for MRdrh, MRdrl and CReh in the V8bsn parameter of the V8bisSignal signal in the Call Type Discrimination package.

[Begin Correction]

F.8.3.4 V8bisSignal

SignalID: v8bs (0x0004)

Signaltype: BR

Parameters:

V8bisSigname

ParameterID: V8bsn (0x0001)

Type: Enumeration

Possible values:

ESi	(0x0001)	V.8bis signal ESi
ESr	(0x0002)	V.8bis signal ESr
MRe	(0x0003)	V.8bis signal MRe
MRdi	(0x0004)	V.8bis signal MRd from initiator
~~MRdrh~~	~~(0x0005)~~	~~V.8bis signal MRd from responder on high power~~
MRdrl	(0x0005)	V.8bis signal MRd from responder on low power
~~Creh~~	~~(0x0007)~~	~~V.8bis signal Cre on high power~~
CRel	(0x0006)	V.8bis signal CRe on low power
CRdi	(0x0007)	V.8bis signal CRd from initiator
CRdr	(0x0008)	V.8bis signal CRd from responder
MS	(0x0009)	V.8bis message MS with contents in signalvalue
CL	(0x000A)	V.8bis message CL with contents in signalvalue
CLR	(0x000B)	V.8bis message CLR with contents in signalvalue
ACK	(0x000C)	V.8bis message ACK with contents in signalvalue
NAK	(0x000D)	V.8bis message NAK with contents in signalvalue
MRdrh	(0x000E)	V.8bis signal MRd from responder on high power
CReh	(0x000F)	V.8bis signal CRe on high power

Default may be provisioned

…

[End Correction]

7.4 Correction in parameter values in Call Type Discrimination package in Annex F

Description:	Correction of conflicting parameter values for dtt parameter in dtone event. in the Call Type Discrimination package.

[Begin Correction]

F.8.2.1 Discriminating tone detected

EventID: dtone (0x0001)

Description:

This event indicates that a signal valid for detection and discrimination of mode was detected. The signal name is given as a parameter. Further logic is needed in some cases to discriminate the call type from this information. The V.8bis related parameters are returned only when V.8bis is supported [5].

Note that some textphones operate with DTMF tones. This package decodes initial DTMF signals according to the specification for text telephones in V.18 [6]. DTMF detection may be indicated also from the "dd" package if that is active.

EventsDescriptor parameters:

none

ObservedEventDescriptor parameters:

DiscriminatingToneType

ParameterID: dtt (0x0001)

Type: Enumeration

Possible values:

For FAX		
CNG	(0x0001)	a T.30 fax calling tone
V21flag	(0x0002)	V21 tone and flags for fax answering
For TEXT		
XCI	(0x0003)	a V.18 XCI
V18txp1	(0x0004)	a V.18 txp signal in channel V.21(1)
V18txp2	(0x0005)	a V.18 txp signal in channel V.21(2)
BellHi	(0x0006)	a Bell 103 carrier on the high channel
BellLo	(0x0007)	a Bell 103 low channel
Baudot45	(0x0008) characters	a Baudot45 initial carrier and
Baudot50	(0x0009)	a Baudot50 initial carrier and characters
Edt	(0x000A)	an EDT initial tone and characters
DTMF	(0x000B)	DTMF signals
For DATA		
Sig	(0x000C~~B~~)	Modulation signal from a mode only used for data, i.e. not V.21, V.23 nor Bell 103
Common to TEXT and DATA:		
CT	(0x000D~~C~~)	a V.25 calling tone
V21hi	(0x000E~~D~~)	a V.21 carrier on the higher frequency channel
V21lo	(0x000F~~E~~)	a V.21 carrier on the low frequency channel
V23hi	(0x0010~~0F~~)	a V.23 high carrier
V23lo	(0x0011~~0~~)	a V.23 low carrier
CI	(0x0012~~1~~)	a V.8 CI with contents in "dtvalue"
Common to FAX, TEXT and DATA:		
ANS	(0x0013~~2~~)	V.25 ANS, equivalent to T.30 CED from answering terminal
ANSbar	(0x0014~~3~~)	V.25 ANS with phase reversals
ANSAM	(0x0015~~4~~)	V.8 ANSam
ANSAMbar	(0x0016~~5~~)	V.8 ANSam with phase reversals
CM	(0x0017~~6~~)	V.8 CM with contents in "dtvalue"
CJ	(0x0018~~7~~)	V.8 CJ
JM	(0x0019~~8~~)	V.8 JM with contents in "dtvalue"
ENDOFSIG	(0x001A~~19~~)	End of reported signal detected reported for continuous or repeated signals
V8BIS	(0x001B~~20~~)	V.8bis signal, with signal type in parameter V8bistype and value in "dtvalue"
		…

[End Correction]

7.5 Missing Keywords in F.8.1.2

Description:	F.8.1.2 neglects to specify "Defined in:" or "Characteristics:"
Reference:	Subject: Re: Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com,MEGACO list <megaco@fore.com>

[Begin Correction]

F.8.1.2 Text Call Types

…

V18 (0x0008)

Description:

This parameter indicates for what text telephone modes the termination is monitored, used in TEXT mode.

Defined in: Termination State

Characteristics: Read / Write

…

[End Correction]

7.6 Duplicated propertyID in F.8.1

Description:	> F.8.1.3 and F.8.1.6 have the same PropertyID string (v8bsup). [CHG] Yes. The authors can specify an appropriate name.
Reference:	Subject: Re: Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com,MEGACO list <megaco@fore.com>

[Begin Correction]

F.8.1.6 PhasereversalDetect

PropertyID: phrevdet~~v8bsup~~ (0x0006)

Type: Boolean

…

[End Correction]

7.7 Duplicated propertyID in F.9.1

Description:	F.9.1.1 and F.9.1.2 have the same PropertyID number (0x01). F.9.1.2 updated.
Reference:	Subject: Re: Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com,MEGACO list <megaco@fore.com>

[Begin Correction]

F.9.1.2 Fax Transport

PropertyID: ftrpt (0x000~~1~~4)
Type: Enumeration

…

[End Correction]

7.8 Duplicated PropertyID in F.10.1

Description:	F.10.1.1 and F.10.1.2 have the same PropertyID number (0x01). F.10.1.2 to be updated.
Reference:	Subject: Re: Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com,MEGACO list <megaco@fore.com>

[Begin Correction]

F.10.1.2 IPFaxTransport

PropertyID: ipftrpt (0x0007~~1~~)
Type: Enumeration

…

[End Correction]

8 Technical and Editorial Corrections to H.248 Annex H

8.1 SCTP Streams

Description:	In chapter H.8 Stream Independence within Annex H it reads: "SCTP can provide up to 65536 unidirectional streams ... " this is correct there can be 65536 unique stream numbers (0-65535). Though the number of streams is limited to what is specified in the INIT / INITACK. There, according to the SCTP RFC 2960 variables: Number of Outbound Streams Number of Inbound Streams are represented by a 16 bit variables where the value of 0 (zero streams) is not allowed. Hence the actual number of streams which may ever be requested and accepted is 0xFFFF (65535). Therefore, the 65536 value in chapter H.8 in Annex H should be 65535.

Reference:	Editor

[Begin Correction]

H.8 Stream Independence

SCTP can provide up to 65535~~6~~ unidirectional streams in each direction of an MGC-MG association. SCTP transmits messages and processes received messages in one stream independent to the order or status of messages in any other streams. H.248 may avoid head-of-line blocking by transmitting unrelated transactions on different streams. Reliability is still provided. Ordering of messages is available per-stream.

[End Correction]

9 Technical and Editorial Corrections to H.248 Annex K

9.1 Superflous information

Description:	The Annoucement Package Annex K contains fields which are superflous and may lead to misinterpretation. The Notifycompletion indicator is a core H.248 element and does not need to be specified in a package. Also Signal type only needs one element ie. TO. Several signal do not need to be specified as these may be overridden by the core protocol.
Reference:	Editor

[Begin Correction]

K.3 Signals

…

`SignalID: apf (0x0001)`

`Description: Initiates the play of a fixed announcement.`

`SignalType:` ~~`OO, `~~`TO (default)`
`SignalDuration: Provisioned.`
~~`NotifyCompletion: Provisioned (default false)`~~

…

`SignalID: apv (0x0002)`

`Description: Initiates the play of a variable announcement.`

`SignalType:` ~~`OO, `~~`TO (default)`
`SignalDuration: Provisioned.`
~~NotifyCompletion: Provisioned (default false)~~

[End Correction]

10 Technical and Editorial Corrections to RFC-3015

This section contains technical and editorial correction to RFC-3015 only, that the faults described in this section do not affect the published H.248 (2000) recommendation.

10.1 Typographical Errors in the ASN.1 in RFC3015

Description:	When producing RFC3015 from recommendation H.248 (2000) two lines were omitted. It missed out the line defining IP4Address, which should be before IP6Address and there is a missing "..." at the end of the ServiceChangeParm definition.
Reference:	Subject: FW: Typos in RFC 3015 Date: Tue, 9 Jan 2001 14:27:55 -0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "Tom Taylor (E-mail)" <taylor@nortelnetworks.com>, "Christian Groves (E-mail)" <Christian.Groves@ericsson.com> CC: "'sob@harvard.edu'" <sob@harvard.edu>

[Begin Correction]

A.2 ASN.1 Syntax Specification

```
                                        …

  IP4Address ::= SEQUENCE
  {
      address        OCTET STRING (SIZE(4)),
      portNumber     INTEGER(0..65535) OPTIONAL
  }

  IP6Address ::= SEQUENCE
  {
      address        OCTET STRING (SIZE(16)),
                                        …

  ServiceChangeParm ::= SEQUENCE
  {
      serviceChangeMethod    ServiceChangeMethod,
      serviceChangeAddress   ServiceChangeAddress OPTIONAL,
      serviceChangeVersion   INTEGER(0..99) OPTIONAL,
      serviceChangeProfile   ServiceChangeProfile OPTIONAL,
      serviceChangeReason    Value,
      serviceChangeDelay     INTEGER(0..4294967295) OPTIONAL,
                             -- 32 bit unsigned integer
      serviceChangeMgcId     MId OPTIONAL,
      timeStamp              TimeNotation OPTIONAL,
      nonStandardData        NonStandardData OPTIONAL,
      ...
      }
```

[End Correction]

11 Implementation Clarifications for H.248

11.1 Returning a Context ID List

Description:	In section 7.2.5 AuditValue it has the following table: The following illustrates other information that can be obtained with the Audit Command: **ContextID TerminationID Information Obtained** : All wildcard Audit of all matching Terminations and The Context to which they are associated. All Root List of all ContextIds

[Begin Clarification]

The list of Context Ids should be returned by using multiple Action Replys, each containing a context from the list.

[End Clarification]

11.2 Service Change Address and Ports

Description:	There is some confusion on when to use either ServiceChange Address or ServiceChange MGCID. The text below offers some advice on

[Begin Clarification]

1) The use of ServiceChangeAddress is not encouraged
2) MGCs must be able to cope with ServiceChangeAddress with either a full address or just a port number in the case of TCP transport.

[End Clarification]

11.3 Audit Response with and without wildcard response (w-)

Description:	There is some confusion on what should be sent in the response to an Audit in the cases where you have wilcarded the context or termination and have specified wildcarded response.

[Begin Clarification]

Assume the gateway has 4 terminations : t1/1, t1/2, t2/1 and t2/2
Assume terminations t1/* has implemented packages aaa and bbb and terminations t2/* has implemented packages ccc and ddd.
Assume context 1 has t1/1 and t2/1 in it, and context 2 has t1/2 and t2/2 in it.
The command:

```
        Context=1{Audit=t1/1{Audit{Packages}}
```

Returns:

```
        Context=1{Audit=t1/1{Packages{aaa,bbb}}}}
```

The command:

```
        Context=*{Audit=t2/*{Audit{Packages}}
```

Returns:

```
        Context=1{Audit=t2/1{Packages{ccc,ddd}}},
        Context=2{Audit=t2/2{Packages{ccc,ddd}}}
```

The command:

```
        Context=*{W-Audit=t1/*{Audit{Packages}}
```

Returns:

```
        Context=*{W-Audit=t1/*{Packages{aaa,bbb}}
```

Wildcard response may also be used for other commands such as Subtract.

[End Clarification]

11.4 Package Extension and Referencing

Description:	The current text in H.248 Section 12.1 is ambiguous about the usage of package names when referencing properties, events, signals and statistics in a base package and an extended package.

[Begin Clarification]

12.1 Guidelines for defining packages
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. 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 optional to publish the base package as an allowed interface. If it does publish A, then A would be reported on the Package 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 the purpose of improved interoperability and backward compatibility, the 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 designed to be extended by others, not directly controlled, and may not have any function on their own or be nonsensical on their own, in which case the MG SHOULD NOT publish the base Packages.

[End Clarification]

11.5 Zero in octetString

Description:	It is unclear why 0 is not included in the nonEscapeChar.
Reference:	Subject: Zero not allowed Date: Sun, 31 Dec 2000 11:30:07 -0600 Reply-To: plong@ipdialog.com From: Paul Long <plong@packetizer.com> Organization: ipDIalog, Inc. To: MEGACO@STANDARDS.NORTELNETWORKS.COM

[Begin Clarification]

The octet zero is not among the permitted characters in octet string, which is used in the ABNF for local and remote descriptors. As the current definition of these descriptors is limited to SDP, and a zero octet would not be a legal character in SDP, this is not a concern.

[End Clarification]

11.6 First Line of SDP a newline

Description:	There is confusion in the ABNF encoding with regards to the use of a new line as the first line of the SDP. The clarification below seeks to clarify this.
Reference:	Subject: RE: [Fwd: RE: Does the First Line of SDP need a newline] Date: Tue, 13 Mar 2001 09:13:57 -0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "'Christian Groves'" <Christian.Groves@ericsson.com>, megaco ietf <megaco@fore.com>

[Begin Clarification]

Referring to section 7.1.8 and the ABNF for Local/Remote, SDP disallows whitespace at the beginning of a line, Megaco ABNF allows whitespace before the beginning of the SDP in the Local/Remote descriptor. Parsers should accept whitespace between the LBRKT following the Local/Remote token and the beginning of the SDP.

[End Clarification]

12 Implementation Clarifications for H.248 annex H

12.1 MTP3 Interworking

Description:	When studying some network scenarios for a certain networks, there is a need to evolve the signalling transport from SS7 MTP3B in an ATM environment to the use of SCTP in IP environments. To provide this M3UA on top of SCTP can be used. It is also seen that M3UA supports flexible implementation scenarios. Therefore some addition indicating the use of M3UA on top of SCTP needs to specified in Annex H in H.248.
Reference:	Subject: MTP 3 interworking Date: Thu, 3 May 2001 16:51:14 +0200 From: "Alf Heidermark (UAB)" <Alf.Heidermark@uab.ericsson.se> To: "'Megaco (E-mail)" <megaco@fore.com>

[Begin Clarification]

To provide interworking between MTP3B and SCTP and to allow for flexible implementations of gateways and controllers in order to offer efficient use of SCTP associations the M3UA layer may be added on top of SCTP.

[End Clarification

13 H.248 Recommendation Series Defect Report Form

DATE:	
CONTACT INFORMATION **NAME:** **COMPANY:** **ADDRESS:** **TEL:** **FAX:** **EMAIL:**	
AFFECTED RECOMMENDATIONS:	

DESCRIPTION OF PROBLEM:	
SUGGESTIONS FOR RESOLUTION:	

NOTE - ATTACH ADDITIONAL PAGES IF MORE SPACE IS REQUIRED THAN IS PROVIDED ABOVE.