INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 1997 - 2000

COM 16-R 3-E
May 1997
Original: English

Questions: 1-23/16

STUDY GROUP 16 – REPORT R 3

SOURCE*: STUDY GROUP 16 (GENEVA MEETING, 17-27 MARCH 1997)

TITLE: IMPLEMENTOR'S GUIDE FOR RECOMMENDATIONS UNDER THE RESPONSIBILITY OF STUDY GROUP 16

CONTENTS

* **Contact:** TSB

Tel: +41 22 730 5860
Fax: +41 22 730 5853
email: bigi@itu.int

1 Introduction

Study Group 16 considered the following Implementor's Guides of ITU-T Recommendations of its responsibility.

The Implementor's Guides contain a compilation of all defects found up to March 1997. These defects may be of an editorial nature of may be technical errors such as omissions, inconsistencies or ambiguities. The Guides also include changes to the Recommendation which have been approved at Study Group level.

It is the intention to update the Guides periodically and to provide updates (if necessary) after each meeting of Study Group 16. Implementors are encouraged to provide input to the next issue.

2 Implementor's Guide for V.8 and V.8bis

The Implementor's Guide for ITU-T Recommendations V.8 and V.8bis will be provided by the Rapporteur for Q.4/16 (Mr. L. Brown, Motorola, United States) as a white contribution for the next meeting of SG 16, based on the text published in TD 92.

3 Implementor's Guide for V.25ter

The Implementor's Guide for ITU-T Recommendation V.25ter is as follows[1]:

3.1 Introduction

This document is a compilation of additional commands and changes to commands contained in ITU-T Recommendation V.25ter. It is intended to be read in conjunction with the Recommendation to serve as an additional authoritative source of information for implementors. The additions and changes defined herein are expected to be included in future versions of the Recommendation.

The first approved version of the Guide was produced at the March 1997 ITU-T Study Group 16 meeting. Wide distribution of this document is expected and encouraged.

3.2 Additions and Changes

3.2.1 Seamless Rate Change Command

Add the following command to § 6.4 as § 6.4.8:

[1] Contact: Fred Lucas
Rapporteur for Q.7/16
General Data Comm, Inc.
United States

Tel:	+1 203 574 1118
Fax:	+1 203 573 0069
email:	fred.lucas@gdc.com

6.4.8 Seamless Rate Change Enable (+MSC)

Parameter

+MSC=<src_v34>

Description

This extended format compound numeric parameter controls whether or not seamless rate change procedures are enabled during V.34 operation.

{NOTE - The addition of other subparameters to control other aspects of Seamless Rate Change Operation, and control of V.pcm SRC Operation, is for further study. The effect that enabling seamless rate change may have on other modem characteristics, such as startup time, is for further study.}

The results of seamless rate change negotiation are reported with the **+MSCR** indication, which is enabled with the same +MR command that enables other modulation reports such as +MRR. The form of the indication is as follows:

+MSCR: <v34_src_mode> e.g. +MSCR: 1

<v34_scr_mode> has a value of zero for no V.34 seamless rate change (SRC), and a value of one for V.34 SRC operation

{NOTE - Additional reported values are for further study.}

Defined values

Seamless Rate Change Control Values

<src_v34 >	Description
0	Disables V.34 seamless rate change
1	Enables V.34 seamless rate change

Recommended default setting

1

Read syntax

+MSC?
The DCE shall transmit a line of information text to the DTE, consisting of:
+MSC: <current setting>
For example, with the recommended default setting, the DCE could report:
+MSC: 1

Test syntax

+MSC=?
The DCE shall transmit a string of information text to the DTE, consisting of:
+MSC: (list of supported values)
For example, a DCE that supported all defined settings would report:
+MSC: (0,1)

3.2.2 Addition to +MV18S in § 6.4.4

Add the new parameter <probing_en> to the +MV18S command.

– **<probing_en>**, which controls activation of the probing in answer mode. Disabling the probing will cause the V.18 DCE to enter the automoding monitor mode when answering;

Add the new parameter <probing_en> to Table 15/V.25ter:

<probing_en>	Description
0	Disable the probing
1	Enable the probing (Default)

3.2.3 Change the +MV18 reporting control in § 6.4.5

Replace Table 16/V.25ter with the following new table:

TABLE 16/V.25*TER*

V.18 Connection report intermediate result codes

+MV18:5BIT	Indicates connection with 5-bit
+MV18:EDT	Indicates connection with EDT
+MV18:DTMF	Indicates connection with DTMF
+MV18:V21	Indicates connection with Recommendation V.21
+MV18:V23M	Indicates connection with Recommendation V.23 in Master Mode (Sending on 1 200 bit/s, receiving on 75 bit/s)
+MV18:V23M	Indicates connection with Recommendation V.23 in Slave Mode, (Sending on 75 bit/s, receiving on 1 200 bit/s)
+MV18:B103	Indicates connection with Bell 103-type modulation
+MV18:V18	Indicates both DCEs are in Recommendation V.18

4 Implementor's Guide for V.61

This Implementor's Guide applies to ITU-T Recommendation V.61, version 01, date of issue 1996.

This Implementor's Guide contains a compilation of all defects found up to *Date*.

These defects may be of an editorial nature or be technical errors such as omissions, inconsistencies or ambiguities.

It is the intention to update this Guide periodically and to provide updates (if necessary) after each meeting of Study Group 16. Implementors are encouraged to provide input to the next issue.

Contact Information

ITU-T Study Group 16/Question 4 Rapporteur	Richard Stuart U.S. Robotics 10440 Little Patuxent Parkway Suite 900 Columbia, MD 21044 United States	Tel: Fax: email:	+1.410.884.4017 +1.410.884.4018 dstuart@usr.com
Implementor's Guide Editor	Akira ATSUTA Matsushita Electric Industrial Co., Ltd. 2-3-8 Shimomeguro, Meguro-ku, TOKYO 153 JAPAN	Tel: Fax: email:	+81.3.5434.7077 +81.3.5434.7159 akira.atsuta@rdmg.mgcs.mei.co.jp
ITU-T Recommendation V.61 Editor	Kenneth D. Ko Rockwell Semiconductor Systems Largo, FL United States	Tel: Fax: email:	+1.813.538.0729 +1.813.538.0610 kenneth.ko@nb.rockwell.com

Document History

Version	Date	Description
1	20 March 1997	Initial version - completed at the ITU-T Study Group 16 meeting, Geneva, 17-27 March 1997

4.1 Introduction

This document is a compilation of reported defects identified with the 1996 edition of the ITU-T V.61 Recommendation. It is intended to 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 the V.61 Recommendation.

The first version of the Guide was produced at the March 1997 ITU-T Study Group 16 meeting. Wide distribution of this document is expected and encouraged.

4.2 Scope

This Guide resolves defects in the following categories:

- editorial errors;
- technical errors such as omissions or inconsistencies;
- ambiguities.

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

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

4.3 Policies for Updating this Document

This document is managed by the ITU-T Study Group 16 Question 4 Rapporteur's Group. It can be revised at any recognized Q.4/16 Rapporteur's Group meeting provided the proposed revisions are accepted by the members of the group and approved at the next SG 16 meeting. A revision history cataloguing the evolution of this document is included.

4.4 Defect Resolution Procedure

Upon discovering technical defects with any components of the V.61 Recommendation, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q.4/16 Rapporteur. Contact information for these parties is included in this 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 V.61 Recommendation. Formal membership in the ITU is not required to participate in this process.

4.5 References

This document refers to the following V-series Recommendations:

– ITU-T Recommendation V.61 (1996), *A simultaneous voice plus data modem, operating at a voice+data signalling rate of 4 800 bit/s, with optional automatic switching to data-only signalling rates of up to 14 400 bit/s, for Use on the General Switched Telephone Network and on Leased Point-to-Point 2-wire Telephone-Type Circuits.*

4.6 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.

4.7 Technical and Editorial Corrections

4.7.1 Technical and Editorial Corrections to ITU-T Recommendation V.61

4.7.1.1 Pre-emphasis filter

Description

An error has been discovered in the pre-emphasis filter used in the audio functions.

V.61 allows the implementors to choose one of the two rates of sampling of audio source signal, i.e. 3 or 4 samples/symbol, and defines two sets of pre-emphasis filter coefficients in Table 2/V.61 and Table 3/V.61 respectively, with the intent of interoperability between modems with different sampling rates; 3 samples/symbol transmitter and 4 samples/symbol receiver or vice versa can communicate.

This intent was unrealizable due to the omitted parentheses, thus the correction below.

[This error was found by Akira Atsuta of Matsushita Electric Industrial Co., Ltd. and reported in White Contribution 17 of SG 16 meeting in Geneva, 17-27 March 1997.]

[Begin Correction]

5.5.2 AUDIO FUNCTIONS

5.5.2.1 Pre-emphasis filter

A pre-emphasis filter is applied to the audio source signal. The filter operates at the **MS** rate and is of the form

$$y(k) = c_0 x(k) + c_1 x(k-1) + c_2 x(k-2)$$

$$y(k) = c_0 \{ \ x(k) + c_1 x(k-1) + c_2 x(k-2) \ \}$$

where $x(k)$ is the signal input to the pre-emphasis filter and $y(k)$ is the output of the pre-emphasis filter.

The coefficients c_0, c_1 and c_2 of the pre-emphasis filter shall be selected from the coefficient sets in Table 2 (for 3 samples/symbol) or Table 3 (for 4 samples/symbol). The coefficients may be adaptively selected or fixed. If adaptive, they shall be updated once per frame.

The index indicating the filter coefficient set in use in the pre-emphasis filter shall be transmitted in the control information once per frame to the receiver. The index shall be transmitted regardless of whether the pre-emphasis coefficients are fixed or adaptive. The index for each set of filter coefficients is shown in Table 2 (for 3 samples/symbol) or Table 3 (for 4 samples/symbol).

NOTE – If the coefficients are fixed, they should be set to values which provide compromise compensation for the expected spectral content in the input signal.

[End Correction]

4.8 Implementation Clarifications

— None —

5 Implementor's Guide for ITU-T Recommendations V.70, V.75 and V.76

5.1 Introduction

This document is a compilation of reported defects identified with the 1996 editions of the ITU-T Recommendations V.70, V.75 and V.76. It is intended to be read in conjunction with the Recommendations 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 Recommendations.

The first approved version of the Guide was produced at the March 1997 ITU-T Study Group 16 meeting. Wide distribution of this document is expected and encouraged.

5.2 Scope

This Guide resolves defects in the following categories:

- editorial errors;
- technical errors such as omissions or inconsistencies;
- ambiguities.

In addition the 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 the normal way through contributions to the ITU-T.

5.3 Policies for Updating this Document

This document is managed by the ITU-T Study Group 16 Question 11 Rapporteur's Group. It can be revised at any recognized Q.10/8 Rapporteur's Group meeting provided the proposed revisions are accepted by the members of the group. A revision history cataloguing the evolution of this document is included.

5.4 Defect Resolution Procedure

Upon discovering technical defects with any components of the V.70-series Recommendations, please provide a written description directly to the editors of the affected Recommendations with a copy to the Rapporteur. Contact information for these parties is included in this 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 V.70-series Recommendations. Formal membership in the ITU is not required to participate in this process.

5.5 References

This document refers to the Recommendations given in the following sections:

5.6 Amendments to Recommendation V.70

5.6.1 Completion of capability exchange

V.70 capabilities are exchanged in H.245 **TerminalCapabilitySet** messages using two $\text{XID}_{\text{C/R}}$ frame exchanges. To ensure that both the remote and the local terminal exchange capsets, a clarification should be added to the text of V.70 requiring the receiver of a **TerminalCapabilitySet** message to initiate the exchange in the reverse direction if it has not already done so.

Additional text

§ 6.4.4.1, at the end of paragraph 2 beginning "The SCF may optionally exchange ….", add:

"For in-band capability exchanges, if the SCF receives a CE-SETPARM indication from the CE, it should issue a CE-SETPARM request on the same DLC, if this has not been issued already."

5.6.2 Clarification on usage of H.245 "maxBitRate"

The following text is proposed for inclusion at the end of § 6.2.1 of Recommendation V.70:

"For V.70 terminals the "maxBitRate" parameter in H.245 Data Application Capability shall be set to 0 for all connections and shall have no meaning. The use of other values is for further study."

5.6.3 Correction of terminology on simultaneous capability sets

Amend the tenth paragraph to read:

"The terminal shall only exchange multiple **simultaneousCapabilities** structures using the optional out-of-band channel."

5.6.4 Addition of G.729B option

Add G.729 Annex B to § 2. References.

In § 5.4 replace the text:

"The definition of a Voice Activity Detector and Comfort Noise Generator for use with the G.729 Annex A coder is for further study."

with the text:

"A V.70 terminal may optionally include the silence suppression algorithms (i.e. Voice Activity Detection and Comfort Noise Generation) defined in Annex B to Recommendation G.729, for use with the coder defined in either G.729 or G.729 Annex A. A V.70 terminal signalling a G.729 Annex B capability shall be assumed to be capable of supporting the operation of the appropriate speech coder without the silence suppression."

5.6.5 Clarification of preferred use of AlternativeCapabilitySet structure

The following text should be added at the end of the eighth paragraph of Section 6.2.1 of Recommendation V.70, to indicate the preference for only one capability in an AlternativeCapabilitySet:

"A DSVD terminal may indicate multiple capabilities in an **AlternativeCapabilitySet** structure, but the preferred number is one. A DSVD terminal shall be able to receive an **AlternativeCapabilitySet** indicating multiple capabilities, but may respond with a single capability in the **AlternativeCapabilitySet** in the reverse direction."

5.6.6 Negotiation of the Suspend/Resume Option

The following changes to Recommendation V.70 have been approved by SG 16, and enable negotiation of the suspend/resume option using the procedures of Recommendation V.8bis.

Amend paragraph 4 of Section 6.2.1 to read:

"~~An~~The out-of-band control channel ~~shall~~or V.8bis may be used for the negotiation of the optional suspend/resume mode of operation."

Amend paragraph 8 of Section 6.2.2 to read:

"If the suspend/resume mode is selected using the out-of-band channel, the MF shall redefine the abort sequence at the time the mode is selected. Suspend/resume DLCs may then be opened. If the suspend/resume mode is selected using V.8bis, the MF shall redefine the abort sequence at the time the data mode is established. If omission of the Address Field is negotiated, only one suspend/resume channel shall be opened. If the Address Field is maintained, one or more suspend/resume channels may be opened."

5.7 Amendments to Recommendation V.75

5.7.1 N401 Value Range

The V76Capability and V76HDLCParameters both contain an indication of the maximum number of octets that can be transferred in an information field (n401). The n401 INTEGER in the V76HDLCParameters has been incorrectly set to an 7-bit field.

Correction: Change in the Annex of V.75 the structure:

```
V76HDLCParameters                        ::=SEQUENCE
{
        crcLength                        CRCLength,
        n401                             ~~INTEGER (1..127)~~INTEGER(1..4095),
        loopbackTestProcedure            BOOLEAN,
                ...
}
```

5.7.2 Missing H.245 message specifications in Table 6/V.75

Add "- TerminalCapabilitySet.sequenceNumber" as first entry in 2nd column of Table 6/V.75.

Add "- RequestMode.ModeDescription.ModeElement.V76ModeParameters" below " - Endsession" entry in 2nd column of Table 6/V.75.

5.7.3 Corrections to Table 3/V.75

Presence of the OPTIONAL "portNumber" in V.70/H.245 messages is required to be present in the H.245 "OpenLogicalChannelAck".

TABLE 3

L-ESTABLISH user-data parameters

L-ESTABLISH			
MF primitives	**Applicable H.245 parameters**	**CE reference**	**Comments**
– request	– OpenLogicalChannel.forwardLogicalChannel Number.LogicalChannelNumber	1a.1	
– indication	– OpenLogicalChannel.forwardLogicalChannel Parameters.portNumber	1a.9	– required to be present for DSVD and shall identify a logical channel user.
	– OpenLogicalChannel.forwardLogicalChannel Parameters.multiplexParameters.v76Logical ChannelParameters	1a.4 - 1a.8	– shall be present for DSVD
	– OpenLogicalChannel.reverseLogicalChannel Parameters.multiplexParameters.v76Logical ChannelParameters	1a.4 - 1a.8	– shall be present for DSVD
	– OpenLogicalChannel.forwardLogicalChannel Parameters.v76Parameters.v75Parameters	1a.3	– shall be present for DSVD
	– OpenLogicalChannel.forwardLogicalChannel Parameters.dataType.AudioData	1a.10	
	– OpenLogicalChannel.forwardLogicalChannel Parameters.dataType.dataDataApplication Capability	1a.10	
	– OpenLogicalChannel.reverseLogicalChannel Parameters	1a.11	– shall be present for DSVD
	– OpenLogicalChannel.reverseLogicalChannel Parameters.v76Parameters.v75Parameters	1a.3	– shall be present for DSVD
	– OpenLogicalChannel.reverseLogicalChannel Parameters.dataType	1a.11	– identical to forward datatype parameters with the exception of the AudioCapability INTEGER
– response – confirm	– OpenLogicalChannelAck.forwardLogical ChannelNumber – OpenLogicalChannelAck.portNumber	1a.12 1a.12	 – shall be present for DSVD

5.7.4 Negotiation of suspend/resume option

The suspendResume choice as defined in the V76LogicalChannelParameters, cannot be selected as an option, but it is an optional function of the V.76 multiplexer.

It is agreed to amend the syntax to CHOICE and add a third choice of noSuspendResume to provide the capability to open logical channels without SuspendResume.

New Syntax

```
V76LogicalChannelParameters             ::=SEQUENCE
{
   hdlcParameters                       V76HDLCParameters,
   suspendResume                        CHOICE
   {
        noSuspendResume                 NULL,
        suspendResumewAddress           NULL,
        suspendResumewoAddress          NULL,
            …
   },
   uIH                                  BOOLEAN,
   mode                                 CHOICE
   {
            eRM                         SEQUENCE
            {
                  windowSize            INTEGER (1..127),
                  recovery              CHOICE
                  {
                        rej             NULL,
                        sREJ            NULL,
                        mSREJ           NULL,
                        …
                  },
                  …
            },
            uNERM                       NULL,
            …
   },
   v75Parameters                        V75Parameters,
   …
}
```

5.7.5 Clarification on usage of H.245 "portNumber"

The following text is proposed for inclusion in § 8.1 of Recommendation V.75:

"NOTE - The portNumber parameter shall have a default value of "0", which indicates an "unspecified" input or output association. Use of other values of this parameter is for further study."

5.7.6 Addition of codepoint for G.729 Annex B

Options are added for G.729 and G.729 Annex A with the silence suppression algorithm defined in G.729 Annex B.

Add G.729 Annex B to § 2. References

New Syntax

```
AudioCapability                                 ::=CHOICE
{
      nonStandard                               nonStandardParameter,
      g711Alaw64k                               INTEGER(1..256),
      g711Alaw56k                               INTEGER(1..256),
      g711Ulaw64k                               INTEGER(1..256),
      g711Ulaw56k                               INTEGER(1..256),
      g722-64k                                  INTEGER(1..256),
      g722-56k                                INTEGER(1..256),
      g722-48k                                INTEGER(1..256),

      g7231                                   SEQUENCE
      {
              maxAl-sduAudioFrames            INTEGER(1..256),
              silenceSuppression              BOOLEAN
      },

      g728                                    INTEGER(1..256),
      g729                                    INTEGER(1..256),
      g729AnnexA                              INTEGER(1..256),
      is11172AudioCapability                  is11172AudioCapability,
      is13818AudioCapability                  is13818AudioCapability,
      …,
      g729wAnnexB                             INTEGER(1..256),
      g729AnnexAwAnnexB                       INTEGER(1..256)

}
```

5.7.7 Correction positioning of segmentation/reassembly header

Amend the first paragraph of § 11.1 to read:

"An H-header octet ~~is~~shall be added ~~before the address field~~as the first octet in the information field of ~~the~~every frame to be transmitted on the DLC. The format of the H octet is shown in Figure 2."

5.7.8 Addition of H.245 codepoints for V.42bis on an ERM channel

The current version of H.245 does not have provisions to signal and negotiate V.42bis data compression parameters. It is agreed to add the v76wCompression parameter as a choice in DataProtocolCapability. This same structure will be called by both OpenLogicalChannel and TerminalCapabilitySet. The use of a CHOICE structure permits the definition of other compression schemes after the extension marker in the future.

It is proposed to allow only the symmetric P1/P2 parameter values for the two directions, consistent with the existing V.42bis.

Proposed syntax

```
DataProtocolCapability  :=  CHOICE
{
        nonStandard                             NonStandardParameter,
        v14buffered                             NULL,
        v42lapm                                 NULL,
        hdlcFrameTunnelling                     NULL,
        h310SeparateVCStack                     NULL,
        h310SingleVCStack                       NULL,
        transparent                             NULL,
        …,
        segmentationAndReassembly               NULL,
        hdlcFrameTunnellingwSAR                 NULL,
        v120                                    NULL,  -- as in H.230
        separateLANStack                        NULL,
        v76wCompression                         CHOICE
        {
                transmitCompression             CompressionType,  -- P0=1
                receiveCompression              CompressionType,  -- P0=2
                transmitAndReceiveCompression        CompressionType,  -- P0=3
                …
        }
}

CompressionType ::=CHOICE                             -- Newly added structure
{
        v42bis              V42bis,                   -- Future methods of compression will
        …                                             -- have to be added after the ext. marker
}

V42bis              ::=SEQUENCE                       -- Newly added structure
{
        numberOfCodewords             INTEGER (1..65536), -- P1
        maximumStringLength           INTEGER (1..256),     -- P2
        …
}
```

5.7.9 Reference to ISO 3309

Add reference to ISO 3309 in § 8.3.1

5.8 Amendments to Recommendation V.76

5.8.1 Clarification of CRC calculation when using suspend/resume

Amend § 5.1.6.1 item b) to read:

b) the remainder of the division (modulo 2) by the generator polynomial $x^8 + x^2 + x + 1$, of the product of x^8 by the content of the frame existing between, but not including, the final bit of the opening flag and the first bit of the FCS, excluding bits inserted for transparency and real time frames as defined in Annex A (including the bits of the suspend and resume flags).

Amend § 5.1.6.2 item b) to read:

b) the remainder of the division (modulo 2) by the generator polynomial $x^{16} + x^{12} + x^5 + 1$, of the product of x^{16} by the content of the frame existing between, but not including, the final bit of the opening flag and the first bit of the FCS, excluding bits inserted for transparency and real time frames as defined in Annex A (including the bits of the suspend and resume flags).

Amend § 5.1.6.3 item b) to read:

b) the remainder of the division (modulo 2) by the generator polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$, of the product of x^{32} by the content of the frame existing between, but not including, the final bit of the opening flag and the first bit of the FCS, excluding bits inserted for transparency and real time frames as defined in Annex A (including the bits of the suspend and resume flags).

5.8.2 Clarification of N401 value

The following text should be added at the end of Section 9.3 of Recommendation V.76 to clarify that the value of N401 is in octets:

"When the value of N401 shall be in octets when signalled using H.245."

5.8.3 Clarification on the reuse of DLCI values

Section 6.1.1 of V.76 has been revised after approval of the Recommendation to clarify the ambiguity regarding the issue of reusing freed DLCI values. Because this change was made after approval of the Recommendation, and in order to ensure reliable interworking and maximum interoperability, it is recommended that the receiver of a DLC establishment attempt should continue with this procedure even if the proposed DLCI is not the DLCI that was expected.

The following text should replace Section 6.1.1 of Recommendation V.76.

"6.1.1 Data Link Connection Identifier (DLCI)

The DLCI is used to identify an individual user information stream as well as to identify SU-to-SU connections. Multiple DLCIs shall be supported but the number is application-specific.

Selection of a DLCI value shall be as follows:

The Initiator shall select DLCI values for new DLCs with increasing values starting from 0.DLCI values, originally chosen by the Initiator, that have been freed by either end should be reused in ascending order by the Initiator rather than leaving unnecessary gaps in the numbering range.

The Responder shall select DLCI values for new DLCs with decreasing values starting from 63 when using one-octet address fields or 8191 when using two-octet address fields. DLCI values, originally chosen by the Responder, that have been freed by either end should be reused in descending order by the Responder rather than leaving unnecessary gaps in the numbering range.

The role of Initiator and Responder shall be made known to the MF by the SU. The means of doing so is beyond the scope of this Recommendation.

Use of the second address field octet is optional. All DLC entities shall be able to receive frames with a two-octet address field. If a frame is received with an address field of a different type from the one negotiated, the receiving DLC entity shall ignore the frame.

In case of collision (i.e. the same DLCI value being selected), the Responder shall back off its attempt to establish a new DLC (i.e. it shall inform its SU of failure to establish the DLC it attempted and continue with the DLC establishment attempt by the Initiator). Note that the procedures described above are intended to reduce the probability of collisions occurring.

The DLCI used on a given DLC is mapped to/from an internal "connection endpoint identifier" for communication between the MF and the SU".

5.9 Contact Information

ITU-T Study Group 16 Question 4/16 Associate Rapporteur	Mr. Chris Hansen Intel Corporation	Tel: Fax: email:	+1 503 264 8877 +1 503 264 3375 chris_hansen@ccm.jf.intel.com
Implementor's Guide Editor	Mr. J. Magill Lucent Technologies United Kingdom	Tel: Fax: email:	+44 1666 510105 +44 1666 510106 hjm@probecom.demon.co.uk
ITU-T Recommendation V.70 Editor	Mr. A. Sadri IBM	Tel: Fax: email:	+1 919 543 3418 +1 919 254 6963 asadri@vnet.ibm.com

ITU-T Recommendation V.75 Editor	Mr. A. Sadri IBM	Tel: Fax: email	+1 919 543 3418 +1 919 254 6968 asadri@vnet.ibm.com
ITU-T Recommendation V.76 Editor	tba	Tel: Fax: email:	+ +

5.10 History

Draft Issue 3	First issue approved by the Rapporteur Group - 27 September 1996
Draft Issue 4	Sections 2.5, 4.2 and 5 added. Approved by the Rapporteur Group - 29 January 1997

6 Implementor's Guide for the ITU-T T.120 Recommendation series - Data Protocols for Multimedia Conferencing

Abstract

This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T T.120-series Recommendations. It is intended to be read in conjunction with the Recommendations 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 T.120-series Recommendations.

Contact Information

ITU-T Study Group 16 Question 3/16 Rapporteur	Bruce DeGrasse BJ Communications 3311 Brookhaven Club Drive Dallas, TX 75234 United States	Tel: Fax: email:	+1.214.241.3139 +1.214.241.3139 73760.2250@compuserve.com
Implementor's Guide Editor	Pasquale Romano Polycom Inc. 2584 Junction Ave San Jose, CA 95134 United States	Tel: Fax: email:	+1.408.474.2864 +1.408.474.2945 promano@polycom.com
ITU-T Recommendation T.120 Editor	John Boucher BT Labs Martlesham Heath, Ipswich, Suffolk, UK IP5 7RE	Tel: Fax: email:	+44.1473.647697 +44.1473.643791 peers_t_w@bt-web.bt.co.uk
ITU-T Recommendation T.121 Editor	Trevor Peers BT Labs Martlesham Heath, Ipswich, Suffolk, UK IP5 7RE	Tel: Fax: email:	+44.1473.644354 +44.1473.643791 t_peers@bt-web.bt.co.uk

ITU-T Recommendation T.122 Editor	Jim Johnstone DataBeam 3191 Nicholasville Rd. Lexington, KY 40503 United States	Tel: +1.606.245.3649 Fax: +1.606.245.3525 email: jjj@databeam.com
ITU-T Recommendation T.123 Editor	Terry Lyons Lucent Technologies 200 Laurel Avenue Middletown, NJ 07748 United States	Tel: +1.908.957.5644 Fax: +1.908.957.5403 email: terry.g.lyons@lucent.com
ITU-T Recommendation T.124 Editor	Jeff Bernstein PictureTel Corporation 100 Minuteman Road Andover, MA 01810 United States	Tel: +1.508.623.4349 Fax: +1.508.749.2806 email: jeff@pictel.com
ITU-T Recommendation T.125 Editor	Terry Lyons Lucent Technologies 200 Laurel Avenue Middletown, NJ 07748 United States	Tel: +1.908.957.5644 Fax: +1.908.957.5403 email: terry.g.lyons@lucent.com
ITU-T Recommendation T.126 Editor	Pasquale Romano Polycom Inc. 2584 Junction Ave San Jose, CA 95134 United States	Tel: +1.408.474.2864 Fax: +1.408.474.2945 email: promano@polycom.com
ITU-T Recommendation T.127 Editor	Trevor Peers BT Labs Martlesham Heath, Ipswich, Suffolk, UK IP5 7RE	Tel: +44.1473.644354 Fax: +44.1473.643791 email: t_peers@bt-web.bt.co.uk

Document History

Version	Date	Description
1	15 February 1996	Initial version - completed at the ITU-T Study Group 8 meeting, Geneva 6-15 February 1996
2	26 April 1996	Updated at the Q.10/8 Rapporteur's meeting, Boston 21-26 April 1996 Sections added: 7.5.4 and 7.5.5
3	9 August 1996	Updated at the Q.10/8 Rapporteur's meeting, Santa Rosa 5-9 August 1996 Sections added: 7.5.6
4	4 October 1996	Updated at the Q.10/8 Rapporteur's meeting, Ismaning 30 September - 4 October 1996 Sections added: 7.4.1, 7.5.7, 7.5.8, 7.5.9 and 7.5.10

5	17 January 1997	Updated at the Q.10/8 Rapporteur's meeting, Newport Beach 13-17 January 1997 Augmented section 7.5.9 with additional clarifications

6.1 Introduction

This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T T.120-series Recommendations. It is intended to be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementor's. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected T.120-series Recommendations.

The first version of the Guide was produced following the February 1996 ITU-T Study Group 8 meeting. Wide distribution of this document is expected and encouraged.

6.2 Scope

This Guide resolves defects in the following categories:

- editorial errors;
- technical errors such as omissions or inconsistencies;
- ambiguities.

In addition the 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 the normal way through contributions to the ITU-T.

6.3 Policies for Updating this Document

This document is managed by the ITU-T Study Group 8 Question 10 Rapporteur's Group. It can be revised at any recognized Q.10/8 Rapporteur's Group meeting provided the proposed revisions are unanimously accepted by the members of the group. A revision history cataloguing the evolution of this document is included.

6.4 Defect Resolution Procedure

Upon discovering technical defects with any components of the T.120 Recommendations series, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q.10/8 Rapporteur. The template for a defect report is enclosed. Contact information for these parties is included in this 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 T.120-series Recommendations. Formal membership in the ITU is not required to participate in this process.

6.5 References

This document refers to the following T.120-series Recommendations:

– ITU-T Recommendation T.120 (1996), *Data Protocols for Multimedia Conferencing*

- ITU-T Recommendation T.121 (1996), *Generic Application Template*
- ITU-T Recommendation T.122 (1993), *Multipoint Communication Service for Audiographic and Audiovisual Conferencing Service Definition*
- ITU-T Recommendation T.123 (1996), *Network Specific Data Protocol Stacks for Multimedia Conferencing*
- ITU-T Recommendation T.124 (1995), *Generic Conference Control*
- ITU-T Recommendation T.125 (1994), *Multipoint Communication Service Protocol Specification*
- ITU-T Recommendation T.126 (1995), *Multipoint Still Image and Annotation Protocol*
- ITU-T Recommendation T.127 (1995), *Multipoint Binary File Transfer Protocol*

6.6 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.

6.7 Technical and Editorial Corrections

6.7.1 Technical and Editorial Corrections to ITU-T Recommendation T.120

— None —

6.7.2 Technical and Editorial Corrections to ITU-T Recommendation T.121

— None —

6.7.3 Technical and Editorial Corrections to ITU-T Recommendation T.122

— None —

6.7.4 Technical and Editorial Corrections to ITU-T Recommendation T.123

6.7.4.1 Use of V.70 UNERM tunneling under PSTN basic profile

Description: V.70 Appendix II suggests alternative protocol stacks to support T.120 using DSVD modems. Alternative A is an attractive solution because it achieves more efficient transmission by mapping Q.922 framing (flags and byte stuffing) onto the corresponding mechanisms of V.76 instead of carrying it redundantly. Furthermore, Alternative A can be considered within the scope of the PSTN basic profile already part of T.123. To make this perfectly clear, a note to that effect will be added in the next edition.

Alternative B is not endorsed, because it does not provide error correction all the way from DTE to DTE. There is also a mismatch trying to apply the SCF for Q.922 to the negotiation of V.76 data link connections. Alternative C also lacks end-to-end error correction and is not ready for use anyway, as its SCF is marked for further study.

[Begin Correction]

6.7.4 PSTN basic profile

Figure 8 defines the PSTN basic profile. Layers above Q.922 are identical to the ISDN basic profile.

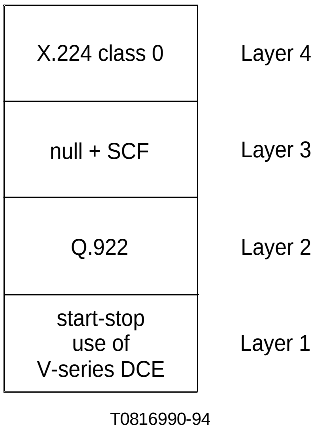


FIGURE 8/T.123

PSTN basic profile

Layer 4

- As specified in 7.1.

Layer 3

- As specified in 7.1.

Layer 2

- Q.922.
- Protocol parameters and options as specified in clause 10.
- Modified frame transparency based on ISO 3309, as specified in clause 11.

Layer 1

- Start-stop transmission by DTE.
- When using V.14: one start bit, one stop bit, eight data bits, no parity.
- Any compatible V-series DCE operating over PSTN may be employed.
- The DTE and DCE may be logical functions that are not physically separated, if integrated equipment can produce the same transmitted signals.
- The choice of V-series DCE is unrestricted and includes, for example, V.34, V.61, and V.70 modems, with optional use of V.42 and V.42bis. Selection of a compatible operating mode may be assisted by V.8 or V.8bis.

NOTE 1 - If the error control function of V.42 is activated, system parameters should be set to avoid adverse interaction with the error correcting operation of Q.922. Important elements are the acknowledgement timer, the maximum number of octets in an information field, and the data forwarding conditions.

NOTE 2 - The effectiveness of V.42bis data compression will vary, depending on how much of the application data exchanged in a conference has already been compressed by other means.

NOTE 3 - V.70 DCE, if made aware that this profile is being employed, may negotiate between themselves the use of enhanced techniques, like UNERM tunnelling for T.120, as long as the service provided at the DTE interface remains start-stop transmission.

[End Correction]

6.7.5 Technical and Editorial Corrections to ITU-T Recommendation T.124

6.7.5.1 Correction of Errors in the Bounds of PDU Parameters

Description: Three typographical errors have been identified in the bounds specifications for T.124 PDU parameters.

These PDUs will be printed in their corrected form in the T.124 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final T.124 document that was submitted for approval in 1995.

As these errors appear in the PDU definitions, failure to correct these errors would result in an incompatible implementation. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

[Begin Correction]

8.7 GCCPDU definitions

...

```
RosterUpdateIndication ::= SEQUENCE
{                                                -- MCS-Send-Data on Node ID Channel or
                                                 -- MCS-Uniform-Send-Data on GCC-Broadcast-Channel
        fullRefresh             BOOLEAN,         -- Conference Roster and all
                                                 -- ApplicationProtocol Sessions refreshed
        nodeInformation         SEQUENCE
        {
```

```
      nodeRecordList    CHOICE
      {
            noChange    NULL,
            refresh     SET (SIZE (1..65536)) OF SEQUENCE
                                          -- One for each node in the conference;
                                          -- no node shall be listed more than once
            {
                  nodeID        UserID,   -- Node ID of the node
                  nodeRecord    NodeRecord
            },
            update      SET (SIZE (1..65536)) OF SEQUENCE
                                          -- One for each node changing its node record;
                                          -- no node shall be listed more than once
            {
                  nodeID        UserID,   -- Node ID of the node
                  nodeUpdate    CHOICE
                  {
                        addRecord         NodeRecord,
                        replaceRecord     NodeRecord,
                        removeRecord      NULL,
                        ...
                  }
            },
            ...
      },
      rosterInstanceNumber    INTEGER (0..65535),
      nodesAdded              BOOLEAN,    -- Nodes have been added since last instance
      nodesRemoved            BOOLEAN,    -- Nodes have been removed since last instance
      ...
} ,
applicationInformation  SET (SIZE (0..65535~~6~~)) OF SEQUENCE
                                          -- One for each Application Protocol Session;
                                          -- all Application Protocol Sessions if full refresh;
                                          -- no Application Protocol Session shall be
                                          -- listed more than once
{
      sessionKey              SessionKey,
      applicationRecordList   CHOICE
      {
            noChange    NULL,
            refresh     SET (SIZE (0~~1~~..65535~~6~~)) OF SEQUENCE
                                          -- One for each node with the
                                          -- Application Protocol Session enrolled;
                                          -- no node shall be listed more than once
            {
                  nodeID      UserID,     -- Node ID of node
                  entityID    EntityID,   -- ID for this Application Protocol Entity at this node
                  applicationRecord ApplicationRecord
            },
            update      SET (SIZE (1..65536)) OF SEQUENCE
                                          -- One for each node modifying its Application Record;
                                          -- no node shall be listed more than once
            {
                  nodeID      UserID,     -- Node ID of node
                  entityID    EntityID,   -- ID for this Application Protocol Entity at this node
                  applicationUpdate CHOICE
                  {
                        addRecord       ApplicationRecord,
                              replaceRecord     ApplicationRecord,
                              removeRecord      NULL,
                              ...
                         }
                    },
                    ...
               },
               applicationCapabilitiesList        CHOICE
               {
                    noChange             NULL,
                    refresh              SET OF SEQUENCE
                                         {
                                                  capabilityID                CapabilityID,
                                                  capabilityClass             CapabilityClass,
                                                  numberOfEntities            INTEGER (1..65536),
                                                       -- Number of Application Protocol Entities
                                                       --  which issued the capability
                                                  ...
                                         },
                    ...
               },
               rosterInstanceNumber      INTEGER (0..65535),
               peerEntitiesAdded         BOOLEAN,        -- Peer Entities have been added since last instance
               peerEntitiesRemoved       BOOLEAN,        -- Peer Entities have been removed since last instance
               ...
          },
          ...
}
                                                  ...


ConductorPermissionGrantIndication ::= SEQUENCE
{                                                 -- MCS-Uniform-Send-Data on GCC-Broadcast-Channel
          permissionList            SEQUENCE (SIZE (0..655355~~6~~)) OF UserID,
                                                  -- Node ID of nodes granted permission
          waitingList               SEQUENCE (SIZE (1..65536)) OF UserID OPTIONAL,
                                                  -- Node ID of nodes waiting form permission
          ...
}
```

[End Correction]

6.7.5.2 Clarification of the Use of Numeric vs. Text Passwords and Conference Names

Description: Due to the use of both numeric as well as text variants of the Password, Convener Password, and Conference Name parameters, for primitives using only a choice of one of these types (in the GCC-Conference-Join and GCC-Conference-Transfer primitives), if the value of the parameter has a numeric value, it is unclear which parameter field should be used to convey this information. This is because during conference creation, the text version as currently defined could contain a string which happens to include only numeric digits.

To correct this ambiguity, it is suggested that the Node Controller be restricted in use of these parameters such that a parameter containing only numeric information must only be placed in the numeric field.

This effectively restricts the text parameter used at the time of conference creation to contain only values that have at least one non-numeric character. Otherwise, that value would never be tested against later.

Also, the ability to enter text strings to use in these parameters should allow for the possibility that the comparison of these parameters is case sensitive. This implies the ability to source either upper or lower case characters.

[Begin Correction]

7.1.2.1 GCC-Conference-Create

...

Conference Name: Name by which the conference to be created is identified. This consists of a numerical string along with an optional Unicode Row 00 text string, from zero to 255 characters each. If both forms of a Conference Name are used, if a node wishes to join this conference, it may specify either form of the name in the join request. In the join request, a numeric value will necessarily be included in numeric variant of the Conference Name. As a result, use of a text Conference Name including only numeric characters will never be compared against and therefore should not be used - that is, the text variant of the Conference Name should include at least one non-numeric character.

Conference Name Modifier: If the requesting or responding node is already joined to a conference with the same Conference Name (either numerical or text portion) as that included in the request, this parameter shall also be included in the corresponding request or response primitive. The value of this parameter shall be unique among all conferences at the corresponding node which have this Conference Name. This modifier, if included, shall be used as the Called Node Conference Name Modifier parameter in a GCC-Conference-Join request by another node attempting to join the conference through a direct connection with the corresponding node. This modifier is also included in the response to a GCC-Conference-Query directed at this node. This parameter is a numerical string up to 255 digits in length.

Conference ID: Locally allocated identifier of the newly created conference. All subsequent references to the conference are made using the Conference ID as a unique identifier. The Conference ID shall be identical with the MCS Domain Selector used locally to identify the MCS Domain associated with the conference.

Convener Password: This optional parameter contains a numeric string, as well as an optional Unicode Row 00 text string, used for the convener to identify itself in later operations, allowing the convener to disconnect and later rejoin the conference, maintaining convener privileges (only when rejoined with a direct connection to the Top GCC Provider). This is the private password which will allow the convener to perform convener-only operations (maximum 255 digits and 255 characters). If this parameter is NULL, then it is not possible for the convener to disconnect and later rejoin maintaining convener privileges. In the join request, a numeric value will necessarily be included in the numeric variant of the Convener Password. As a result, use of a text Convener Password including only numeric characters will never be compared against and therefore should not be used - that is, the text variant of the Convener Password should include at least one non-numeric character.

Password: This is a numeric string, as well as an optional Unicode Row 00 text string, to serve as a Password to enter the conference (maximum 255 digits and 255 characters). If no Password is specified, the conference is not Password protected. In the join request, a numeric value will necessarily be included in the numeric variant of the Password. As a result, use of a text Password including only numeric characters will never be compared against and therefore should not be used - that is, the text variant of the Password should include at least one non-numeric character.

NOTE - If the conference is Password protected, the Node controller must specify a numeric Password, and may also specify a text Password. The numeric Password is required to allow for nodes which have no suitable text entry mechanism. In the case that a text password is used, there is no assumption that the numeric Password that must also be included is generated by the user. It may be more convenient and secure to use a machine generated numeric Password.

...

7.1.2.3 GCC-Conference-Join

...

Conference Name: Name of the conference being joined. In the request, this parameter is either a numeric string or a Unicode Row 00 text string, maximum 255 characters. If both the numeric and text parts of the Conference Name were used at the time the conference was created, the Node Controller shall determine whether to send the Conference Name as a text string or numeric string based on its value. A value consisting only of numeric digits shall be specified as a numeric string, while a value including at least one non-numeric character shall be specified as a text string~~either part may be included in this request~~. In the confirm, this parameter includes the full Conference Name including both numerical and text forms, if both were used at the time the conference was created.

Called Node Conference Name Modifier: If the node directly connected to the joining node (the node to which a connection is attempting to be established) has included a Conference Name Modifier as part of the name by which this conference is known, this parameter shall be included in the request primitive and shall indicate the Conference Name Modifier as it is known to the node directly connected to the joining node. This parameter is a numeric string up to 255 digits in length.

Calling Node Conference Name Modifier: If a conference already exists at the node issuing the join request with a name identical with the Conference Name of the conference to be joined, this parameter shall be included in the GCC-Conference-Join request and shall indicate the Conference Name Modifier by which the conference shall be known at the local node. This parameter, if included, shall be different from any Conference Name Modifier already in use for any other conference the local node is currently joined to with the same Conference Name. If used, this parameter becomes the Called Node Conference Name Modifier by which another node attempting to join this conference through a connection to the local node refers to this conference. This modifier is also included as the Conference Name Modifier parameter in any GCC-Conference-Query response from this node (if the conference is listed). This parameter is a numeric string up to 255 digits in length.

Conference ID: At the Top GCC Provider (the indication/response primitives) this parameter is the Conference ID of the conference to which the requesting node wishes to join. In the confirm primitive, this parameter is returned by GCC indicating the locally allocated ID by which all subsequent references to the conference are indicated. The Conference ID shall be identical with the MCS Domain Selector used locally to identify the MCS Domain associated with the conference.

Convener Password: This is an optional parameter which is either a numeric string or a Unicode Row 00 text string which may be used by a conference convener rejoining a conference after disconnecting (maximum 255 digits or characters). If this identifier matches the corresponding identifier used when the conference was created, the joining node is given the privileges of the convener, but only if joining with a direct connection to the Top GCC Provider (rather than via an intermediate MCU). The convener, with the correct Convener Password, is allowed to join even conferences which are locked. If the conference is Password protected, the correct Password must be given in addition to the Convener Password to successfully join the conference. If the Convener Password is present but does not match, the request to join shall be rejected. The criterion used to determine if the Convener Password matches the originally specified value is determined by the Node Controller. The Node Controller shall determine whether to send the Convener Password as a text string or numeric string based on its value. A value consisting only of numeric digits shall be specified as a numeric string, while a value including at least one non-numeric character shall be specified as a text string.

Password: The password parameter is used to gain access to a password protected conference. In the request form of this primitive, this parameter shall only contain a password if a result of Challenge Response Required has been received in a previous GCC-Conference-Join confirm for this conference or if the Password In The Clear Required parameter is set in the Conference Descriptor for this conference in a previous GCC-Conference-Query confirm. In the case of a password in the clear this is either a numeric string or a Unicode Row 00 text string (maximum 255 digits or characters). A text string may only be used if a text password was defined at the time of conference creation in addition to the numeric password. The Node Controller shall determine whether to send the Password as a text string or numeric string based on its value. A value consisting only of numeric digits shall be specified as a numeric string, while a value including at least one non-numeric character shall be specified as a text string. In the case of an encrypted password this parameter contains the password encoded using one of the algorithms specified in the previously received challenge. In the case of a password sent in response to a challenge (either in the clear or encrypted), this parameter shall also include a tag which shall be identical to the tag received in the challenge. In the case of a password initiated in response to the Password In The Clear Required flag in the GCC-Conference-Query indication, no tag is required. In the request form of this primitive, this parameter may also include a challenge to the receiving node. There are no restrictions on when a challenge may be included in this parameter.

In the response form of this primitive, this parameter may contain a challenge to the requester indicating that a password is required for joining this conference. For this case, this parameter includes information specifying which forms of the password will be accepted (either in the clear, and/or encrypted via a list of non-standard encryption algorithms), an integer tag used to identify this challenge, and any additional information required for encryption. Should this parameter contain challenge, the result parameter of this primitive shall be set to Challenge Response Required. In this case, no connection is established by this exchange. This parameter in the response form of this primitive may also include a Password (either in the clear or encrypted) in response to a challenge by the requesting node.

If this parameter in the indication is not in a format satisfactory to the receiving node, that node should issue a response with Invalid Challenge Response as the result. If this parameter in the indication is of the correct format, but does not contain the correct password, the response should include Invalid Password as the result.

...

7.1.2.12 GCC-Conference-Transfer

...

Password: This parameter indicates the password that the transferring nodes shall use in the GCC-Conference-Join request to join the new conference. This is a numeric string or a Unicode Row 00 text string (maximum 255 digits or characters). The Node Controller shall determine whether to send the Password as a text string or numeric string based on its value. A value consisting only of numeric digits shall be specified as a numeric string, while a value including at least one non-numeric character shall be specified as a text string. This parameter shall only be used if the Password In The Clear Required flag is set for this conference.

[End Correction]

6.7.5.3 Correction of Object Identifier

Description: In the Object Identifier used to define the T.124 protocol version, the keyword "itu" should properly be "itu-t".

[Begin Correction]

ANNEX B

(this annex forms an integral part of this Recommendation)

Object Identifier Assignments

Table B-1 lists the assignment of Object Identifiers defined for use by this Recommendation.

TABLE B-1

Object Identifier Value	Description
{itu-t recommendation t 124 version(0) 1}	This Object Identifier is used to indicate the version of this Recommendation in use as the MCS Controller. At this time there is a single standardized version defined.

[End Correction]

6.7.5.4 Indication to Join a Default Conference or to Wait for a Conference

Description: There are situations where an MCU accepting incoming connections may wish for a particular connection to exclusively join a particular conference. The conference to be joined may have been determined by out-of-band mechanisms such as reservation of a particular physical port for a given conference. In such cases, the current specification of T.124 allows only two possibilities: either the MCU invites the connecting node into a conference, or the connecting node manually selects a conference to join. In the former case, there is no provision for security. In the latter case, the MCU could artificially limit the list of conferences to join to one, but this may still require unnecessary user intervention at the connecting node to actually join (just because there is a single conference listed in the query indication, doesn't mean that this is the one that the user should join - the user may wish to join an unlisted conference or a conference which has not yet been created).

While it would be possible to add an optional security mechanism to the invite primitive in order to correct this situation, a much simpler correction would be to add an optional field to the query response/confirm to allow a particular conference to be marked as the "default" conference which would indicating that the connecting node should attempt to join this conference. The later approach is recommended.

It has also been deemed necessary to allow the MCU to indicate to a node that the MCU intends to invite the node into a conference. In some cases, this invitation may not be immediate. In order to avoid the possibility of the node disconnecting from the MCU if no response is given, and to avoid the need for the node prompting the user unnecessarily to join or create a conference, a flag is proposed to be added which can indicate the intent of the MCU in this case.

Finally, in order for the MCU to indicate its intent as to the types of conferences available to be joined, it is useful to add an optional parameter that indicates if no unlisted conferences are available.

[Begin Correction]

7.1.2.2 GCC-Conference-Query

...

TABLE 7-1

GCC-Conference-Query - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Response**	**Confirm**
Node Type	M	M(=)	M	M(=)
Asymmetry Indicator	C	C(=)	C	C(=)
Conference Descriptor List				C
Wait for Invitation Flag			O	O
No Unlisted Conference Flag			O	O
Calling Address	O	O(=)		
Called Address	O	O		
User Data	O	O(=)	O	O(=)
Result			M	M(=)

...

TABLE 7-3

Contents of a Conference Descriptor

Parameter	**Description**
Conference Name	Conference Name of the conference. If the requesting node wishes to join this conference, this parameter is the value that shall be used in the Conference Name parameter of the GCC-Conference-Join request. This parameter is a numerical string along with an optional Unicode Row 00 text string, maximum 255 characters each. If both forms of the name are given, either form may be specified in the join request.
Conference Name Modifier (conditional)	If at the node returning the response, the conference is known by a name which includes a Conference Name Modifier, this parameter is included. If the requesting node wishes to join this conference, this is the Conference Name Modifier that shall be used in the Called Node Conference Name Modifier parameter of the GCC-Conference-Join request. This parameter is a numerical string up to 255 digits in length.
Conference Description (conditional)	An optional Unicode text string, up to 255 characters in length, used to describe the conference. This parameter may be particularly useful in cases where more than one conference in the Conference Descriptor List has the same Conference Name as a means of distinguishing between these conferences.
Locked/Unlocked	Flag indicating whether the conference is currently locked or unlocked.
Password In The Clear Required	Indicates that the conference is password protected with a password that may be used without encryption in a GCC-Conference-Join request without first being challenged for the password.
Network Address (conditional)	Address information provided to the requesting node. This is provided only if the optional Network Address parameter had been included in the connection establishment primitive at that node (either GCC-Conference-Create, GCC-Conference-Join, or GCC-Conference-Invite).
Default Conference Flag	An optional flag indicating whether a particular conference should be considered the default conference to join. In a meet-me conference where the user is expected to manually choose a conference to join from a list, this parameter should be FALSE for all conferences. In a meet-me conference where *a priori* information allows an MCU to determine which conference should be joined (and security reasons, for example, preclude the use of GCC-Conference-Invite), this flag may be set to TRUE for one and only one conference. If this flag is TRUE for more than one conference, or is TRUE for a Locked conference, it should be ignored by the receiver.

...

Wait for Invitation Flag: Optional flag that may be set by an MCU. When TRUE this flag indicates that the receiving Node should wait to receive an invitation to a conference and need not attempt to join or create a conference. The absence of this flag, or a FALSE value, does not imply whether or not attempts to join or create a conference should be made. This flag shall not be set to TRUE by a terminal or multiport terminal. No time-limit is implied by a TRUE value for this flag. A node wishing to determine whether the MCU continues to intend it to wait for the invitation may re-issue a GCC-Conference-Query request in order to receive the current setting of this flag in the confirm.

No Unlisted Conference Flag: Optional flag that when TRUE indicates that there are no unlisted conferences available to be joined. Absence of this flag, or a FALSE value, does not imply any information as to whether or not unlisted conferences are available.

Calling Address: Optional address to be included in the MCS-Connect-Provider primitive on establishing an MCS connection.

...

8.7 GCCPDU Definitions

...

```
ConferenceDescriptor       ::= SEQUENCE
{
        conferenceName                  ConferenceName,
        conferenceNameModifier          ConferenceNameModifier OPTIONAL,
        conferenceDescription           TextString OPTIONAL,
        lockedConference                BOOLEAN,
        passwordInTheClearRequired      BOOLEAN,
        networkAddress                  NetworkAddress OPTIONAL,
        ...,
        defaultConferenceFlag           BOOLEAN
}
```

...

```
ConferenceQueryResponse ::= SEQUENCE
{                                               -- MCS-Connect-Provider response user data
        nodeType                        NodeType,
        asymmetryIndicator              AsymmetryIndicator OPTIONAL,
        conferenceList                  SET OF ConferenceDescriptor,
        result                          ENUMERATED
        {
                success             (0),
                userRejected        (1),
                ...
        },
        userData                        UserData OPTIONAL,
        ...,
        waitForInvitationFlag           BOOLEAN OPTIONAL,
        noUnlistedConferenceFlag        BOOLEAN OPTIONAL
}
```

[End Correction]

6.7.5.5 Typographical Errors

Description: There a number of typographical errors that have been found in the T.124 text. Corrections to these errors are proposed here.

Note that the change to Figure 8-5 is shown as a replacement of the upper figure with the figure below. This is due to the inability to use change-bars within figures.

[Begin Correction]

7.1.2.3 GCC-Conference-Join

...

TABLE 7-4

GCC-Conference-Join - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Response**	**Confirm**
Conference Name	M			M
Called Node Conference Name Modifier	C			C(=RQ)
Calling Node Conference Name Modifier	C			C(=RQ)
Conference ID		M	M(=)	M
Convener Password	O	O(=)		
Password	C	C(=)	C	C(=)
Caller Identifier	O	O(=)		
Calling Address	O	C~~O~~(=)		
Called Address	O	C~~O~~		
...	**...**	**...**	**...**	**...**

...

Calling Address: Optional address to be included in the MCS-Connect-Provider primitive on establishing an MCS connection. See Recommendation T.122 for the interpretation of this parameter. The presence of this parameter in the indication is conditional. It is present only when the Node initiating the GCC-Conference-Join request is attempting to establish a direct connection to the Node containing the Top GCC Provider for the conference being joined.

Called Address: Optional address to be included in the MCS-Connect-Provider primitive on establishing an MCS connection. See Recommendation T.122 for the interpretation of this parameter. The presence of this parameter in the indication is conditional. It is present only when the Node initiating the GCC-Conference-Join request is attempting to establish a direct connection to the Node containing the Top GCC Provider for the conference being joined.

...

7.1.2.10 GCC-Conference-Terminate

...

TABLE 7-13

GCC-Conference-Terminate - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Confirm**
Conference ID	M	M	M(=RQ)
~~Requesting Node ID~~		~~C~~	
Reason for termination	O	O(=)	
Result			M

Conference ID: Identifier of the conference to which the primitive refers.

~~*Requesting Node ID:* Node ID of the node which initiated the termination. This parameter is present except in the case of GCC initiated termination.~~

Reason for termination: Indication of the reason for termination of the conference. This contains one of a list of possible reasons: requested normal termination, requested timed conference termination, no more participants in automatically terminating conference, error termination.

...

7.1.2.12 GCC-Conference-Transfer

The GCC-Conference-Transfer request primitive is used by a Node Controller to cause selected nodes in a conference to disconnect from that conference and join another conference. This primitive shall only be issued by the conference Convener or a convener-designated node. Some MCUs in a conference may be already joined to both the originating and destination conferences prior to the transfer taking place. If so, these MCUs shall not be included in the list of Transferring~~Destination~~ Nodes in this request. If an MCU is not joined with both conferences but will be connected to nodes which, after the transfer, will be joined to both conferences (if not all nodes are transferred), then that MCU shall be joined to the destination conference prior to issuing the GCC-Conference-Transfer request. Any MCU which is included in the list of Transferring~~Destination~~ Nodes (those which are intended to transfer) shall complete the transfer operation (disconnecting from the originating conference and joining the destination conference) prior to processing any new GCC-Conference-Join indications. This allows the join request from the nodes below that MCU in the connection hierarchy to be successfully completed. Nodes which are in the process of transferring and receive a GCC-Conference-Terminate indication for the originating conference may proceed directly to joining the destination conference without disconnecting if they have not already done so. This situation could arise if the MCU to which the node is connected was also instructed to transfer. Table 7-15 shows the parameters and types of this primitive. Figure 7-14 shows the sequence of events when using this primitive.

...

Destination Conference Name: Name of the conference to which the designated nodes are instructed to join. This is the name by which the conference is known at the MCU or MCUs to which the transferring~~destination~~ nodes are connected. If the conference had been created using both the numerical and text forms of the Conference Name, either form may be used in this primitive.

...

7.1.3 Conference Establishment Requirements

...

- When establishing calls among the various node types (terminals, multiport terminals, and MCUs), ~~Table 7-16~~ constraints are placed on the actions of the calling node in initiating the initial connection. Table 7-16 shows these constraints for each permutation of calling and called node type. The definition of the actions shown in the table, numbered 1, 2, 3, and 4 are as follows:

...

7.3.3.3 GCC-Application-Roster-Report

...

TABLE 7-27

Contents of the Updated Application Roster for each Application Protocol Session

Parameter	**Description**
Session Key	The Session Key (including the Session ID, if any) designating the particular Application Protocol Session.
Application Protocol Entity List Updated Flag	A flag indicating whether the Application Protocol Entity List for this Session Key has been updated in this report. If so, the Application Protocol Entity List is included as the following parameter.
Application Protocol Entity List (conditional)	A list of the Application Protocol Entities enrolled in the conference as part of this Application Protocol Session including an Application Record for each. The contents of each entry in this list is shown in Table 7-28.
Instance Number	The instance number for the Application Roster for this Session Key. This is a 16-bit number which is incremented modulo 2^{16} each time the contents of the Application Roster changes for this Session Key. This allows Application Protocol Entities to perform operations with respect to a particular capability set which may be in the process of changing avoiding any race conditions.
...	...

...

7.4.2 Ownership and Persistence

...

Registry entries are not deleted automatically when the owner unenrolls~~detaches~~ from the conference. Their content persists unchanged indefinitely. However, ownership of an entry is removed when the owner unenrolls~~detaches~~. This allows a surviving Application Protocol Entity to modify (for a Parameter entry only) or delete an orphaned entry if its usefulness has expired. The first Application Protocol Entity that requests to store an item in an orphaned Parameter entry becomes its new owner.

...

7.4.4.6 GCC-Registry-Monitor

...

TABLE 7-38

GCC-Registry-Monitor - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Confirm**
Conference ID	M	M~~(=)~~	M(=RQ)
Enable/Disable	M		M(=)
Registry Key	M	M~~(=)~~	M(=RQ)
Registry Item		M	
Owner		M	
Modification Rights		C	
Result			M

...

8.3.10 An Example of a Roster Update

...

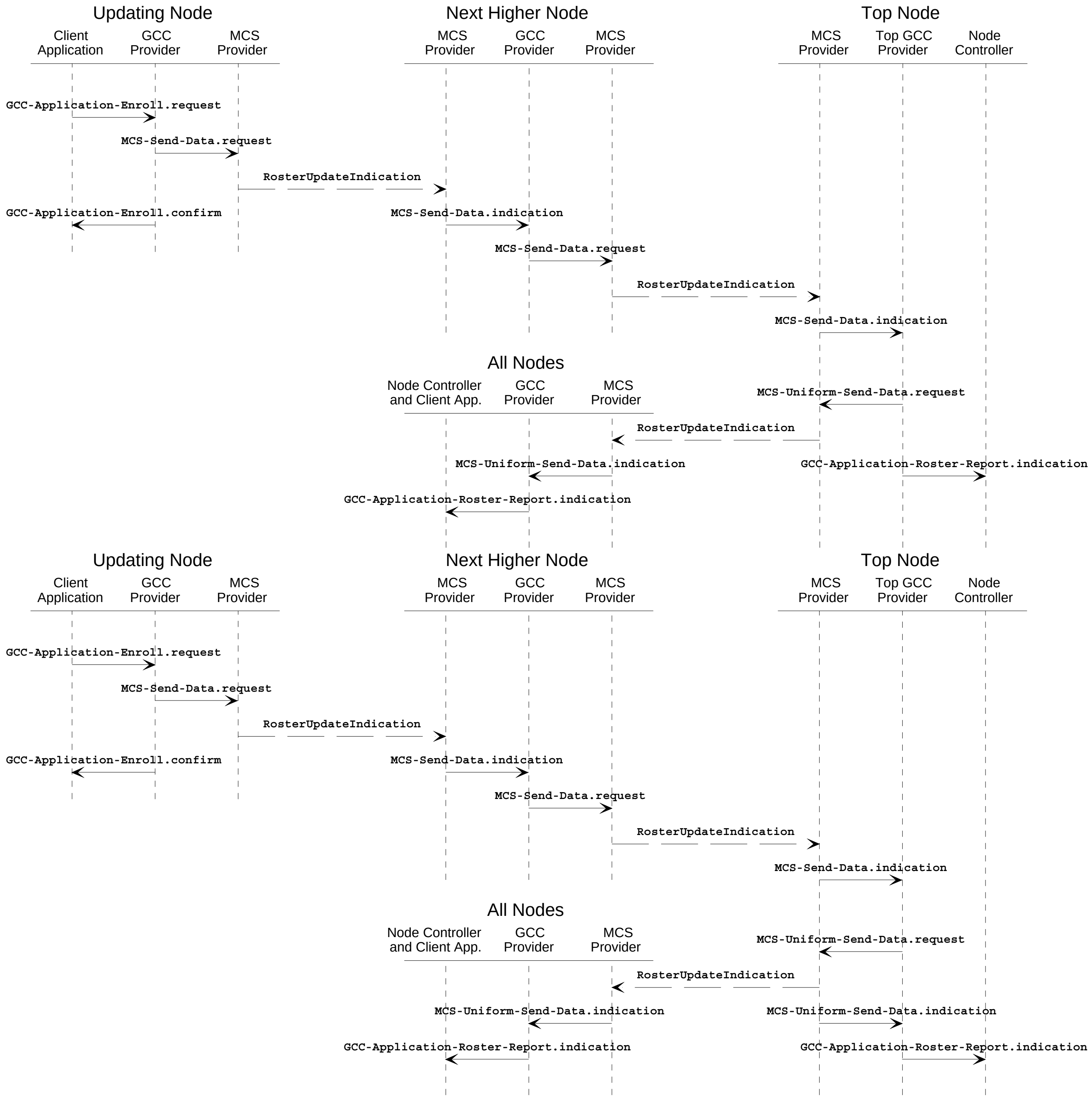


FIGURE 8-5

An example of updating the Application Roster

[End Correction]

6.7.5.6 Conference Priority Indication

Description: In order to support proposed mechanisms for specialized conference establishment procedures that may be included in future versions of T.124, and to allow similar non-standard conference establishment procedures, it is necessary to provide a Conference Priority parameter in the GCC-Conference-Invite and GCC-Conference-Create primitives. Given that there may be a multiplicity of procedures for making use of this parameter, it is also necessary to provide an associated indication of the scheme in which this parameter is to be interpreted.

[Begin Correction]

7.1.2.1 GCC-Conference-Create

...

TABLE 7-1

GCC-Conference-Create - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Response**	**Confirm**
...	**...**	**...**	**...**	**...**
Local Network Address	O		O	
Conference Priority	O	O(=)		
User Data	O	O(=)	O	O(=)
Result			M	M(=)

...

Local Network Address: If included in either the request or response, the local GCC Provider at the corresponding node shall use this information to include as the Network Address parameter in the Conference Descriptor List sent as part of the response to a GCC-Conference-Query request from another node.

Conference Priority: An optional parameter to specify the priority of a conference. This may be used in some situations to determine whether or not the indication should be accepted. The parameter includes two subparameters: the priority and the scheme. The priority is an integer value ranging from zero to 65535. The scheme indicates the procedures by which the priority value is to be interpreted. At this time, only non-standard schemes are supported. Standardized procedures for interpretation of this parameter are for further study.

...

7.1.2.4 GCC-Conference-Invite

...

TABLE 7-5

GCC-Conference-Invite - Types of primitives and their parameters

Parameter	**Request**	**Indication**	**Response**	**Confirm**
...	...	...	...	...
Local Network Address			O	
Conference Priority	O	O(=)		
User Data	O	O(=)	O	O(=)
Result			M	M(=)

...

Local Network Address: If included in the response, the local GCC Provider at the corresponding node shall use this information to include as the Network Address parameter in the Conference Descriptor List sent as part of the response to a GCC-Conference-Query request from another node.

Conference Priority: An optional parameter to specify the priority of a conference. This may be used in some situations to determine whether or not the indication should be accepted. The parameter includes two subparameters: the priority and the scheme. The priority is an integer value ranging from zero to 65535. The scheme indicates the procedures by which the priority value is to be interpreted. At this time, only non-standard schemes are supported. Standardized procedures for interpretation of this parameter are for further study.

...

8.7 GCCPDU Definitions

...

```
TerminationMethod ::= ENUMERATED
{
        automatic   (0),
        manual      (1),
        ...
}

ConferencePriorityScheme ::= CHOICE
{
        nonStandardScheme           nonStandardParameter,
        …
}

ConferencePriority ::= SEQUENCE
{
        priority            INTEGER (0..65535),
        scheme              ConferencePriorityScheme,
        …
}
```

...

```
ConferenceCreateRequest        ::= SEQUENCE
{                                                    -- MCS-Connect-Provider request user data
        conferenceName              ConferenceName,
        convenerPassword            Password OPTIONAL,
        password                    Password OPTIONAL,
        lockedConference            BOOLEAN,
        listedConference            BOOLEAN,
        conductibleConference       BOOLEAN,
        terminationMethod           TerminationMethod,
        conductorPrivileges         SET OF Privilege OPTIONAL,
        conductedPrivileges         SET OF Privilege OPTIONAL,
        nonConductedPrivileges      SET OF Privilege OPTIONAL,
        conferenceDescription       TextString OPTIONAL,
        callerIdentifier            TextString OPTIONAL,
        userData                    UserData OPTIONAL,
        ...,
        conferencePriority          ConferencePriority OPTIONAL

}
```

...

```
ConferenceInviteRequest  ::= SEQUENCE
{                                                    -- MCS-Connect-Provider request user data
        conferenceName              ConferenceName,
        nodeID                      UserID,         -- Node ID of the sending node
        topNodeID                   UserID,         -- Node ID of Top GCC Provider
        tag                         INTEGER,
        passwordInTheClearRequired  BOOLEAN,
        lockedConference            BOOLEAN,
        listedConference            BOOLEAN,
        conductibleConference       BOOLEAN,
        terminationMethod           TerminationMethod,
        conductorPrivileges         SET OF Privilege OPTIONAL,  -- No privilege shall be listed more than
                                    once
        conductedPrivileges         SET OF Privilege OPTIONAL,  -- No privilege shall be listed more than
                                    once
        nonConductedPrivileges      SET OF Privilege OPTIONAL,  -- No privilege shall be listed more than
                                    once
        conferenceDescription       TextString OPTIONAL,
        callerIdentifier            TextString OPTIONAL,
        userData                    UserData OPTIONAL,
        ...,
        conferencePriority          ConferencePriority OPTIONAL
}
```

[End Correction]

6.7.5.7 Relationship of T.120 to H.243 in H.320 Conferences

Description: In H.320 multipoint conferences, H.243 is currently used to perform some basic conference establishment functions. For H.320 terminals and MCUs that also support T.120, it is currently completely ambiguous what procedures should be used for conference establishment. In particular, it has been left unclear how the procedures of T.124 and H.243 should be used, or not used, in this case.

The following is a proposed Appendix to T.124 that recommends procedures for use of T.124 and H.243 as part of an H.320/T.120 multipoint conference in order to remove this ambiguity.

[Begin Correction]

APPENDIX I

(this appendix does not form an integral part of this Recommendation)

Relationship of T.120 to H.243 in H.320 Conferences

I.1 Introduction

It is the intent of T.120 is to eventually take over the responsibilities of H.243 for multipoint conference establishment in H.320 audio/video/data conferences. In the meantime, H.243 and T.120 will continue to coexist. This Appendix provides recommendations as to how H.243 and T.120 should relate during the long transitional phase from H.243 to T.120 conference control.

NOTE – H.243 Section 15 places normative requirements on terminals and MCUs in this regard.

I.2 Conference Selection and Password Protection

Both T.124 and H.243 provide means of establishing a logical connection from a terminal to a particular conference hosted at an MCU. Both Recommendations also provide a password protection mechanism for verifying the authority of a node to join the desired conference. It is envisioned that both T.124 and H.243 will continue to be used for some time. The following sections describe means by which T.124 and H.243 should be used during the conference establishment process.

The assumption is made that at most one H.243 conference is associated with each T.124 conference and vice versa. Thus, if a terminal joins either one, the choice of the other conference is then predetermined. If this assumption is not valid, then different procedures may apply.

For the purpose of selecting a conference to join (in the case that the MCU allows such a choice) and for the purpose of password protection, an MCU should choose to use the T.124 or the H.243 mechanisms, but not both. The choice of which mechanism to use may depend on whether the conference is dial-in or dial-out, may be made separately for each MCU port, and may depend on the capabilities of the connecting terminal on each port. In the case of terminals that support T.120, it is preferred that the T.124 mechanisms be used for these purposes rather than the H.243 mechanism.

I.2.1 T.124 Conference Establishment

The MCU may choose to use the T.124 mechanisms for conference establishment for a given terminal. The MCU may, for example, allow the terminal to choose among a list of conferences to join by providing this list in the GCC-Conference-Query response and allowing selection of a conference using the GCC-Conference-Join primitive. The MCU may instead have a predetermined conference that it wishes the connecting terminal to join. In this case, it should set the Default Conference Flag in the GCC-Conference-Query response's corresponding Conference Descriptor to indicate that the connecting terminal should join the indicated conference. In either case, if the MCU requires password protection, the T.124 password protection mechanisms (Password In The Clear Required or Challenge Response Required) should be used to verify that the connecting terminal is authorized to join the conference.

Alternatively, as may be the case for a prearranged dial-out, if the choice of conference is already determined and password protection is not required, the MCU can decide to take the initiative and to invite the terminal into a T.124 conference by issuing a GCC-Conference-Invite request. In this case, it should set the Wait for Invitation Flag in the GCC-Conference-Query response to indicate that the connecting terminal should not take action to join or create a T.124 conference.

Once the MCU receives a GCC-Conference-Join-Request with a conference name and the corresponding correct password or decides to take the initiative and to invite a terminal into a T.124 conference, the terminal may then start receiving the audio and video associated with the corresponding H.243 conference.

In this case, the MCU should not use the H.243 mechanism to request a password from the connecting terminal. H.243 may still be used for video control and other functions.

I.2.2 H.243 Conference Establishment

Alternatively, the MCU may choose to use the H.243 mechanism for conference establishment. In this case, if a terminal calls into an MCU, the MCU may use the H.243 password to determine in which conference the terminal belongs. Once a correct H.243 password is supplied, the terminal may start receiving the audio and video associated with the conference.

Since H.243 provides a password mechanism, the additional requirement of a T.124 conference password in this case is superfluous (assuming that only one T.124 conference is permitted in conjunction with the H.243 conference). The MCU may decide to take the initiative and to invite the terminal into the T.124 conference associated with the H.243 conference by issuing a GCC-Conference-Invite request. If queried by the terminal, the MCU should set the Wait for Invitation Flag in the GCC-Conference-Query response to indicate that the terminal need not take action to join or create a T.124 conference. Alternatively the MCU can wait for the terminal to issue a GCC-Conference-Join request. In this case, if queried by the terminal, the MCU should return a single Conference Descriptor in its GCC-Conference-Query response, with the Default Conference Flag set and the Password In The Clear Required Flag reset.

Terminals that support T.120 should allow the H.243 password mechanism to be used for conference establishment. This should be considered an interim approach used by MCUs until they support the T.124 conference establishment procedures described in the previous section.

I.3 Alternative Node ID

The Alternative Node ID of T.124 was envisioned for use with the terminal numbering of H.243, allowing nodes in an audio/video/data conference running both T.120 and H.243 to correlate T.124 conference roster information to H.243 site information.

When issuing the GCC-Conference-Announce-Presence request, nodes should include their assigned H.243 terminal number in the Alternative Node ID field. If a node's H.243 terminal number is reassigned during a conference, the node should re-issue a GCC-Conference-Announce-Presence request with the new terminal number contained in the Alternative Node ID field.

NOTE - This requires that each terminal remember the content of the most recently received H.230 C&I code TIA.

The Alternative Node ID is a two octet field. The first octet should contain the H.243 MCU ID (M), and the second octet should contain the H.243 Terminal ID (T).

[End Correction]

6.7.5.8 Error in the Description of Restrictions for Locked Conferences

Description: In the description of locked conferences in two portions of the T.124 text, the text mistakenly states that the only way for a node to be joined to a locked conference is via the **GCC-Conference-Add** primitive. This is a clear error since the GCC-Conference-Add primitive operates by making use of the GCC-Conference-Invite primitive and that references to using the GCC-Conference-Invite primitive for locked conferences appear in several portions of the Recommendation. The text should be corrected to state that the **GCC-Conference-Invite or GCC-Conference-Add** primitives may be used to invite nodes into locked conferences.

[Begin Correction]

7.1.2.1 GCC-Conference-Create

...

Conference Locked: Setting this flag immediately locks a conference, preventing anyone from joining this conference unless they are explicitly added using the GCC-Conference-Invite primitive (or indirectly inviting via the GCC-Conference-Add primitive). To lock a conference at any time after issuing this primitive, the primitive GCC-Conference-Lock may be used. To unlock a conference, GCC-Conference-Unlock may be used. ~~Since support of the GCC-Conference-Add and GCC-Conference-Unlock primitives are optional, a locked conference created at an MCU (which is not also a terminal) which did not support either of these primitives would be of little use. Such an MCU may reject the conference creation stating locked-conference-not-supported as the reason. If either GCC-Conference-Add indication/response (along with GCC-Conference-Invite request/confirm) or GCC-Conference-Unlock indication/response primitives are supported, an MCU shall allow creation of a locked conference and shall not make use of this reason for rejection.~~

...

7.1.2.6 GCC-Conference-Lock

The GCC-Conference-Lock request primitive may be used by a Node Controller to lock a conference, preventing other nodes from dialing into the conference at all. This primitive is valid only if issued by the Convener or a convener-designated node. While locked, participants may be added to a conference only by using the GCC-Conference-Invite primitive (or indirectly inviting via the GCC-Conference-Add primitive). The order of GCC-Conference-Lock and GCC-Conference-Unlock primitives exchanged between a node and the Top GCC Provider is preserved. Table 7-11 shows the parameters and types of this primitive. Figure 7-7 shows the sequence of events when using this primitive.

...

[End Correction]

6.7.5.9 Clarification of Signalling the Preference of the Called Node

Description: T.124 requires that a calling node be prepared to establish a connection using one of the defined connection establishment procedures. Section 7.1.3 in T.124 restricts the procedures that can be used based on the Node Type of the called and calling nodes.

It is sometimes the case that the calling node has *a priori* information that enables it to select among the allowed procedures when establishing a particular connection. On the other hand, it may also be the case that a node attempts to establish a connection without any *a priori* information that might allow it to choose among the possible procedures. In such cases, the calling node must analyse the possibilities offered by the called node in order to select which procedure is most appropriate for that call. For example, a multiport terminal calling another node without any *a priori* information may want to prompt the user to select a conference to join if the called node is hosting a meet-me conference, while it may want to simply connect automatically without user intervention if the called node has no such conference.

While the editor of T.124 had assumed that this indication of intent was clearly explained based on the descriptions of the parameters of the GCC-Conference-Query response primitive, the overwhelming response from implementors of T.124 indicate that how to interpret the GCC-Conference-Query response as a signal of the called node's intent is not clear at all.

It is necessary for a clear policy to be stated in T.124 on how this intent is to be signalled when a connection is being established. One such policy which is not clearly expressed in the published Recommendation is that a multiport terminal that does not intend to host a meet-me conference must not include any Unlocked conferences in its Conference Descriptor List. This means that when a conference is created that is not intended to be a meet-me conference, it must be locked, unlisted, or both.

[Begin Correction]

7.1.3 Conference Establishment Requirements

...

- The calling node shall be responsible for initiating the initial conference establishment procedure (either requesting creation of a new conference, joining a conference at the called node, or inviting the called node into a conference). Note that this, in general, does not preclude the called node from taking actions to establish a conference over the same physical connection as well, although care must be taken in this case to ensure that this action does not interfere with the action of the calling node. A called MCU may set the Wait for Invite Flag in the GCC-Conference-Query response and in doing so relieve the calling node of the responsibility of initiating a connection.
- When establishing calls among the various node types (terminals, multiport terminals, and MCUs), Table 7-16 constraints are placed on the actions of the calling node in initiating the initial connection. The definition of the actions shown in the table, numbered 1, 2, 3, and 4 are as follows:

  1) Calling node requests creation of a new conference at the Called node.
  2) Calling node attempts to join existing conference at the Called node.
  3) Calling node creates a conference locally and invites the Called node.
  4) Calling node invites the Called node into an existing conference.

TABLE 7-16

Actions of the Calling Node for Conference Establishment

Calling Node	**Called Node**		
	Terminal	Multiport Terminal	MCU
Terminal	Either 1 or 3.	Either 1 or 2.	Either 1 or 2.
Multiport Terminal	Either 3 or 4.	Either 1, 2, 3, or 4.	Either 1 or 2.
MCU	Either 3 or 4.	Either 3 or 4.	Either 1, 2, 3, or 4.

In some cases, the calling node may need to make use of information from the called node contained in the GCC-Conference-Query confirm to help choose among the allowed procedures defined in Table 7-16. For example, when a Terminal or Multiport Terminal calls a Multiport Terminal, it may make use of the presence or absence of unlocked conferences in the Conference Descriptor List to determine whether it is most appropriate to create a conference automatically (actions 1, 3, or 4), or to allow the user to attempt to join an existing conference at the called node (action 2). Therefore, when creating any conference, making it listed and unlocked indicates that it is to be available to be joined by a calling node. If the intent is not to allow a conference to be joined by a caller, the conference shall be created as locked, unlisted, or both.

Table 7-16b defines the set of rules for determining the default action of the calling node (as well as the called node) as a function of the node types and the parameter settings contained in the GCC-Conference-Query confirm.

TABLE 7-16B

Rules for Determining the Default Action of the Called and Calling Nodes

Calling Node Type	**Called Node Type**	**Unlocked Conferences in List**	**Default Conference Flag**	**Wait for Invite Flag**	**No Unlisted Conference Flag**	**Default Action of Calling Node**	**Default Action of Called Node**
Terminal	Terminal	*	*	*	*	Caller invites	Wait for caller
Multiport Terminal	Terminal	*	*	*	*	Caller invites	Wait for caller
MCU	Terminal	*	*	*	*	Caller invites	Wait for caller
Terminal	Multiport Terminal	FALSE	*	*	*	Caller creates remotely	Wait for caller
Terminal	Multiport Terminal	TRUE	NOT PRESENT or FALSE for all conferences	*	NOT PRESENT or FALSE	Caller chooses conference to join	Wait for caller
Terminal	Multiport Terminal	TRUE	NOT PRESENT or FALSE for all conferences	*	TRUE	Caller chooses listed conference to join	Wait for caller
Terminal	Multiport Terminal	TRUE	TRUE for one conference	*	*	Caller joins default conference	Wait for caller
Multiport Terminal	Multiport Terminal	FALSE	*	*	*	Caller invites	Wait for caller
Multiport Terminal	Multiport Terminal	TRUE	NOT PRESENT or FALSE for all conferences	*	NOT PRESENT or FALSE	Caller chooses conference to join	Wait for caller
Multiport Terminal	Multiport Terminal	TRUE	NOT PRESENT or FALSE for all conferences	*	TRUE	Caller chooses listed conference to join	Wait for caller
Multiport Terminal	Multiport Terminal	TRUE	TRUE for one conference	*	*	Caller joins default conference	Wait for caller
MCU	Multiport Terminal	*	*	*	*	Caller invites	Wait for caller
Terminal	MCU	FALSE	*	NOT PRESENT or FALSE	NOT PRESENT or FALSE	Caller chooses unlisted conference to join OR Caller repeats query	Wait for caller
Terminal	MCU	FALSE	*	NOT PRESENT or FALSE	TRUE	Caller repeats query	Wait for caller
Terminal	MCU	TRUE	NOT PRESENT or FALSE for all conferences	NOT PRESENT or FALSE	NOT PRESENT or FALSE	Caller chooses conference to join	Wait for caller
Terminal	MCU	TRUE	NOT PRESENT or FALSE for all conferences	NOT PRESENT or FALSE	TRUE	Caller chooses listed conference to join	Wait for caller
Terminal	MCU	TRUE	TRUE for one conference	NOT PRESENT or FALSE	*	Caller joins default conference	Wait for caller
Terminal	MCU	*	*	TRUE	*	Caller waits for invite from called node	Called node invites
Multiport Terminal	MCU	FALSE	*	NOT PRESENT or FALSE	NOT PRESENT or FALSE	Caller chooses unlisted conference to join OR Caller repeats query	Wait for caller
Multiport Terminal	MCU	FALSE	*	NOT PRESENT or FALSE	TRUE	Caller repeats query	Wait for caller
Multiport Terminal	MCU	TRUE	NOT PRESENT or FALSE for all conferences	NOT PRESENT or FALSE	NOT PRESENT or FALSE	Caller chooses conference to join	Wait for caller
Multiport Terminal	MCU	TRUE	NOT PRESENT or FALSE for all conferences	NOT PRESENT or FALSE	TRUE	Caller chooses listed conference to join	Wait for caller
Multiport Terminal	MCU	TRUE	TRUE for one conference	NOT PRESENT or FALSE	*	Caller joins default conference	Wait for caller

Calling Node Type	**Called Node Type**	**Unlocked Conferences in List**	**Default Conference Flag**	**Wait for Invite Flag**	**No Unlisted Conference Flag**	**Default Action of Calling Node**	**Default Action of Called Node**
Multiport Terminal	MCU	*	*	TRUE	*	Caller waits for invite from called node	Called node invites
MCU	MCU	*	*	*	*	FOR FURTHER STUDY	FOR FURTHER STUDY

...

7.1.4.4 Point-to-Point Conference Establishment

A point-to-point conference is distinct from the other varieties in that it involves only two terminal nodes with no MCU present. In the case where it is known which terminal is the calling terminal (the initiator of the physical connection) and which is the called terminal, a point-to-point conference may be established by the calling terminal first querying the called terminal by issuing a GCC-Conference-Query request. This allows the terminal to determine if the other node is a terminal, MCU, or multiport terminal without requiring *a priori* knowledge to that effect. The GCC-Conference-Query confirm generated in response to this request indicates, in the case of a point-to-point call, that the directly connected node is a user-terminal. Once it is known that the directly connected node is a terminal, the conference is established by the calling terminal by issuing a GCC-Conference-Create request to create a new conference or by creating a conference locally, and inviting the other terminal by issuing a GCC-Conference-Invite request. Typically, the conference would be specified with an arbitrary Conference Name such as "0", and would be ~~un~~locked, unlisted, and automatically terminating.

...

7.1.4.5 Conference Establishment among Multiport Terminals

...

In general, it is typically desirable that a connection of this kind be made automatically, like a point-to-point call rather than like a meet-me conference. Specifically, if neither the called nor calling nodes are already part of conferences connecting them to other nodes, the call should be treated exactly as a point-to-point call. If the calling node already has an ongoing conference, it is typical that this node would simply invite the new node into the existing conference.

If instead the called node has an ongoing conference, the action ~~may~~ depends on whether that conference is locked or unlocked, listed or unlisted~~, or password protected~~. If the conference is unlisted, without *a priori* information, the calling node would not be aware of its presence and would treat the call as if there were no conference present at that node (if it did have *a priori* information that an unlisted conference was present, it could join that conference). If the conference is locked, the calling node has no way to join that conference and again would treat the call as if that conference were not present.~~ If the conference is password protected, again, the calling node would likely treat the call as if there were no conference present at that node, although it does have the option of prompting the user for a password and attempting to join the conference.~~ In either of these cases, once a new conference is established, the called node may choose to transfer the nodes connected to the previous conference into the new conference. If there are one or more listed conferences~~is a single conference~~ at the called node which are not locked,~~is "open" (not locked, unlisted, or password protected)~~ the calling node should assume that the called node is hosting a meet-me conference and~~simply attempt to join this conference.~~ ~~If there is more than one "open" conference at the calling node, this~~ would typically require a user to decide ~~explicitly~~ which one to join - in this case, the procedure would not be entirely automatic.

If there are conferences present at both nodes~~ (assuming an "open" conference at the called node)~~, the calling node would normally either invite the called node into its conference, or would attempt to join the called node's conference. In either case, the multiport terminal with the existing conference that was not enlarged to include the other multiport terminal should transfer all subordinate nodes to the conference that it has newly become joined (or invited) to.

[End Correction]

6.7.5.10 Typographical Error in Summary of GCC Abstract Services

Description: In Table 6-1 of T.124, the summary of GCC primitives and PDUs, an entry for GCC-Conductor-Please response is listed. There is, however, No GCC-Conductor-Please response primitive.

[Begin Correction]

6.8 Summary of GCC Abstract Services

...

TABLE 6-1

GCC Primitives and PDUs

Functional Unit	**Primitives**	**Term**	**MCU**	**Associated PDUs**	**Dir.**	**Term**	**MCU**
...	...	...	...	...	...	...	...
	GCC-Conductor-Please request	O	O	-	-	-	-
	GCC-Conductor-Please indication	O	O	-	-	-	-
	~~GCC-Conductor-Please response~~	~~O~~	~~O~~	-	-	-	-
	GCC-Conductor-Please confirm	O	O	-	-	-	-
...	...	...	...	...	...	...	...

[End Correction]

6.7.6 Technical and Editorial Corrections to ITU-T Recommendation T.125

— None —

6.7.7 Technical and Editorial Corrections to ITU-T Recommendation T.126

The corrections in this section apply to the following Recommendation:

– ITU-T Recommendation T.126 (1995), *Multipoint Still Image and Annotation Protocol*

6.7.7.1 Update Document References

Description: Update references to reflect the current status of reference documents.

[Begin Correction]

2.0 Normative references

The following ITU Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The ITU Secretariat maintains a list of the currently valid ITU Recommendations.

- ITU-T Recommendation F.710 (1988), General Principles for Audiographic Conference Service
- ITU-T Recommendation H.221, *Frame Structure for a 64 to 1920 Kbits/s Channel in AudioVisual Teleservices*
- ITU-T Recommendation T.4 (1993), *Standardization of Group 3 Facsimile Apparatus for Document Transmission*
- ITU-T Recommendation T.6 (1988), *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus*
- ITU-T Recommendation T.35 (1991), *Procedure for the Allocation of CCITT Defined Codes for Non-Standard Facilities*
- ITU-T Recommendation T.42 (1994), *Continuous-Tone Colour Representation Method for Facsimile*
- ITU-T Recommendation T.50 (1992), *International Reference Alphabet - 7-Bit Coded Character Set for Information Interchange*
- ITU-T Recommendation T.81 (1992), *Information Technology - Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines*
- ITU-T Recommendation T.82 (1993), Information Technology - *Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression*
- ITU-T Recommendation T.120 (1996~~x~~), *Data Protocols for Multimedia Conferencing*~~*Introduction to the Audiographics and Audiovisual Conferencing Recommendations* – in development~~
- ITU-T Recommendation T.121 (1996), *Generic Application Template*
- ITU-T Recommendation T.122 (1993), *Multipoint Communication Service for Audiographic and Audiovisual Conferencing Service Definition*
- ITU-T Recommendation T.123 (1993), *Network Specific Data Protocol Stacks for Multimedia Conferencing*~~*Audiographics and Audiovisual Teleconference Applications*~~
- ITU-T Recommendation T.124 (1995~~x~~), *Generic Conference Control*~~*for Audiovisual and Audiographic Terminals and Multipoint Control Units* – in development~~
- ITU-T Recommendation T.125 (1994), *Multipoint Communication Service Protocol Specification*

- ITU-T Recommendation X.680 (1994), *Information Technology - Abstract Syntax Notation One (ASN.1) - Specification of Basic Notation*
- ITU-T Recommendation X.690 (1994), *Information Technology - ASN.1 Encoding Rules - Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)*
- ITU-T Recommendation X.691 (199x), *Information Technology - ASN.1 Encoding Rules - Specification of Packed Encoding Rules (PER)* – in development
- ITU-R BT 601-1:1992, Encoding Parameters of Digital Television for Studios, Volume RBT
- ITU-R BT 709:1990, Basic Parameter Values for the Studio and for International Programme Exchange, Volume XI-1
- ISO/IEC 10646-1 (1993), *Information technology - Universal Multiple-Octet Coded Character Se(USC) - Architecture and Basic Multilingual Plane*
- CIE 1976 (L*a*b*) space, *CIE publication No. 15.2, Colorimetry, 2nd Ed. (1986)*

[End Correction]

6.7.7.2 Replace Inappropriate Uses of the Word Conference with Session

Description: Most occurrences of the whole word "conference" should be replaced by the word "session".

[Begin Correction]

3. Definitions

For the purposes of this Recommendation, the following definitions apply.

...

5. **Application conferencing:** The use of SIPDUs to accomplish simple application conferencing among conferenced SICEs. One SICE shall run an application that is controllable by any SICE in the session~~conference~~.

...

11. **Non-standard capability:** The capability is outside the scope of this Specification but it has been determined through negotiation that it is recognized among all session~~conference~~ participants.

...

— SPECIAL INSTRUCTIONS — REPLACE ALL OCCURRENCES OF THE WHOLE WORD "conference" WITH THE WORD "session" IN CLAUSES 5.1 THROUGH 5.9 INCLUSIVE AND IN CLAUSES 6. THROUGH 8.8 INCLUSIVE

...

9. SIPDU Definitions

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- ConductorPrivilege
--          List of privileges that are awarded by the SICE at the conducting
--          node to other SICEs in the session~~conference~~.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- ArchiveClosePDU
--          The ArchiveClosePDU is used to close an archive that was
--          previously opened during an SI session~~conference~~.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- ArchiveOpenPDU
--        The ArchiveOpenPDU is used to open an archive at a
--        remote terminal that supports this capability.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
ArchiveOpenPDU   ::=   SEQUENCE
{
      archiveHandle                  Handle,
                                           -- Unique handle that is used to reference
                                           -- this archive during the session~~conference~~
      mode                           ArchiveMode,
                                           -- Indicates the access restrictions placed
                                           -- on the archive
      header                         ArchiveHeader,
                                           -- Specifies information used to identify
                                           -- the archive.  If the archive
                                           -- is being created, this is the information
                                           -- that is used to identify the archive in
                                           -- the future.
      maxEntries                     INTEGER (1..65535) OPTIONAL,
                                           --  This parameter allows remote terminals
                                           --  to estimate the local resource usage for
                                           --  the specified archive so they can
                                           --  signal an error early in the archiving
                                           --  process.  It is only to be specified if
                                           --  the archive open mode is set to "create"
```

```
    nonStandardParameters    SET OF NonStandardParameter OPTIONAL,
                                 -- Allowed only if the corresponding
                                 -- non-standard capabilities are present
                                 -- in the negotiated capability set.
    ...
}
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- WorkspaceCreatePDU
--        This PDU causes a workspace to be created and its
--        attributes to be set.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
WorkspaceCreatePDU   ::=   SEQUENCE
{
    workspaceIdentifier              WorkspaceIdentifier,
                                         -- Identifier that will be used to reference
                                         -- this workspace in the future.
    appRosterInstance                INTEGER (0..65535),
                                         -- Indicates which application roster
                                         -- instance (returned in the
                                         -- GCC-Application-Roster-Report
                                         -- indication) was valid when this PDU
                                         -- was issued.  This is used to eliminate
                                         -- race conditions that can occur when
                                         -- terminals enter a session~~conference~~ while
                                         -- a workspace is being created.
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- WorkspaceRefreshStatusPDU
--        This PDU is used by a SICE to announce or remit its status as
--        the session~~conference~~ refresh SICE for SICEs that join late.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
WorkspaceRefreshStatusPDU   ::=   SEQUENCE
{
    refreshStatus            BOOLEAN,
                                 -- TRUE indicates that the SICE sourcing
                                 -- this PDU is functioning as the session~~conference~~
                                 -- wide refresher.
                                 -- FALSE indicates that the SICE sourcing
                                 -- this PDU has ceased to function as the
                                 -- session~~conference~~ wide refresher.
    nonStandardParameters    SET OF NonStandardParameter OPTIONAL,
                                 -- Allowed only if the corresponding
                                 -- non-standard capabilities are present
                                 -- in the negotiated capability set.
    ...
}
```

[End Correction]

6.7.7.3 Use of Uniform Send for Archive Open

Description: If two SICEs attempt to open an archive for writing at nearly the same time, the use of MCS-Send-Data for the ArchiveOpenPDU results in a race condition whereby different SICEs may not agree on which of the opens was successful. To remedy this problem, MCS-Uniform-Send-Data must be used for the ArchiveOpenPDU.

[Begin Correction]

6.2 Use of MCS Data Services

...

TABLE 6-3

Use of MCS Data Primitives for SIPDUs

…	…	…	…
ArchiveOpenPDU	SI-CHANNEL	MCS-Uniform-Send-Data	Low
…	…	…	…

[End Correction]

6.7.7.4 Referencing T.121 for Application Enrolment Procedure

Description: Clauses 8.1 and 8.2.1 through 8.2.5 should be deleted and replaced by a reference to T.121 with appropriate specification of parameters in order to correct differences between the application initialization procedure defined in these sections with the procedures defined in T.121. Similarly, clause 7 should be revised to reference T.121 in describing the use of GCC primitives. Also, updated terminology for session types used in T.121.

[Begin Correction]

5.1 SI Application Enrolment

An SI application enrols via the application enrolment mechanism specified in ~~Draft~~ ITU-T Recommendation T.121~~4~~. ~~A SICE receives an indication from GCC when it may begin the enrollment process. There are two scenarios where this happens. The first is the normal conference start up process, where the local node controller joins a conference. After it locally establishes itself within the conference, a GCC-Application-Permission-To-Enroll indication is issued to all local applications. The second occurs during an established conference when one node decides to invoke a session of a specific application at multiple nodes.~~

6.1 MCS Token and Channel Usage

Table 6-2 describes MCS channel and token usage for ~~default (static), multicast, or private~~ SICE sessions of the types defined in ITU-T Recommendation T.121. In the case of ~~the default SICE~~ a session type requiring static channels and tokens, the Channel and Token IDs shown in Table 6-2 shall be used~~static tokens and channels (except for the MCS-USER-ID channels) shall be used as defined in Table 6-2~~ (symbolic IDs shown). For all other ~~multicast or private SICE~~ session~~s~~ types, the Resource IDs shown in the table shall be used for allocating dynamic tokens and channels. The given Resource IDs shall be encoded as two octet T.50 text strings using the characters shown in quotes in Table 6-2.

TABLE 6-2

Description of SI Tokens and Channels

Mnemonic	**Mnemonic for Static Channel and Token IDs**	**Resource IDs for Dynamic Channels and Tokens~~Multicast or Private Sessions~~**	**Description**
SI-{MCS-USER-ID}-CHANNEL	–	–	Certain SIPDUs are sent directly to individual SICEs. To do this, the individual MCS-USER-ID channels of the peer SICEs in the MCS domain are used.
SI-CHANNEL	SI-CHANNEL-0	"C0"	This channel bears all SIPDUs to be broadcast to all peer SICEs in a domain.
SI-BITMAP-CREATE-TOKEN	SI-TOKEN-0	"T0"	This token is used to restrict bitmap creation to allow only a single bitmap creation at a time. This token shall always be used for bitmaps destined to hard-copy workspaces. For soft-copy workspaces, this token shall only be used if the Soft-Copy-Bitmap-No-Token-Protection capability is not present in the negotiated capability set.
SI-WORKSPACE-REFRESH-TOKEN	SI-TOKEN-1	"T1"	This token is used to allow a specific SICE to become the designated workspace refresher. The SICE that holds this token is responsible for refreshing workspaces when a new peer SICE enrols into the conference.

7. Use of GCC

A SICE uses the services of GCC (ITU-T Recommendation T.124) in the manner specified in ITU-T Recommendation T.121. The use of GCC by a SICE shall comply with the procedures outlined in ITU-T Recommendation T.121 in addition to the procedures explicitly described in this

Recommendation.~~the following GCC service primitives to enroll and exchange capabilities, to allocate unique handles, to determine conductorship mode, and to allocate and register dynamic channels and tokens. Table 7-1 describes each of the primitives used by a SICE.~~

~~Table 7-1 – GCC primitives needed by a SICE~~

~~GCC Primitive~~	**~~Default Session~~**	**~~Multicast or Private Sessions~~**
~~GCC-Application-Permission-To-Enroll~~	~~Indication~~	~~Indication~~
~~GCC-Application-Enroll~~	~~Request~~ ~~Confirm~~	~~Request~~ ~~Confirm~~
~~GCC-Application-Roster-Report~~	~~Indication~~	~~Indication~~
~~GCC-Registry-Retrieve-Entry~~	–	~~Request~~ ~~Confirm~~
~~GCC-Registry-Register-Channel~~	–	~~Request~~ ~~Confirm~~
~~GCC-Registry-Assign-Token~~	–	~~Request~~ ~~Confirm~~
~~GCC-Registry-Allocate-Handle~~	~~Request~~ ~~Confirm~~	~~Request~~ ~~Confirm~~
~~GCC-Conductor-Assign~~	~~Indication~~	~~Indication~~
~~GCC-Conductor-Release~~	~~Indication~~	~~Indication~~
~~GCC-Conductor-Permission-Grant~~	~~Indication~~	~~Indication~~

7.1 GCC Unique Handles

All handles used in the SI protocol are acquired from GCC using the GCC-Registry-Allocate-Handle primitive. Handles can be allocated at any time not just immediately prior to their use. It is suggested that applications allocate blocks of handles to minimize network traffic and perform this operation when idle to avoid associated latencies during periods of protocol activity.

8. Protocol Specification

--- SPECIAL INSTRUCTIONS --- DELETE CLAUSES 8.1, 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5

8.1 Session Initialization and Management

Session initialization and management for T.126 shall be performed according to ITU-T Recommendation T.121. The functions of the Application Resource Manager (ARM) described in T.121 shall apply to any T.126 session. For T.126, the following initialization parameters shall be used. For the Application Protocol Key, the value {itu-t recommendation t 126 version(0) 1} shall be used. The required channel and token resources are defined in Table 6-2. The numeric values of the static channel and token IDs are specified in ITU-T Recommendation T.120.

~~8.2 SI Application Enrollment and Initialisation~~

8.2~~8.2.6~~ SI Capabilities

~~SICEs use the application enrollment mechanism for capabilities negotiation. The Application Capabilities List parameter of the GCC-Application-Enroll request PDU is used to specify the list of capabilities supported by the local SICE.~~

Capabilities exchange and negotiation shall be performed according to ITU-T Recommendation T.121. The SI capabilities that can be advertised and negotiated are described in Table 8-5. ~~The Application Capabilities List included in the GCC-Application-Enroll request is composed of list capabilities to be advertised as supported by this SICE.~~ ~~Note that a~~At least one of the following capabilities shall be included in the~~is~~ advertised Application Capabilities L~~l~~ist: Hard-Copy-Image~~Workspace~~ or Soft-Copy-Workspace. If Soft-Copy-Workspace is included, then at least one of the following shall also be included: Soft-Copy-Image or Soft-Copy-Annotation.

In Table 8-5 certain capabilities are indicated as being dependent on other capabilities. This implies that the capability shall not be included in the Application Capabilities List unless the capability on which it depends is also included.

SICEs are made aware of the Application Capabilities that are valid for a given session via GCC-Application-Roster-Report indications. The conditions under which this event is generated are described in ITU-T Recommendation T.121. A SICE may be required to process GCC-Application-Roster-Report indications multiple times during a session and shall adhere to the bounds imposed by the capabilities reported in this fashion. Many components of this Recommendation heavily depend on this mechanism for proper operation. Clauses within this Recommendation describing such components include rules governing the interpretation of the applicable parameters conveyed by the GCC-Application-Roster-Report indication.

~~The result of the capabilities negotiation procedure is made apparent to the SICE by the receipt of a GCC-Application-Roster-Report indication from the GCC provider. The application roster report includes the Application Roster for peer SICEs within the indicated conference — that is, SICEs which have designated the same Session Key. The Application Roster includes a list of nodes for which a peer SICE has enrolled. For each node, the list contains the GCC User ID of that node, and the Application User ID of the peer SICE at that node. The Application Roster also includes an instance number, a flag indicating whether new nodes have been added since the last instance, a flag indicating whether nodes have been removed since the last instance, a flag indicating whether the Application Capabilities List has been updated since the last instance, and if so, the new Application Capabilities List. In the case of a newly enrolling SICE, the Application Capabilities List is always updated since previous instances of the list are not available to this SICE.~~

~~When first enrolling, a SICE shall ignore received GCC-Application-Roster-Report indications in which the SICE is not included (if there is no entry in the application roster which has the GCC User ID of the local node and the Application Protocol Entity ID of the local SICE as indicated in the GCC-Application-Enroll confirm). Once a roster is received which does include the local SICE, that SICE is now considered part of the conference and may proceed examining the roster to determine how to proceed.~~

~~The Application Capabilities List received as part of the GCC-Application-Roster-Report indication corresponds to the collapsed Application Capabilities Lists of all enrolled peer SICEs. That is, the list includes an entry for each capability which has been issued by any peer SICE. For each entry, it includes the Capability ID, the number of peer SICEs (including the local one) which had advertised~~

~~this capability as part of their enroll procedure, and, in the case of capabilities of the MIN or MAX class, the minimum or maximum value of the parameter among all peer SICEs which advertised this capability. For each capability, the rules used to determine the result of the capability exchange is indicated in the Node Count Rule column of Table 8-5. The notation in this column explaining how these rules are to be applied is described in Table 8-6.~~

~~At any time while a SICE is enrolled in a conference, the SICE may receive additional GCC-Application-Roster-Report indications from the GCC provider indicating that the contents of the roster have changed. This may be due to new peer SICEs enrolling in the conference, peer SICEs leaving the conference, or peer SICEs having modified their enrollment information.~~

~~If, at any time, the local SICE desires to indicate a change in its Application Capabilities List, it may re-enroll. This is done by re-issuing a GCC-Application-Enroll request to the GCC provider with the Enroll/Un-enroll flag set to Enroll and the updated Application Capabilities List, as well as the other parameters normally included in the enroll request. The result of this is potentially a change to the Application Roster, resulting in receipt of a GCC-Application-Roster-Report indication to the local SICE as well as all other peer SICEs in the conference.~~

~~If, at any time, the local SICE desires to terminate its participation in the conference, it may un-enroll by issuing a GCC-Application-Enroll request to the GCC provider with the Enroll/Un-enroll flag set to Un-enroll. In this case, no other parameters need be included.~~

~~If at any time the SICE receives a GCC-Application-Roster-Report indication in which it is no longer included (if there is no entry in the application roster which has the GCC User ID of the local node and the Application Protocol Entity ID of the local SICE as indicated in the GCC-Application-Enroll confirm), the SICE shall immediately issue an MCS-Detach-User request to the MCS provider to detach the connection to the specified conference. At this point, the SICE is no longer considered enrolled to the conference. The SICE may re-issue a GCC-Application-Enroll to attempt to become re-enrolled in the conference.~~

~~If at any time the SICE receives a GCC-Application-Permission-To-Enroll indication with the Grant/Revoke flag set to Revoke, if the SICE has already attached to MCS, it shall immediately issue an MCS-Detach-User request to the MCS provider to detach the connection to the specified conference. At this point, the SICE is no longer considered enrolled to the conference and shall not attempt to re-enroll.~~

[End Correction]

6.7.7.5 Missing Capability Table Entry for Soft-Copy-Drawing-Rotation

Description: Drawing element rotation was intended to be a negotiated capability. A capability was omitted from the table of SI capabilities in the document which erroneously implies that this function is mandatory for baseline implementations. The text is ambiguous as well.

[Begin Correction]

TABLE 8-5

Application Capabilities List Elements

Capability Name (Default Value if Numeric Class): **Description**	**ID**	**Class**	**SICE Count Rule**	**Parameter**	**Dependency**
...	**...**	**...**	**...**	**...**	**...**
Soft-Copy-Annotation-Drawing-Highlight: This capability is used to negotiate the ability to make use of the Highlight line style for drawing.	28	L	=D	–	Soft-Copy-Annotation
Soft-Copy-Annotation-Drawing-Rotation: This capability is used to negotiate the ability to specify the optional rotation parameter that defines a rotation to be applied to annotation drawing elements.	62	L	=D	–	Soft-Copy-Annotation
Soft-Copy-Annotation-Bitmap-Format-T.82: This capability is used to negotiate the ability to use T.82 (JBIG) compression format for encoding annotation bitmaps. This capability implies the ability to handle either 8-bit greyscale, or up to 8 palletized~~palettized~~ bitplanes with a 1:1 pixel aspect ratio and the ability to only handle bitmaps encoded without the use of JBIG resolution reduction.	29	L	=D	–	Soft-Copy-Annotation
...	**...**	**...**	**...**	**...**	**...**

...

8.6.2.2 Rotation

This is an optional parameter which determines the rotation to be applied to the specified drawing element. Use of this parameter is conditional upon successful negotiation of the Soft-Copy-Annotation-Drawing-Rotation capability. The parameter~~is~~ consists of two components: the angle of rotation, and the axis of rotation. The angle of rotation is specified in integer units of minutes of arc within the range (0..21599). The direction of rotation is defined to be counterclockwise. The rotation is performed relative to a specified axis of rotation. This axis is specified as a relative offset from the anchor point. Its coordinates are bounded to the range (-32768..32767).

[End Correction]

6.7.7.6 Interpretation of Workspace Plane Coordinate System

Description: Clarification of the interpretation of negative valued plane coordinates.

[Begin Correction]

8.3.1.2 Plane Coordinate System

All planes on a workspace are of the same size with the origin of each co-located. Within each plane, pixels are indexed from the origin (0,0) which is defined to be the upper left corner of the workspace, to the size of the workspace (X-1, Y-1) which is defined to be the lower right corner of the workspace, where X and Y are the number of pixels in the workspace, in the horizontal and vertical dimensions, respectively, as specified when the workspace was created. The sizes X and Y shall be greater than or equal to one and less than or equal to the negotiated maximum values of Soft-Copy-Workspace-Max-Width and Soft-Copy-Workspace-Max-Height, respectively.

Positional references to a workspace plane are designated by specifying a Point. A point is an ordered pair of Workspace Coordinates specifying the horizontal and vertical position in the workspace, respectively. The value of a Workspace Coordinate is defined to be within the range -21845, 43690. The use of negative values allows objects (e.g. the origin of a bitmap, or a control point in a drawing element) to be positioned ~~the ability to reference certain objects (e.g. the origin of a bitmap, or a control point in a drawing element) relative to a point~~ above or to the left of the origin of the workspace plane.

[End Correction]

6.7.7.7 Remove Transparent Background Color Restriction

Description: The restriction excluding transparent as a valid workspace background color should be removed. This represents no change to the protocol format given that the PDU definitions can currently support the specification of a background color.

[Begin Correction]

8.3.1.5.4 Background Color

...

~~For any of the palettes which may be used, Transparent shall not be specified as the background color. If received, the color White shall be used instead.~~

[End Correction]

6.7.7.8 Protected Plane Access List

Description: The current mechanism of specifying planes as accessible only by the creator of a workspace has a defect when used in a conference where a node is designated as the refresher for late joiners. Refreshed workspaces that were originally created with one or more planes marked creator protected cannot be recreated by the refresher with the same attributes. This is because the creator of the refreshed workspace is not the original creator. To correct this defect, a protected plane access list need be added to the workspace create transaction to allow the specification of one or more SICEs that have privileges to modify protected planes. The refresher could then specify itself as well as the original privileged SICEs in order to accurately restore the workspace state.

[Begin Correction]

6.2 Use of MCS Data Services

...

TABLE 6-3

Use of MCS Data Primitives for SIPDUs

SIPDU	**Channel**	**MCS Data Primitive**		**Priority**
		Synchronized Workspaces	**Unsynchronized Workspaces** **(or Hardcopy exchanges for valid SIPDUs)** **or** **Exchanges Involving Single SICE ~~Only~~ Access~~Creator~~ Protected Workspace Planes*** **	
…	…	…		…

* This indicates that the priority shall be Medium if the destination plane has the annotation usage designator set or in the case of bitmap operations, if the bitmap destination is the pointer plane, and Low otherwise.

** The use of the SINonStandardPDU is beyond the scope of this Recommendation.

***~~The SICE that created the workspace shall not use~~ MCS-Send-Data shall not be used unless the protectedPlaneAccessList includes only one SICE, the destination plane of the operation is designated protected and the SICE ~~it~~ guarantees to never change the plane protection of the ~~a~~ plane to unprotected.

...

8.3.1.6 Workspace Plane Parameters

When a workspace is created, each plane is defined to have a number of characteristics: whether or not the plane is editable, the intended usage of the plane, and whether the plane is protected for exclusive use by the list of SICEs specified in the protectedPlaneAccessList ~~creator~~. The later ~~two~~ of these characteristics is an~~are~~ attribute~~s~~ of the workspace plane which may be modified after the workspace has been created.

TABLE 8-9

Plane Attributes

Attribute	**Default Value**	**Description**
protection	p~~creatorP~~rotected = FALSE	This flag indicates ~~any~~ access restrictions for a plane. The p~~creatorP~~rotected sub-parameter, if set to TRUE, restricts all access to the plane to be by the SICEs specified in the protectedPlaneAccessList~~creator~~ of the workspace only. A value of FALSE imposes no restrictions.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation.

...

8.3.1.6.3 Plane Protection

Each plane in a workspace may be designated as either unprotected or ~~creator-~~protected. This designation is specified when the workspace is created as part of the plane-attributes parameter in the WorkspaceCreatePDU. Its value may also be modified for an existing workspace by using the WorkspaceEditPDU. Only SICEs specified in the protectedPlaneAccessList ~~the creator of the workspace~~ may edit this attribute using the WorkspaceEditPDU. If a WorkspaceEditPDU is received from any node other than the SICEs specified in the protectedPlaneAccessList ~~node which created the specified workspace~~ which indicates a change in the value of this attribute, this change shall be ignored (other attribute changes shall still be processed as normal).

If a workspace plane is designated as unprotected, any SICE in a conference may direct information (bitmaps or drawing elements) toward the plane. Protected workspace planes automatically revert to unprotected when all the SICEs identified by the protectedPlaneAccessList for that workspace leave the session or become inactive as indicated by a GCC-Application-Roster-Report indication. Workspace planes shall not be designated protected for archive workspaces. All parameters associated with plane protection shall be ignored in the case of archive workspaces.

If a workspace plane is designated as ~~creator-~~protected, only the SICEs specified in the protectedPlaneAccessList ~~SICE which issued the WorkspaceCreatePDU to create this workspace~~ may direct information toward the plane. Moreover, if the protectedPlaneAccessList contains only a single SICE that~~the creating~~ SICE can choose to use MCS-Send-Data instead of MCS-Uniform-Send-Data for transactions directed to a workspace plane when the destination workspace plane is synchronized, if that~~the creating~~ SICE never intends to unprotect the target plane.

...

8.3.4 Editing Workspace, Plane, and View Attributes

...

If the plane protection attribute is among those in the list of attributes to be modified, the receiving SICE shall check that the User ID of the source of the WorkspaceEditPDU is that same as that of any SICE listed in the protectedPlaneAccessList included in the WorkspaceCreatePDU used to create the workspace~~that User ID of the SICE which created the workspace~~. If so, this attribute shall be modified as indicated in the PDU. If not, this attribute shall not be modified.

8.3.5 Copying Workspace Contents

...

The WorkspacePlaneCopyPDU is allowed from all SICEs if the specified destination plane is designated unprotected. If the designated plane is ~~creator~~ protected, the WorkspacePlaneCopyPDU may~~shall~~ only be transmitted by the SICEs listed in the protectedPlaneAccessList ~~creator~~ of the workspace. If a SICE receives a WorkspacePlaneCopyPDU with a destination plane specified which is protected from a SICE which is not allowed to modify this plane, the SICE shall ignore the received SIPDU.

The WorkspacePlaneCopyPDU is allowed only between planes which have identical usage designators and identical editability flags (either both editable or both permanent). If a SICE receives a WorkspacePlaneCopyPDU specifying a source and destination plane which do not meet these qualifications, the SICE shall ignore the received SIPDU.

...

9. SIPDU Definitions

...

```
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
-- PlaneProtection
--          This enumeration identifies the possible access restrictions
--          that can be imposed on a workspace plane.
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
PlaneProtection   ::=   SEQUENCE
{
      p~~creatorP~~rotected                    BOOLEAN,
                                                  -- Only the SICEs granted access via the
                                                  -- protectedPlaneAccessList can submit~~workspace~~
                                                  -- ~~creator can submit ~~data to thise plane
      ...
}
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- WorkspaceCreatePDU
--        This PDU causes a workspace to be created and its
--        attributes to be set.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
WorkspaceCreatePDU   ::=   SEQUENCE
{
      workspaceIdentifier              WorkspaceIdentifier,
                                              -- Identifier that will be used to reference
                                              -- this workspace in the future.
      appRosterInstance                INTEGER (0..65535),
                                              -- Indicates which application roster
                                              -- instance (returned in the
                                              -- GCC-Application-Roster-Report
                                              -- indication) was valid when this PDU
                                              -- was issued. This is used to eliminate
                                              -- race conditions that can occur when
                                              -- terminals enter a conference while
                                              -- a workspace is being created.
      synchronized                     BOOLEAN,
                                              -- TRUE specifies that the
                                              -- workspace contents
                                              -- stacking order must be consistent
                                              -- everywhere.
                                              -- In many cases, this implies
                                              -- the use of
                                              -- MCS-UNIFORM-SEND-DATA for SIPDU
                                              -- submission.
                                              -- FALSE specifies that the
                                              -- workspace contents
                                              -- do not have to be consistent
                                              -- in stacking
                                              -- order therefore the use of
                                              -- MCS-SEND-DATA is
                                              -- acceptable for all content
                                              -- submitting transactions
      acceptKeyboardEvents             BOOLEAN,
                                              -- If TRUE this workspace can accept
                                              -- remote keyboard events.
      acceptPointingDeviceEvents       BOOLEAN,
                                              -- If TRUE this workspace can accept
                                              -- remote pointer device events.
      protectedPlaneAccessList         SET (SIZE (1..65536)) OF MCSUserID OPTIONAL,
                                              -- The ability to modify any protected plane
                                              -- in this workspace is restricted only to
                                              -- SICEs on this list.  The creator of the
                                              -- workspace is NOT automatically granted
                                              -- access to these planes unless explicitly
                                              -- on this list.
      workspaceSize                    WorkspaceSize,
                                              -- This value specifies the width and
                                              -- height of the new workspace in pixels.
      workspaceAttributes              SET OF WorkspaceAttribute OPTIONAL,
                                              -- Editable attributes of the workspace
```

```
    planeParameters                 SEQUENCE (SIZE (1..256)) OF SEQUENCE
                                            -- This sequence contains
                                            -- plane parameters.
                                            -- Its length is the number of planes
                                            -- in the workspace.
    {
        editable                    BOOLEAN,
                                            -- This item specifies whether objects
                                            -- created on this plane are editable
                                            -- If not editable, each plane is treated
                                            -- as a bit-map image.
        usage                       SET (SIZE (1..MAX)) OF PlaneUsage,
                                            -- This item specifies restrictions
                                            -- on the usage of this plane
                                            -- (image data or annotation data).
                                            -- At least one use shall be included.
                                            -- A particular usage designator shall be
                                            -- listed no more than once.
        planeAttributes             SET OF PlaneAttribute OPTIONAL,
                                            -- List of attributes
                                            -- A particular attribute shall be
                                            -- listed no more than once.
        ...
    },

    viewParameters                  SET (SIZE (1..256)) OF SEQUENCE
                                            -- Each entry in this list (if any) defines
                                            -- a view to be created in association with
                                            -- this workspace.
    {
        viewHandle                          Handle,
                                                    -- Identifier of the view to be created
        viewAttributes                      SET OF WorkspaceViewAttribute OPTIONAL,
                                                    -- Editable attributes of the view
        ...
    } OPTIONAL,

    nonStandardParameters           SET OF NonStandardParameter OPTIONAL,
                                            -- Allowed only if the corresponding
                                            -- non-standard capabilities are present
                                            -- in the negotiated capability set.
    ...
}
```

[End Correction]

6.7.7.9 Clarification of View Range

Description: Workspace views should be allowed to be sized such that the view region can encompass areas that lie outside of the workspace. The contents of the view that corresponds to these undefined areas shall be left unspecified.

[Begin Correction]

8.3.1.7 Workspace View Parameters

When a workspace is created, it may be created along with one or more workspace views. Each view has a several editable attributes. These attributes are listed in Table 8-10.

TABLE 8-10

View Attributes

Attribute	**Default Value**	**Description**
viewRegion	fullWorkspace	This attribute defines the portion of the workspace to be associated with this view. This may be set to either fullWorkspace, which indicates that the view corresponds to the complete workspace, or partialWorkspace. In the later case, a rectangular region is specified, in workspace coordinates, indicating the desired portion of the workspace to view. If t~~T~~he viewRegion ~~shall not~~ extends beyond the boundary of the workspace in any dimension the displayed contents of those portions of the view that extend beyond the workspace boundary are not specified by this Recommendation.
viewState	focus	This attribute indicates the initial state of the view, either hidden, background, foreground, focus, or nonStandardState.
UpdatesEnabled	TRUE	This attribute specifies whether or not it is recommended that updates to the workspace associated with this view are displayed, or if the content of the view is instead fixed until updates are enabled. This attribute is to be interpreted as a recommendation on how to display the view rather than a requirement. This attribute applies independently from independent SICEs such that all SICEs disabling updates must re-enable updates for updates to be re-enabled for this view.
SourceDisplayIndicator	–	This attribute specifies the size and position of the view relative to the display at the sourcing terminal. This is to be interpreted as a recommendation to the receiving SICE to display the view in a similar manner. However, there is no requirement that the receiving SICE actually display the view in the indicated manner.
NonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation.

8.3.1.7.1 View Region

Each workspace view is defined to cover a rectangular region of the workspace to which it corresponds. The viewRegion parameter defines the size and location of this region. Its default value is to view the entire workspace. Alternatively, it may be set to view a particular sub-region of the workspace. If t~~T~~he viewRegion ~~shall not~~ extends beyond the boundary of the corresponding workspace in any dimension the displayed contents of those portions of the view that extend beyond the workspace boundary are not specified by this Recommendation.

[End Correction]

6.7.7.10 Change to Mandatory Interpretation of Focus View Visibility

Description: It has been determined that the language in the text regarding focus view visibility is too strong and should be changed as described below.

[Begin Correction]

5.3 Workspaces

...

To make a workspace visible, one or more workspace views may be defined for a workspace. Each view corresponds to a rectangular region within the workspace to be viewed (which, in the simplest case, is the entire workspace). Within an SI session, only one view among all workspaces is designated as the focus view. This view should~~is required to~~ be made visible at all nodes.

...

8.3.1.7.2 View State

When a workspace view is created, the state of the view may be specified as having one of the following values:

- Hidden – Indicates this view should not be shown to the user.
- Background – Indicates that the display of this view is optional.
- Foreground – Indicates that the display of this view is desirable.
- Focus – Indicates that ~~the display of~~ this view is the focus of the current session~~mandatory~~.

[End Correction]

6.7.7.11 Typographical Error In The Discussion of the SI View State

Description: Typographical error correction required.

[Begin Correction]

8.3.1.7.2 View State

...

A SICE may choose to send PDUs transmitted to workspaces in whose views are in different states at different throughput rates. For example, a SICE may send PDUs to workspaces~~ which do~~ whose views are Background or Hidden at a lower rate than PDUs to workspaces with Foreground views or the workspace with a view in the Focus state. This would minimize interference on information bound for the more important workspaces from information being sent in the background. The mechanism for controlling the throughput of PDUs intended for different workspaces is a local matter not covered by this Recommendation.

[End Correction]

6.7.7.12 Effects of Changes to the Application Roster on Workspace Creation

Description: A race condition between the workspace create and the GCC-Application-Roster-Report transactions may result in WorkspaceCreatePDUs to be received with an instance number that is greater than the most recently received GCC-Application-Roster-Report indication. Similar ambiguities exist when first joining a session.

[Begin Correction]

8.3.2 Creating a Workspace

...

Next, the SICE shall determine how it should process the WorkspaceCreatePDU. First, the SICE shall compare the Application Roster instance number specified in the WorkspaceCreatePDU against the last Application Roster instance number that was received that signalled new SICEs entering the session. If the new Application Roster instance is less than the this number, the SICE ~~if it has received the Application Roster with the instance number associated with the Application Roster Instance parameter in the SIPDU. If it has not, it~~ shall ignore the WorkspaceCreatePDU (and all WorkspaceCreatePDUs with the same Workspace Handle received later on other priority channels). If the ~~designated~~ Application Roster instance number specified in the WorkspaceCreatePDU is greater than or equal to the last Application Roster instance number that was received that signalled new SICEs entering the session and less than or equal to the last Application Roster instance number received ~~had been received (even if a later instance has subsequently been received)~~, the WorkspaceCreatePDU is accepted and processed, creating the workspace with the characteristics designated by~~in~~ the parameters in the SIPDU. Because race conditions exist between the receipt of an GCC-Application-Roster-Report indications and the receipt of the WorkspaceCreatePDU, it is possible for the Application Roster instance number specified in the WorkspaceCreatePDU to be greater than the latest instance number reported by an

GCC-Application-Roster-Report indication. In this case, the SICE receiving the WorkspaceCreatePDU shall process in advance or buffer the workspace create transaction and any other valid transactions directed to the workspace until an GCC-Application-Roster-Report indication is received that identifies an instance number which is equal to the one specified in the workspace create. The SICE shall not source any exchanges to this workspace (except acknowledgements) until this time because the correct capabilities set that binds the new workspace is not know until the referenced GCC-Application-Roster-Report indication is received. If the SICE chooses to process any workspace transactions in advance of the receipt of the proper GCC-Application-Roster-Report indication it must not assume that the capabilities associated with the new workspace are within the bounds defined by its own capability set given that the roster change prompting the new application roster may have been triggered by the SICE being either forcibly or voluntarily removed from the session.

A SICE that receives a WorkspaceCreatePDU before it receives its first GCC-Application-Roster-Report indication shall process in advance or buffer the workspace create transaction and any other valid transactions directed to the workspace until a GCC-Application-Roster-Report indication is received that identifies an instance number which is equal to the one specified in the workspace create. The same care must be taken with respect to the temporary capability ambiguity as described above. When the Application Roster does arrive, any workspaces whose Application Roster instance number is less than the instance number reported by the GCC-Application-Roster-Report indication shall be locally discarded.

[End Correction]

6.7.7.13 Determination of Order for Workspace Attribute Changes

Description: There are cases in which the receipt of PDUs transmitted on each of three priority channels can interleave with those sent from other SICEs such that the relative order of reception is different on different priority channels. A method for resolving the order of application of focus changes in conjunction with workspace creation is already specified in the text. This algorithm is not applicable to other cases not already mentioned. These new cases, involving the WorkspaceEditPDU must be added and the mechanism for resolution changed so that it is applicable in all cases.

[Begin Correction]

8.3.2 Creating a Workspace

...

If more than one workspace is being created at the same time by different SICEs, it is possible that the three WorkspaceCreatePDUs from one~~the first~~ SICE will be interleaved with the three from other SICEs~~the second~~ (i.e. the order that they are received on each of the three priority channels may be different). If multiple~~both of the~~ WorkspaceCreatePDUs specify views with Focus set as the viewState, assertion of focus is applied in the order that the WorkspaceCreatePDUs are received on the high priority channel.~~since the order of receipt is indeterminate (normally the viewState of the view in the workspace first created would be changed to the Foreground state when the second workspace is created with a view in the Focus state), the value of the Workspace Handle is used to~~

~~determine which view will remain in the Focus state, and which will be changed to the Foreground state. The view corresponding to the WorkspaceCreatePDU with the highest numbered Workspace Handle (interpreted as an unsigned integer) shall remain in the Focus state.~~

...

8.3.4 Editing Workspace, Plane and View Attributes

...

On receipt of a WorkspaceEditPDU, a SICE shall examine the Workspace Handle and determine whether it has a copy of this workspace. If so, the SICE shall apply the new workspace, plane, and view attributes indicated in the received WorkspaceEditPDU to the local copy of the workspace.

If more than one workspace is being edited or created at the same time by different SICEs, it is possible that the three WorkspaceEditPDUs or WorkspaceCreatePDUs from one SICE will be interleaved with the three from the other SICEs (i.e. the order that they are received on each of the three priority channels may be different). If multiple WorkspaceEditPDUs or WorkspaceCreatePDUs specify views with Focus set as the viewState, assertion of focus is applied in the order that the WorkspaceEditPDUs and WorkspaceCreatePDUs are received on the high priority channel. Similarly, if WorkspaceEditPDUs modifying workspace, plane, or view attributes for the same workspace are received interleaved on the three priority channels, attribute changes are applied in the order that the WorkspaceEditPDUs are received on the high priority channel.

[End Correction]

6.7.7.14 Using WorkspacePlaneCopyPDU for Editable Planes

Description: There currently exists no way to specify the handles that new objects should be mapped to when copying objects between editable planes. This precludes any subsequent unambiguous access to objects created during the copy process. To correct this defect, a separate set of source and destination copy descriptors need be introduced into the WorkspacePlaneCopy PDU. The copy descriptor used to specify the source data and destination for non-editable planes contains parameters equivalent to those found in the current specification. This is acceptable given that no handle based referencing of plane resident objects is allowed for non-editable planes once the object creation process is complete.

Editable planes allow handle based referencing of objects for editing and deletion operations performed after the objects are created. The copy descriptor for editable planes must therefore include a set of handles (one per object to be copied) to be used in subsequent operations that shall be used to reference the object copies to be created. This change also obviates the need for the bounding rectangle parameter in several PDUs given that objects to be copied are directly enumerated.

Additionally, inappropriate uses of the term "note" were removed.

[Begin Correction]

8.3.5 Copying Workspace Contents

If a SICE wishes to copy a rectangular region of one plane of a workspace to another rectangular region of the same plane, a different plane in the same workspace, or a different workspace, it may do so by sending a WorkspacePlaneCopyPDU to all peer SICEs in the manner indicated in Table 6-3. The content of the WorkspacePlaneCopyPDU is shown in Table 8-21.

TABLE 8-21

WorkspacePlaneCopyPDU

Parameter	**Description**
sourceWorkspaceIdentifier	This specifies the source workspace from which a portion of a plane is to be copied. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifer in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall not be included.
sourcePlane	This parameter indicates the plane from which the source rectangle is to be copied. Its value may span the range (0..N-1) where N is the number of planes in the source workspace.
~~**SourceRegion**~~	~~This parameter is a pair of points which define the rectangular region of the source plane from which information is to be copied. Each component of this parameter may span the range (-21845..43690). In the case of a permanent plane, the source information is the set of pixel values within the source rectangle. In the case of an editable plane, the source information is the set of drawing or bitmap objects which lie within the rectangle. If no boundingRectangle had been defined for the object, the object is considered within the source region if, in the case of a drawing element, all control points lie within the source region, and in the case of a bitmap, if the bitmap is fully contained within the source region. If the boundingRectangle parameter had been specified for the object, the object is considered within the source region if the bounding rectangle lies completely within the source region.~~
DestinationWorkspaceIdentifier	This specifies the destination workspace to which a portion of a plane is to be copied. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifer in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall correspond to the current time rather than the time of creation.

destinationPlane	This parameter indicates the plane to which the source rectangle is to be copied. Its value may span the range (0..N-1) where N is the number of planes in the destination workspace. The destination plane shall have the same editability flag setting and the same usage designator setting as the source plane.
~~**DestinationRegion**~~	~~This parameter is a destination rectangle to specify the location that the region is copied in the destination plane. If the Soft-Copy-Scaling capability is not present in the negotiated capability set, the size of this rectangle shall exactly equal the size of the source rectangle. Otherwise, the source information is scaled by the ratio of the destination dimensions to the source dimensions prior to placing the information in the destination workspace. For a permanent workspace, all of the pixels in the destination region are overwritten by values from the source region. For an editable plane, the objects from the source region are overlaid onto the destination plane with a z-order above any existing objects.~~
copyDescriptor	This parameter describes the source and destination data within the designated source and destination planes. If the source and destination planes are permanent, this parameter indicates a rectangular region in the source plane to be copied to the destination (see Table 8-21a). If the source and destination planes are editable, this parameter indicates a list of objects in the source plane to be copied and a corresponding set of new object handles to be used for the newly created objects in the destination plane (see Table 8-21b). Note that in the case of editable planes, scaling of the copied objects as part of the copy operation is not possible. One method of accomplishing the equivalent function is to follow the WorkspacePlaneCopy operation with object edits to adjust parameters controlling object size and position.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-21A

Permanent Plane Copy Descriptor

Parameter	**Description**
sourceRegion	This parameter is a pair of points which define the rectangular region of the source plane from which information is to be copied. Each component of this parameter may span the range (-21845..43690). The source information is the set of pixel values within the source rectangle.
destinationRegion	This parameter is a destination rectangle used to specify the location that the region is copied in the destination plane. If the Soft-Copy-Scaling capability is not present in the negotiated capability set, the size of this rectangle shall exactly equal the size of the source rectangle. Otherwise, the source information is scaled by the ratio of the destination dimensions to the source dimensions prior to placing the information in the destination workspace. All of the pixels in the destination region are overwritten by values from the source region.

TABLE 8-21B

Editable Plane Copy Descriptor

Parameter	**Description**
objectList	This parameter indicates a list of objects in the source plane to be copied and a corresponding set of new object handles to be used for the newly created objects in the destination plane. The objects from the source region are overlaid onto the destination plane with a z order above any existing objects. The copied objects maintain the same relative z order relationship as in the source plane.
destinationOffset	This parameter defines an offset to be added to the coordinates of all of the copied objects. If not present, a zero offset is assumed.
planeClearFlag	When FALSE, the destination objects are appended to the existing set of objects in the destination plane. When TRUE, all existing objects in the destination plane are deleted prior to the copy operation.

...

8.4.1 Creating Bitmaps

...

TABLE 8-23

BitmapCreatePDU Parameters

Parameter	**Description**
…	…

bitmapSize	This parameter specifies the horizontal and vertical size of the bitmap in pixels. Note that the pixel aspect ratio of the bitmap may not be square although the workspace coordinate system assumes a square pixel reference grid. In this case, the number of pixels the bitmap spans in the workspace will be different from the number of pixels in the bitmap itself. In the case of a bitmap format which includes more than one color component, this parameter represents the size of the largest component. The bitmapSize parameter includes a width and height sub-parameter. The allowable range of these parameters is dependent on the destinationAddress selected:
	Width: destinationAddress = hardCopyDevice: (1..Hard-Copy-Image-Bitmap-Max-Width)
	destinationAddress = softCopyImagePlane: (1..Soft-Copy-Image-Bitmap-Max-Width)
	destinationAddress = softCopyAnnotationPlane: (1..Soft-Copy-Annotation-Bitmap-Max-Width)
	destinationAddress = softCopyPointerPlane: (1..Soft-Copy-Pointing-Bitmap-Max-Width)

	Height: destinationAddress = hardCopyDevice: (1..Hard-Copy-Image-Bitmap-Max-Height) destinationAddress = softCopyImagePlane: (1..Soft-Copy-Image-Bitmap-Max-Height) destinationAddress = softCopyAnnotationPlane: (1..Soft-Copy-Annotation-Bitmap-Max-Height) destinationAddress = softCopyPointerPlane: (1..Soft-Copy-Pointing-Bitmap-Max-Height) The bitmapSize parameter shall reflect the actual dimensions of the bitmap encoded in the bitmapData parameter of this PDU (along with any additional BitmapCreateContinuePDUs). If a bitmap is received for which a bitmap dimension is less than that indicated by this parameter, the bitmap shall be padded to the indicated size. In the case of a hard copy destination, it shall be padded with white, in the case of a soft copy~~softcopy~~ destination, it shall be padded with transparent. If a bitmap is received for which a bitmap dimension is greater that that indicated by this parameter, the bitmap shall be truncated to the indicated size. ~~In any case, the bitmap size indicated by this parameter (if the optional bitmapRegionOfInterest or boundingRectangle parameters are not included) shall determine its size as used in determining whether the bitmap is contained within a source rectangle for the purpose of copying a portion of a workspace plane.~~
bitmapRegionOf Interest (Optional)	This optional parameter selects the sub-region within the associated bitmap that is to be displayed. ~~Note that i~~If the destination is an editable workspace plane, a SICE is required to store the entire transmitted bitmap. The default values for the upper left and lower right offsets if not supplied are (0,0) for the upper left and (bitmap width - 1, bitmap height - 1) for the lower right. If this parameter is used, the anchorPoint and scaling parameters apply to the region of interest rather than to the original bitmap borders.
pixelAspectRatio	Describes the pixel aspect ratio of the bitmap. Different values are allowable depending on the bitmap destination, the bitmap format, and the negotiated capability set. These constraints on pixelAspectRatio are listed in Table 8-5. The range of possible values is shown in Table 8-25.
scaling (Optional)	This optional parameter is only allowed if the Soft-Copy-Scaling is part of the negotiated capability set. This parameter if present indicates the offset from the anchor point, in Workspace coordinates, of the lower right corner of the bitmap within the workspace. This parameter may only be used for soft copy~~softcopy~~ bitmaps (it will be ignored for hardcopy bitmaps). If this parameter is not present for a soft copy~~softcopy~~ bitmap the lower right hand corner is determined from the bitmapSize, bitmapRegionOfInterest (if present), and the pixelAspectRatio.

~~**boundingRectangle (Optional)**~~	~~This is a parameter which defines the extent of the bitmap relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this bitmap is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this bitmap shall be included in the region copied regardless of the actual region spanned by the bitmap. If this parameter is not specified, the actual extent of the bitmap (after any scaling) determines whether the bitmap is enclosed by the copy region. This parameter is mandatory if a non-one-to-one pixel aspect ratio is specified for the bitmap on an editable workspace plane. This is to avoid ambiguities in determining the extent of the bitmap with respect to workspace coordinates.~~
checkpoints (Optional)	This optional parameter, when present, specifies that checkpointing is in effect for this exchange and correspondingly specifies the set of Token IDs that are to be used for checkpointing. Each Token ID is used to track the status of a portion of the bitmap in transit at all the receivers by the sender. If this parameter is present, receivers should immediately inhibit all the tokens in this set upon receipt of this PDU.
bitmapFormatHeader	Specifies the algorithm used to encode the bitmap data and associated parameters. ~~Note that c~~Certain values are applicable only to certain bitmap destinations. ~~Note also that s~~Some of the formats listed shall not be used unless the corresponding capability has been negotiated. See Table 8-5 for required capabilities. See clause 8.4.4 for the details of each specific encoding format supported by this Recommendation. If destinationAddress = hardCopyDevice Choice of: 1. bitmapHeaderUncompressed 2. bitmapHeaderT4 3. bitmapHeaderT6 4. bitmapHeaderT82 5. bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list)
	If destinationAddress = softCopyImagePlane Choice of: 1. bitmapHeaderUncompressed 2. bitmapHeaderT81 3. bitmapHeaderT82 4. bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list) If destinationAddress = softCopyAnnotationPlane6 Choice of: 1. bitmapHeaderUncompressed 2. bitmapHeaderT82 3. bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list) If destinationAddress = softCopyPointerPlane Choice of: 1. bitmapHeaderUncompressed 2. bitmapHeaderT82 3. bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list)

bitmapData (Optional)	Encoded pixel data representing the bitmap (See Table 8-26).
moreToFollow	TRUE signals that more BitmapCreateContinuePDUs will follow, carrying additional data to complete the bitmap transaction. FALSE signals that the transaction is complete with this PDU and no more will follow.
NonStandard Parameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.4.3 Editing Bitmaps

...

TABLE 8-33

BitmapEditPDU Parameters

Parameter	**Description**
…	…
~~**boundingRectangle (Optional)**~~	~~This is a parameter which defines the extent of the bitmap relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this bitmap is to be considered within the copy region.~~
…	…

~~8.6.2.3 Bounding Rectangle~~

~~This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this drawing element shall be included in the region copied. This is regardless of the actual region spanned by the control points of the object. If this parameter is not specified, the control points of the object determine whether the drawing element is enclosed by the copy region. This parameter is mandatory if a non-zero rotation is specified for the drawing element on an editable workspace plane. This is to avoid ambiguities in determining the location of rotated control points.~~

...

8.6.3 Creating Drawing Elements

...

TABLE 8-51

DrawingCreatePDU

Parameter	**Description**
…	…
~~boundingRectangle (Optional)~~	~~This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this drawing element shall be included in the region copied regardless of the actual region spanned by the control points of the object. If this parameter is not specified, the control points of the object determine whether the drawing element is enclosed by the copy region. This parameter is mandatory if a non-zero rotation is specified for the drawing element on an editable workspace plane. This is to avoid ambiguities in determining the location of rotated control points.~~
…	…

8.6.5 Editing Drawing Elements

...

TABLE 8-54

DrawingEditPDU

Parameter	**Description**
…	…
~~boundingRectangleEdit (Optional)~~	~~This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region.~~

...

9. SIPDU Definitions

...

```
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
```

~~-- BoundingRectangle~~
~~--   Defines a rectangular region relative to an anchor point.~~

```
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
```

~~BoundingRectangle   ::=   SEQUENCE~~
~~{~~
~~upperLeft   PointDiff16,~~
~~lowerRight   PointDiff16~~
~~}~~

...

```
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
-- DrawingType
--         A DrawingType specifies the shape of a drawn element.
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
DrawingType  ::=   CHOICE
{
       point                        NULL,
                                             -- Unconnected points
       openPolyLine                 NULL,
                                             -- Points connected with straight lines.
                                             -- The last point is not connected to the
                                             -- first
       closedPolyLine               NULL,
                                             -- Points connected with straight lines.
                                             -- The last point is connected to the
                                             -- first
       rectangle                    NULL,
                                             -- A rectangle defined by two corners.
       Ellipse                      NULL,
                                             -- An ellipse
       nonStandardDrawingType  NonStandardIdentifier,
                                             -- Negotiated non-standard type
       ...
}

----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
-- EditablePlaneCopyDescriptor
--         Paired list of handles for source objects and their copies.
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
EditablePlaneCopyDescriptor     ::=      SEQUENCE
{
       objectList                                SEQUENCE (SIZE (1..65536)) OF SEQUENCE
       {
               sourceObjectHandle                Handle,
            destinationObjectHandle          Handle
                                                     -- This handle is used to reference
                                                     -- the new copy of the source object
                                                     -- in the future.
      },
      destinationOffset                      WorkspacePoint OPTIONAL,
                                                     -- This parameter defines an offset to be
                                                     -- added to the coordinates of all of the
                                                     -- copied objects.  If not present, zero
                                                     -- offset is assumed.
      planeClearFlag                         BOOLEAN,
                                                     -- When FALSE, the destination objects are
                                                     -- appended to the existing set of objects in
                                                     -- the destination plane.  When TRUE, all
                                                     -- existing objects in the destination plane
                                                     -- are deleted prior to the copy operation.
      …
}
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- PenThickness
--         This type specifies the thickness of the pen that is used to
--         draw graphical elements
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
PenThickness    ::=   INTEGER (1..255)


----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- PermanentPlaneCopyDescriptor
--         Describes source and destination regions within the corresponding
--         planes to be copied from and to. This is only to be used when the
--         source and destination planes are permanent.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
PermanentPlaneCopyDescriptor   ::=   SEQUENCE
{
        sourceRegion                         WorkspaceRegion,
                                                     -- Source rectangle to be copied.
        destinationRegion                    WorkspaceRegion,
                                                     -- Destination rectangle to be copied.
                                                     -- May be restricted by caps to be the
                                                     -- same size as the source region.
      …
}
```

...

```
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
-- BitmapCreatePDU
--          This PDU is used to initiate a bitmap transmission.
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
BitmapCreatePDU   ::=    SEQUENCE
{
      bitmapHandle                      Handle,
                                                -- Handle to be used to reference this object
                                                -- in the future
      destinationAddress                BitmapDestinationAddress,
                                                -- Destination address of the bitmap
      attributes                        SET OF BitmapAttribute OPTIONAL,
                                                -- List of editable attributes
                                                -- of the bitmap.
      anchorPoint                       WorkspacePoint OPTIONAL,
                                                -- Point of origin of the bitmap with
                                                -- respect to the destination workspace.
                                                -- Only needed for soft_copy bitmaps.
                                                -- Default is (0,0).
      bitmapSize                        ~~BitmapSize~~,
                                                -- Width and height of the total bitmap
                                                -- represented in the bitstream.  For a
                                                -- multi-component bitmap, this is the
                                                -- size of the largest component.
      bitmapRegionOfInterest            BitmapRegion OPTIONAL,
                                                -- Region of interest within the bitmap
                                                -- to be applied to the workspace
                                                -- Default is full bitmap.
      pixelAspectRatio                  PixelAspectRatio,
                                                -- Pixel aspect ratio of the bitmap
      scaling                           PointDiff16 OPTIONAL,
                                                -- Offset in workspace coordinates
                                                -- of the lower right hand corner of
                                                -- the bitmap relative to the anchor point.
                                                -- Default is no scaling.
                                                -- Only needed for softcopy bitmaps.
      ~~boundingRectangle       BoundingRectangle OPTIONAL,~~
                                                ~~-- Bounding rectangle of bitmap~~
                                                ~~-- specified relative to the anchor point.~~
                                                ~~-- Used in determining the extend of the bitmap~~
                                                ~~-- when a workspace region is being copied.~~
                                                ~~-- This parameter is mandatory if a non-one-to-one~~
                                                ~~-- pixel aspect ratio is specified for a bitmap~~
                                                ~~-- on an editable workspace plane.~~
      checkpoints                       SEQUENCE (SIZE (1..100)) OF TokenID OPTIONAL,
                                                -- Tokens to be used for checkpointing the
                                                -- bitmap create exchange
      bitmapFormatHeader                CHOICE
                                                -- The following headers provide image
                                                -- bitstream parameters that
                                                -- are outside the scope of the corresponding
                                                -- coding standard but
                                                -- are necessary for image decompression.
                                                -- NOTE - Some bitmap formats
```

```
                                          -- are disallowed depending on the value off
                                          -- the destinationAddress parameter.
   {
      bitmapHeaderUncompressed               BitmapHeaderUncompressed,
                                                -- Parameters for the
                                                -- uncompressed pixel
                                                -- representation
      bitmapHeaderT4                         BitmapHeaderT4,
                                                -- Parameters for T4 (G3)
                                                -- encoded bitstreams outside
                                                -- the T.4 standard's scope
      bitmapHeaderT6                         BitmapHeaderT6,
                                                -- Parameters for T6 (G4)
                                                -- encoded bitstreams outside
                                                -- the T.6 standard's scope
      bitmapHeaderT81                        BitmapHeaderT81,
                                                -- Parameters for T.81 (JPEG)
                                                -- encoded bitstreams outside
                                                -- the T.81 standard's scope
      bitmapHeaderT82                        BitmapHeaderT82,
                                                -- Parameters for T.82 (JBIG)
                                                -- encoded bitstreams outside
                                                -- the T.82 standard's scope
      bitmapHeaderNonStandard                NonStandardParameter,
      ...
   },
   bitmapData                BitmapData OPTIONAL,
                                   -- Compression format specific
                                   -- bitmap data
                                   -- padded to be byte aligned.
   moreToFollow              BOOLEAN,
                                   -- Indicates whether or not this is the last
                                   -- block of data for the bitmap
   nonStandardParameters     SET OF NonStandardParameter OPTIONAL,
                                   -- Allowed only if the corresponding
                                   -- non-standard capabilities are present
                                   -- in the negotiated capability set.
   ...
}
```

...

```
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-- BitmapEditPDU
--        This PDU is used to change bitmap attributes.
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
BitmapEditPDU   ::=    SEQUENCE
{
   bitmapHandle                    Handle,
                                         -- Handle used to reference this bitmap
   attributeEdits                  SET OF BitmapAttribute OPTIONAL,
                                         -- List of attributes to be edited
   anchorPointEdit                 WorkspacePoint OPTIONAL,
                                         -- Point of origin of the bitmap with
                                         -- respect to the destination workspace.
```

```
	bitmapRegionOfInterestEdit		BitmapRegion OPTIONAL,
						-- Region of interest within the bitmap
						-- to be applied to the workspace
	scalingEdit				PointDiff16 OPTIONAL,
						-- Offset in workspace coordinates
						-- of the lower right hand corner of
						-- the bitmap relative to the anchor point.
	~~BoundingRectangleEdit			BoundingRectangle OPTIONAL,~~
						~~-- Bounding rectangle of bitmap~~
						~~-- specified relative to the anchor point.~~
						~~-- Used in determining the extend of the bitmap~~
						~~-- when a workspace region is being copied.~~
	nonStandardParameters			SET OF NonStandardParameter OPTIONAL,
						-- Allowed only if the corresponding
						-- non-standard capabilities are present
						-- in the negotiated capability set.
	...
}
```

...

```
---------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
-- DrawingCreatePDU
--	A drawingCreate PDU is used to deposit one or more
--	drawing elements to a workspace plane.
---------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
DrawingCreatePDU ::=  SEQUENCE
{
	drawingHandle				Handle OPTIONAL,
						-- Handle to be used to reference this
						-- drawing object in future exchanges.  Note
						-- that editing and deleting objects is
						-- only valid if the target plane is
						-- of type "editable".
	destinationAddress			DrawingDestinationAddress,
						-- Destination of drawing
	drawingType				DrawingType,
						-- Which basic drawing shape this element
						-- represents.
	attributes				SET OF DrawingAttribute OPTIONAL,
						-- Attributes of the drawing object.
						-- NOTE - All attributes have default
						-- values that are assumed if the
						-- attribute is not specified
	anchorPoint				WorkspacePoint,
						-- Point of origin of the drawing element.
						-- This forms the first of the control points
						-- and is the point from which all other control
						-- points are defined relative to.
	rotation			RotationSpecifier OPTIONAL,
						-- Specifies a rotation angle and
						-- point of revolution for the drawing element.
~~    BoundingRectangle            BoundingRectangle OPTIONAL,~~
~~                                          -- Bounding rectangle of drawing element~~
~~                                          -- specified relative to the anchor point.~~
~~                                          -- Used in determining the extend of the object~~
~~                                          -- when a workspace region is being copied.~~
~~                                          -- This parameter is mandatory if a non-zero~~
~~                                          -- rotation is specified for a drawing element~~
~~                                          -- on an editable workspace plane.~~
    sampleRate                     INTEGER (1..255) OPTIONAL,
                                          -- For applicable types, this indicates
                                          -- the rate at which points were acquired
                                          -- by the transmitting terminal (in samples
                                          -- per second) so they can be
                                          -- replayed at a similar pace if desired
    pointList                      PointList,
                                          -- List of control points that define the
                                          -- drawing shape. The interpretation
                                          -- of the control point list is dependent on
                                          -- the value of the "type" parameter.
                                          -- Note that the control points in the list
                                          -- are differentially encoded from
                                          -- the previous.
    NonStandardParameters          SET OF NonStandardParameter OPTIONAL,
                                          -- Allowed only if the corresponding
                                          -- non-standard capabilities are present
                                          -- in the negotiated capability set.
  ...
}
```

...

```
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
-- DrawingEditPDU
--        A DrawingEditPDU is used to alter one or more of
--        a drawing elements attributes or parameters.
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
DrawingEditPDU ::= SEQUENCE
{
    drawingHandle                  Handle,
                                          -- Identifier of item to be edited.
    attributeEdits                 SET OF DrawingAttribute OPTIONAL,
                                          -- List of attribute changes.
    anchorPointEdit                WorkspacePoint OPTIONAL,
                                          -- Point of origin of the drawing element.
    rotationEdit                   RotationSpecifier OPTIONAL,
                                          -- Specifies a rotation angle and
                                          -- point of revolution for the drawing element.
~~    BoundingRectangleEdit          BoundingRectangle OPTIONAL,~~
~~                                          -- Bounding rectangle of drawing element~~
~~                                          -- specified relative to the anchor point.~~
~~                                          -- Used in determining the extend of the object~~
~~                                          -- when a workspace region is being copied.~~
```

```
    pointListEdits                  PointListEdits OPTIONAL,
                                          -- List of control point changes.
                                          -- Note that the index refers to
                                          -- the point list not including
                                          -- the anchor point.
    nonStandardParameters           SET OF NonStandardParameter OPTIONAL,
                                          -- Allowed only if the corresponding
                                          -- non-standard capabilities are present
                                          -- in the negotiated capability set.
    ...
}
```

...

```
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
-- WorkspacePlaneCopyPDU
--  This PDU causes a portion of a plane to be copied
--  to another plane (either intra or inter-workspace).
--  The source and destinations must either both be
--  permanent or both be editable, and they must have the same
--  usage designator otherwise copy for that plane will
--  not take place.
--  If the planes are editable, objects with any of their control
--  points falling totally within the source rectangle are copied.
--  If the Scaling capability has been negotiated in the case of a
--  softcopy workspace then it is not necessary for the source and
--  destination rectangles to be the same size.
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
WorkspacePlaneCopyPDU ::=  SEQUENCE
{
    sourceWorkspaceIdentifier             WorkspaceIdentifier,
                                                -- Workspace to be copied
    sourcePlane                           DataPlaneID,
                                                -- Source plane identifier.
    ~~SourceRegion                          WorkspaceRegion,~~
                                                ~~-- Source rectangle to be copied.~~
    destinationWorkspaceIdentifier        WorkspaceIdentifier,
                                                -- Destination workspace identifier.
                                                -- May be the same as the
                                                -- source workspace
    destinationPlane                      DataPlaneID,
                                                -- Destination plane identifier.
                                                -- May be the same as the
                                                -- source plane.
    ~~DestinationRegion                     WorkspaceRegion,~~
                                                ~~- Destination rectangle to be copied.~~
                                                ~~-- May be restricted by caps to be the~~
                                                ~~-- same size as the source region.~~         …
    copyDescriptor                        CHOICE
    {
        permanentPlaneCopyDescriptor            PermanentPlaneCopyDescriptor,
        editablePlaneCopyDescriptor       EditablePlaneCopyDescriptor,
        ...
    },
    nonStandardParameters               SET OF NonStandardParameter OPTIONAL,
                                                -- Allowed only if the corresponding
                                                -- non-standard capabilities are present
                                                -- in the negotiated capability set.
    ...
}
```

[End Correction]

6.7.7.15 Reuse of Handles by the Refresher and Operation of the Refresher in Conducted Mode

Description: When workspaces are being refreshed due to new SICEs joining a session, there is currently no mechanism to allow the refresher to indicate that a particular workspace corresponds to one that had previously existed, nor is it possible to indicate which objects within a refreshed workspace correspond to objects in the original workspace. This prevents a node that had been in the conference when the original workspaces were in existence from associating the old and new versions of the same workspaces and objects such that the effects of the deletion and retransmission of workspaces can be minimized. This can be corrected by allowing a refresher to reuse workspace and object handles that correlate in a one to one fashion with those that are being refreshed.

In the same section, the text is not clear about the absence of any requirements on the refresher to obtain SI and GCC privileges when in conducted mode to perform refresh operations.

[Begin Correction]

5.10 Conducted Mode Behavior

When a conference is in conducted mode, the SICE at the conductor node may grant a set of privileges to perform various actions to one or more nodes in the conference. Without such privileges or global conducted-mode permission from the conducting node, a SICE is restricted from performing these actions unless the SICE is the designated refresher for the session.

...

8.3.6 Workspace Refreshing for Late Arrivers

SICEs may choose to implement facilities to retransmit active (non archived) workspaces which have been deleted in response to the reception of a GCC-Application-Roster-Report indication from the GCC provider indicating that one or more new SICEs have joined the conference. The retransmitted data must conform to the constraints imposed by the new capabilities list contained within the GCC-Application-Roster-Report indication. A refresher is allowed to reuse workspace and object handles that correlate in a one to one fashion with those that are being refreshed. Such reuse would suggest an association between the old and new version of the same workspace or object to SICEs in the conference that were present before the late joiner that triggered the refresh events arrived. To guarantee that only one SICE in a conference performs this function (should multiple SICEs be capable of it), a SICE must attempt to establish itself as the refresher by first grabbing the SI-WORKSPACE-REFRESH-TOKEN. Upon successful acquisition, the SICE shall then broadcast a WorkspaceRefreshStatusPDU (see Table 8-22) with the parameters set to indicate that this SICE is the designated refresher. This indicates to other SICEs that the refresh token has been grabbed. In conducted mode sessions, the refresher need not be granted SI nor GCC conductor privileges to perform operations related to refreshing.

...

8.9 Conducted Mode Operation

...

Table 8-67 shows the effect of each of the privileges on the ability to transmit each SIPDU. The presence or absence of a privilege has no effect on the operation of receivers. If a SICE receives a PDU from another SICE which does not have the privilege to transmit that PDU, it shall process it as normal. A session refresher may initiate transactions needed to perform its function without any SI or GCC conductor privileges.

[End Correction]

6.7.7.16 Effects of Newly Negotiated Capability Sets

Description: The text needs to be corrected to properly indicate the applicability of changed capability sets. Expanding (relaxed restrictions) capability sets imposed by changes to the application roster due to nodes leaving a session apply to all workspaces.

An ambiguity regarding which workspaces to delete when a roster change includes new SICEs was also clarified.

[Begin Correction]

8.3.7 The Effect of Changes to the Application Roster

If a SICE receives a GCC-Application-Roster-Report indication from the GCC provider, it shall examine the new application roster. If the application roster indicates that no new SICEs have enrolled since the last roster instance, the SICE shall examine the received Application Capabilities List, and apply the rules indicated in Table 8-5 to generate the newly negotiated capability set. If either the capability set has not changed, or the capability set has been expanded (i.e. new capabilities have been added to the negotiated list, MIN capabilities have increased in their negotiated value, or MAX capabilities have decreased in their negotiated value, but the opposite has not occurred for any capability), current workspaces shall have their roster instances reset to the one reported by the new GCC-Application-Roster-Report indication and shall be bound from then on by the new capabilities set reported.~~no change is made to the current workspaces, but the next workspace that is created may make use of the new capability set, and the Application Roster Instance parameter shall be set to the new instance value indicated in the GCC-Application-Roster-Report indication.~~

...

~~In all cases, workspaces created using a previous Application Roster Instance retain the negotiated capability set of that instance in determining what operations are allowed for that workspace. All references in this Recommendation to the negotiated capability set in describing operations on workspaces imply the capability set for the Application Roster Instance specified in the WorkspaceCreatePDU, which is not necessarily the same as the current negotiated capability set.~~

When a new GCC-Application-Roster-Report is received which contains new SICEs in the conference, after the next creation of a workspace, all ~~other~~ workspaces that have Application Roster Instance numbers less than the Application Roster Instance number in the GCC-Application-Roster-Report are automatically deleted. A workspace refresher may recreate these workspaces once deleted (see clause 8.3.6).

[End Correction]

6.7.7.17 Workspace Caching

Description: The rules for workspace caching are ambiguous in the case where it is necessary to delete workspaces that have not been previously viewed or those that have the preserve flag set. This correction clarifies the rules for deletion order in these cases.

[Begin Correction]

8.3.8 Workspace Caching

...

In the case of a newly created Focus workspace, if a SICE does not have sufficient storage to create the workspace without deletion of another workspace, but it no longer has any workspaces in the previously viewed workspace queue (or if there are not enough to free-up sufficient space for the new workspace), it shall delete the previous Focus workspace, if there was one, regardless of whether the Preserve flag was set or not. If this still does not free up sufficient room for the new workspace, a workspace in the preserved workspace store may be deleted, although a SICE should be designed to avoid this situation by allowing room for a Focus workspace of maximum negotiated size. If this situation cannot be avoided, workspaces in the preserved workspace store shall be deleted in the following order until sufficient space becomes available for the new workspace. First delete workspaces with the Preserve flag set to FALSE in the order least recently created to most recently created, then delete workspaces with the Preserve flag set to TRUE in the order least recently created to most recently created.

[End Correction]

6.7.7.18 Unsynchronized BitmapCreatePDU Typographical Error

Description: A typographical error indicated that MCS-SEND-DATA was mandated for the CreateBitmapPDU in the case of an unsynchronized workspace when, in fact, either MCS-SEND-DATA or MCS-UNIFORM-SEND-DATA are allowed in this case.

[Begin Correction]

8.4.1 Creating Bitmaps

Before a bitmap create exchange is initiated, a SICE shall grab the SI-BITMAP-CREATE-TOKEN if the Soft-Copy-Bitmap-No-Token-Protection capability has not been successfully negotiated. Upon completion of the exchange, the token shall be freed if it was grabbed. This token is used to prevent multiple bitmap create exchanges from happening simultaneously in a conference. Note that if the destination workspace is unsynchronized, then some overlap may occur between two sequential bitmap create exchanges if~~due to the mandated use of~~ MCS-SEND-DATA is used instead of MCS-UNIFORM-SEND-DATA for unsynchronized workspaces.

[End Correction]

6.7.7.19 Application of Modifications During Reception of Bitmap Data

Description: PDUs initiating the creation of objects directed at a workspace plane can arrive interspersed between the set of PDUs used to create a bitmap object on the same plane. The text is ambiguous with respect to how these objects should be applied to the plane relative to the application of the bitmap data. Moreover, if a WorkspacePlaneCopyPDU is received during a multi-PDU bitmap creation, it is unclear whether the copy includes the new bitmap or not (either partially or wholly). In both cases, the start of the bitmap transmission shall be used as the reference when determining when to apply the bitmap data.

[Begin Correction]

8.4.1 Creating Bitmaps

...

For bitmap create exchanges involving bitmap data streams that exceed the maximum number of octets allowable per data parameter or that are purposely broken into smaller payloads for latency minimization, multiple PDUs must be used. BitmapCreateContinuePDUs shall be issued in a manner described in clause 6.2 until all encoded pixel data has been broadcast to the conference. The parameters of this PDU are described in Table 8-28. These PDUs shall be issued in order such that concatenation of their data payload at all receivers less pad bits forms an exact copy of the bitmap datastream at the transmitter. The last such PDU shall have the moreToFollow flag set to FALSE to signal the end of the transaction.

PDUs initiating the creation or editing of objects directed at a workspace plane can arrive interspersed between the set of PDUs used to create a bitmap object on the same plane. The reception of the BitmapCreatePDU (the initial PDU in the set) shall be used as the reference when determining how to apply the bitmap data relative to other object data. When the bitmap create sequence is completed, all object creation or modification PDUs received after the reception of the BitmapCreatePDU shall be rendered in the same manner as if the bitmap had been created with a single BitmapCreatePDU. Similarly, a WorkspacePlaneCopyPDU which includes this plane as the source and/or destination plane that arrives interspersed between the set of bitmap creation PDUs is treated in the same manner as if the bitmap had been created with a single BitmapCreatePDU. These rules apply whether the effected plane is permanent or editable.

[End Correction]

6.7.7.20 Limit of 255 Control Points Per Drawing Element is Incorrect

Description: There is an ambiguity in the description of the limits on the size of the control point lists for some drawing element types and a related ASN.1 error for the PointListEdits abstract type. The intent was to allow the specification of up to 65535 control points per point, closed and open polyline drawing elements. The DrawingCreatePDU limits the number of control points that can be specified upon initial creation to 255 but this is not intended to imply that more points cannot be added via the DrawingEditPDU. For example, the method by which an open polyline with 1024 control points is created is to issue the first DrawingCreatePDU with some number of the control points (less than 255) with subsequent issues of DrawingEditPDUs to extend the control point list. This was always the intent but was unfortunately not accurately recorded in the text.

[Begin Correction]

8.6.1.1 Point

A drawing element of type Point is a sequence of individual points. The anchor point, and the points in the point list correspond to the centres of the points to be drawn. Each point is created from the specified nib shape of the specified thickness. The control point definition is shown in Table 8-45.

TABLE 8-45

Definition of Point Drawing Shape Control Point List

Control Point Index	**Description**
-	Anchor point/First point to be drawn
0	Second point to be drawn relative to anchor point
1	Third point to be drawn relative to the previous control point
...	
N = (0..65534~~254~~)	Last point to be drawn relative to the previous control point. The DrawingCreatePDU only allows the specification of up to 255 initial control points. Subsequent DrawingEditPDU exchanges must be performed to extend the size of the control point list beyond this limit.

8.6.1.2 Open Polyline

A drawing element of type Open Polyline is a sequence of straight lines connecting the anchor point and successive points in the point list specified in the DrawingCreatePDU. For an open polyline, there is no line connected between the anchor point and the last point in the point list. In the case of an open polyline, there shall be at least one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with less than one point is received, the drawing element shall be ignored. The control point definition is shown in Table 8-46.

TABLE 8-46

Definition of Open Polyline Drawing Shape Control Point List

Control Point Index	Description
-	Anchor point
0	First line segment endpoint relative to anchor point
1	Second line segment endpoint relative to the previous control point
...	
N = (1..65534~~254~~)	Last line segment endpoint relative to the previous control point. The DrawingCreatePDU only allows the specification of up to 255 initial control points. Subsequent DrawingEditPDU exchanges must be performed to extend the size of the control point list beyond this limit.

8.6.1.3 Closed Polyline

A drawing element of type Closed Polyline is a sequence of straight lines connecting the anchor point and successive points in the point list specified in the DrawingCreatePDU. For a closed polyline, a line is also connected between the anchor point and the last point in the point list. In the case of a closed polyline, there shall be at least one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with less than one point is received, the drawing element shall be ignored. The control point definition is shown in Table 8-47.

TABLE 8-47

Definition of Closed Polyline Drawing Shape Control Point List

Control Point Index	Description
-	Anchor point
0	First line segment endpoint relative to anchor point
1	Second line segment endpoint relative to the previous control point
...	
N = (1..65534~~254~~)	Last line segment endpoint relative to the previous control point. The closing line segment is drawn to the starting point. The DrawingCreatePDU only allows the specification of up to 255 initial control points. Subsequent DrawingEditPDU exchanges must be performed to extend the size of the control point list beyond this limit.

8.6.1.4 Rectangle

A drawing element of type Rectangle is a rectangular region whose upper left corner and lower right corner are specified by the anchor point and a single point in the point list, respectively. In the case of a rectangle, there shall be exactly one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with more than one point is received, the additional points shall be ignored. If a list with less than one point is received in the case of a DrawingCreatePDU, the drawing element shall be ignored. The control point definition is shown in Table 8-48.

TABLE 8-48

Definition of Rectangle Drawing Shape

Control Point Index	Description
-	Anchor point/Upper Left Corner
0	Lower Right Corner relative to anchor point

8.6.1.5 Ellipse

A drawing element of type Ellipse is defined by a bounding rectangle whose upper left corner and lower right corner are specified by the anchor point and a single point in the point list, respectively. The size of this rectangle corresponds to the length of the two axes of the ellipse. The ellipse is positioned so that no part of it protrudes beyond the bounding rectangle. The ellipse drawing type shall not be used unless the Soft-Copy-Annotation-Drawing-Ellipse capability is present in the negotiated capability set. In the case of an ellipse, there shall be exactly one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with more than one point is received, the additional points shall be ignored. If a list with less than one point is received in the case of a DrawingCreatePDU, the drawing element shall be ignored. The control point definition is shown in Table 8-49.

NOTE - The bounding rectangle used in defining this drawing element does not correspond to an actual area of the workspace overwritten by this drawing element. Only the line which forms the ellipse itself (and optionally the fill area within the ellipse) is modified in the case of a non-editable workspace plane, or opaque in the case of an editable workspace plane.

TABLE 8-49

Definition of Ellipse Drawing Shape

Control Point Index	Description
-	Anchor point/Upper Left Corner of bounding rectangle
0	Lower Right Corner of bounding rectangle relative to anchor point

8.6.1.6 Non-Standard

A drawing element of Non-Standard shape is permitted to have from 1 up to 65534~~256~~ control points in the point list. The meaning of these control points is beyond the scope of this Recommendation. The control point definition is shown in Table 8-50.

NOTE - Non-Standard drawing shapes should be specified such that the drawing element does not extend beyond the smallest size rectangle which can enclose all of the control points. This allows the decision of whether the rectangular copy region for the WorkspacePlaneCopyPDU surrounds the control points (to determine whether the drawing element is to be included in the copy) to be consistent with whether that copy region surrounds the actual drawing element.

TABLE 8-50

Definition of Non-Standard Drawing Shape Control Point List

Control Point Index	Description
-	Anchor point/Non-Standard Control Point 0
0	Non-Standard Control Point 1 relative to anchor point
1	Non-Standard Control Point 2 relative to the previous control point
...	
N = (0..65534~~254~~)	Non-Standard Control Point N relative to the previous control point. The DrawingCreatePDU only allows the specification of up to 255 initial control points. Subsequent DrawingEditPDU exchanges must be performed to extend the size of the control point list beyond this limit.

...

9. SIPDU Definitions

...

PointListEdits ::= SEQUENCE SIZE (1..255) OF SEQUENCE
{
initialIndex INTEGER (0..65534~~254~~)
initialPointEdit PointDiff16
subsequentPointEdits PointList OPTIONAL
...
}

[End Correction]

6.7.7.21 The Positions of Drawing Element Lines and Curves That are Greater Than One Pixel Wide are Ambiguous

Description: The Recommendation does not define how drawing element component lines and curves drawn with a pen thickness greater than 1 pixel should be aligned with the drawing trajectory specified by the control point list.

[Begin Correction]

8.6.2.4 Pen Thickness

The pen thickness parameter is an optional parameter (part of the drawing attribute set) which determines the thickness of the line The pen thickness specifies the width of pen nib used to draw the line portion of a drawing element in units of pixels. If drawing is supported in a conference, the pen thickness range of 3 to 16 pixels must be supported. A wider range than this may be negotiated via the Soft-Copy-Annotation-Drawing-Pen-Min-Thickness and Soft-Copy-Annotation-Drawing-Pen-Max-Thickness capabilities. The minimum thickness may be negotiated as low as one pixel, and the maximum as high as 255 pixels. If this parameter is not specified, a default value of 3 shall be used. For line thicknesses greater than one pixel, the drawing element component lines and curves shall be drawn centered about the trajectory defined by the related control point list.

[End Correction]

6.7.7.22 Square Nib Orientation

Description: The orientation of the square drawing nib with respect to the workspace coordinate axes need be specified in the recommendation.

[Begin Correction]

8.6.2.6 Pen Nib

The pen nib parameter is an optional parameter (part of the drawing attribute set) which determines the shape of the nib used to draw the line portion of drawing elements. All lines are composed of a nib continuously translated along the path of the line (or curve). In the case of a line style with dots or dashes, the nib is translated along the path of the line with periodic gaps where the nib is effectively raised and lowered again on the other side of the gap. If the pen nib parameter is not present, a circular nib shall be used. The circular nib is defined to be a solid circle of diameter equal to the pen thickness. If the Soft-Copy-Annotation-Drawing-Pen-Square-Nib parameter is present in the negotiated capability set for the designated workspace, a square nib may be used instead by specifying square as the pen nib parameter. The square nib is defined to be a solid square region with the pen thickness as the length of each side. The sides of the nib are parallel or perpendicular to either the X or Y axis of the workspace coordinate system.

[End Correction]

6.7.7.23 Archive Open Rejection and Multiple Opens

Description: The requirements of a SICE when an attempt to open an archive is rejected are not currently defined in the text. The use of an ArchiveClosePDU to indicate this situation is appropriate. In a related situation, when an archive must be closed to accommodate a late joiner, automatic closing of the archive is specified in the current text. This would result in the possibility of closing the archive in different states at each SICE. To correct this, the use of an explicit ArchiveClosePDU in this situation is also appropriate. Finally, the uniqueness requirement of archive handles for each open instances of the same archive is not clear in the current text.

[Begin Correction]

8.8 Archives

An archive is a collection of workspaces which may be saved beyond the extent of a single conference. If an archive is present at every peer SICE in a conference, it may be opened by a SICE if the Archive-Support capability is present in the negotiated capability set. Once opened, that SICE may perform workspace operations on workspaces contained within. An archive may be opened for reading, writing, creation (of a new archive), or any combination of these. Multiple SICEs may simultaneously open an archive for reading, but only one SICE may open the same archive for creation or writing. If multiple SICEs open an archive, each open shall use a unique archiveHandle. To open an archive, the ArchiveOpenPDU shall be broadcast to all peer SICEs. This is done in the manner described in Table 6-3. The content of the ArchiveOpenPDU is shown in Table 8-61. Upon receiving the ArchiveOpenPDU, all SICEs shall send an ArchiveAcknowledgePDU to the SICE requesting the open on its Used ID channel. The SICE requesting the open shall wait until all peer SICEs marked as Active in the current application roster have responded to the open request before proceeding with archive operations. If not all Active peer SICEs acknowledge, the archive shall not be considered open. If the result parameter in any of the ArchiveAcknowledgePDUs returned indicates an unsuccessful open, The SICE issuing the ArchiveOpenPDU shall explicitly issue an ArchiveClosePDU specifying the same archive handle included in the failed open transaction. This insures that all SICEs that successfully opened the archive close it. The SICE shall also monitor any roster changes as indicated from GCC so as to be able to recognize when an Active peer SICE that has not responded has left the conference. If the roster changes with any new SICEs being added to the conference since the roster instance that was valid at the time the archive was opened, the archive shall~~is assumed to~~ be ~~automatically~~ closed by issuing an ArchiveClosePDU.~~and that a~~ Another ArchiveOpenPDU must be transmitted in order to perform any further archive operations.

[End Correction]

6.7.7.24 Use of Object Handles in Archive Copy Operations

Description: Handles for objects stored in an archive need to be guaranteed to be unique across sessions within which the archive could be used. The current text does not ensure this.

[Begin Correction]

8.8 Archives

...

Once an archive has been opened, it may be operated on by the SICE which opened it using any of the workspace PDUs. A SICE which has not opened an archive shall not perform any archive operations on that archive. If an archive has been opened for reading but not writing (or creation), workspaces may not be created, deleted, or edited. Only workspace plane copy operations are allowed where the source is a workspace in this archive and the destination is an active workspace or a workspace in another archive. WorkspacePlaneCopy operations involving editable planes shall substitute an ordinal number for the normal GCC unique handles used to reference objects. When copying objects into an archive, the ordinal number shall be the position of the object in the EditablePlaneCopyDescriptor object list (index base 0) if the planeClearFlag is set. If the planeClearFlag is not set, the handle shall be the ordinal number representing the position of the object in the EditablePlaneCopyDescriptor object list (index base 0) added to the greatest handle value of any object that exist within the destination plane at the time of the copy. If an archive has been opened for writing (or creation) but not for reading, workspaces may be created, deleted, or edited, and workspace plane copies may be performed where the source workspace is an active workspace or a workspace in another archive and the destination is a workspace in this archive. If an archive is opened for both reading and writing (or creation), any workspace operation may be performed. When performing a workspace operation on an archived workspace, the workspaceIdentifier parameter is set to the Handle of the archive as indicated in the ArchiveOpenPDU, along with the entry name of the particular workspace within the archive.

[End Correction]

6.7.7.25 Correction of Encoding Rule Variant

Description: In the description of the encoding rules used to encode the SIPDUs, one aspect of the variant of Packed Encoding Rules to be used was left ambiguous.

[Begin Correction]

9. SIPDU Definitions

Each SIPDU is transported as one MCSSDU across an MCS connection. A standard ASN.1 data value encoding is used to transfer SIPDUs between peer SICEs. For all PDUs, the BASIC ALIGNED variant of the Packed Encoding Rules of ITU-T Recommendation X.691 shall be used.

[End Correction]

6.7.7.26 Referencing T.120 for Static Channel and Token IDs

Description: The expected approval of Recommendation T.120 allows removal of Annex B which defined the numeric values of the Static Channel and Token IDs. It had been agreed that T.120 is to be the central location to document the Static Channel and Token IDs.

[Begin Correction]

~~Annex B~~

~~Static Channel and Token ID Assignments~~

~~(this annex forms an integral part of this Recommendation)~~

~~1.1 Static Channel ID Assignments~~

~~Table B-1 lists the numerical assignment of static channel IDs for the static channels allocated for use by this Recommendation. The numerical assignment of static channel IDs is intended to be centralized in Recommendation T.120, but is included here until T.120 is completed.~~

~~Table B-1 – Static Channel ID Assignments~~

~~Symbolic Name~~	**~~Channel ID~~**
~~SI-CHANNEL-0~~	~~8~~

~~1.2 Static Token ID Assignments~~

~~Table B-2 lists the numerical assignment of static token IDs for the static tokens allocated for use by this Recommendation. The numerical assignment of static token IDs is intended to be centralized in Recommendation T.120, but is included here until T.120 is completed.~~

~~Table B-2 – Static Token ID Assignments~~

~~Symbolic Name~~	**~~Token ID~~**
~~SI-TOKEN-0~~	~~8~~
~~SI-TOKEN-1~~	~~9~~

[End Correction]

6.7.7.27 Correction of Object Identifier

Description: In the Object Identifier used to define the T.126 Application Protocol Key, the keyword "itu" should properly be "itu-t".

[Begin Correction]

ANNEX C

Object Identifier Assignments

(this annex forms an integral part of this Recommendation)

Table C-3 lists the assignment of Object Identifiers defined for use by this Recommendation.

TABLE C-3

Object Identifier Value	**Description**
{itu-t recommendation t 126 version(0) 1}	This Object Identifier is used to indicate the version of this Recommendation. At this time there is a single standardized version defined.

[End Correction]

6.7.7.28 Removal of Inappropriate Uses of the Term "Note"

Description: In several locations in the text, the term "note" is used inappropriately. These have been removed.

[Begin Correction]

8.4.1 Creating Bitmaps

...

If present, a checkpoint token shall be uninhibited by each receiver as its corresponding bitmapData payload is made ready for display locally. ~~Note that a~~A checkpoint token (delivered in the optional checkpoint parameter of the BitmapCreatePDU and BitmapCreateContinuePDU) is considered to correspond to a bitmapData payload if it was delivered via the same PDU.

...

TABLE 8-29

BitmapCheckpointPDU Parameters

Parameter	**Description**
bitmapHandle	This parameter references the bitmap and shall be specified with the same value used in the BitmapCreatePDU for this exchange.
passedCheckpoint	This parameter a list of tokens corresponding to passed checkpoints by the transmitting SICE. ~~Note that a~~A checkpoint is considered passed if all the receiving terminals have uninhibited it. Receiving terminals can display the portion of the bitmap in transit corresponding to the passed checkpoints specified by this parameter.
percentComplete	This parameter's value reflects the cumulative percentage of the bitmap exchange that is complete inclusive of the bitmap information inferred by the passedCheckpoint parameter. The value range is (1..100).

...

8.4.3 Editing Bitmaps

...

TABLE 8-33

BitmapEditPDU Parameters

Parameter	Description
…	…
bitmapRegionOf InterestEdit (Optional)	This optional parameter selects the sub-region within the associated bitmap that is to be displayed. ~~Note that a~~A SICE is required to store the entire transmitted bitmap if the destination plane is editable and the destination is not a hard copy device. If this parameter is not present, the region of interest is not modified.
…	…

...

8.7.2 Remote Pointing Device Events

When a SICE that has been granted pointingDeviceEvent permission wishes to send a pointing device event, it shall do so by sending a RemotePointingDeviceEventPDU to the SICE which created the workspace (the issuer of the WorkspaceCreatePDU). This is done in the manner indicated in Table 6-3. The content of the RemotePointingDeviceEventPDU is shown in Table 8-58. The initial remote pointer position shall be undefined and all button states shall be assumed to be buttonUp by the SICE granting the pointingDeviceEvent permission until receipt of the first RemotePointingDeviceEventPDU. ~~Note that a~~A SICE may grant the pointingDeviceEvent permission to multiple other SICEs. How multiple pointing device states are simultaneously interpreted or merged is beyond the scope of this Recommendation. Note, though, that care must be taken to properly handle the case of a SICE with pointingDeviceEvent permission leaving the conference or having its pointingDeviceEvent permission revoked with any button not in a buttonUp state.

[End Correction]

6.7.8 Technical and Editorial Corrections to ITU-T Recommendation T.127

None.

T.120 Recommendation Series Defect Report Form

DATE:

CONTACT INFORMATION

NAME:
COMPANY:
ADDRESS:

TEL:
FAX:
E-MAIL:

AFFECTED RECOMMENDATIONS:

DESCRIPTION OF PROBLEM:

SUGGESTIONS FOR RESOLUTION:

NOTE - Attach additional pages if more space is required than is provided above

7 Implementor's Guide for T.170-series

The Implementor's Guide for the T.170-series will be forwarded by the Rapporteur for Q.2/16 (Mr. Blaschitz) on the basis of TD 24, which can be considered technically stable. It was however considered that editing and updating work has to take place in order to completely align TD 24 with the T.170-series of Recommendations.

8 Implementor's Guide for H.310

ABSTRACT: This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T H.310 (H.222.1 and H.245)-series Recommendations. It is intended to be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.310-series Recommendations.

Contact Information

ITU-T Study Group 15/ Question 2 Rapporteur	Sakae Okubo Graphics Communication Lab 6F Annex Toshin Bldg 4-36-19 Yoyogi Shibuya-ku, Tokyo 151 Japan	Tel: Fax: email:	+81 3 5351 0181 +81 3 5351 0185 okubo@gctech.co.jp
Implementers Guide Editor	Jeff Lynch IBM Corporation CE6A/B664 800 Park Offices Drive RTP, NC, 27709, USA	Tel: Fax: email:	+1-919-254-4454 +1-919-254-5483 jjlynch@vnet.ibm.com
ITU-T Recommendation H.310 Editor	Sakae Okubo Graphics Communication Lab 6F Annex Toshin Bldg 4-36-19 Yoyogi Shibuya-ku, Tokyo 151 Japan	Tel: Fax: email:	+81 3 5351 0181 +81 3 5351 0185 okubo@gctech.co.jp
ITU-T Recommendation H.222.1 Editor			
ITU-T Recommendation H.245 Editor	Mike Nilsson BT Martlesham Heath Ipswich, Suffolk IP5 7RE England	Tel: Fax: email:	+44-1473-645413 +44-1473-643791 nilsson_m_e@bt-web.bt.co.uk

Document History

Revision	Date	Description
A	20 December 1996	Initial draft - completed at the ITU-T Study Group 16 Rapporteurs meeting, Boulder, Colorado, December 1996
B	21 February 1997	Revision B - completed at the ITU-T Study Group 16 Rapporteurs meeting, Boston, MA, February 1997
C	27 March 1997	Revision C - completed at the ITU-T Study Group 16 meeting, Geneva, Switzerland, March 1997

8.1 Introduction

This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T H.310-series Recommendations. It is intended to be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementor's. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.310-series Recommendations.

8.2 Scope

This Guide resolves defects in the following categories:

- editorial errors;
- technical errors such as omissions or inconsistencies;
- ambiguities.

In addition the 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 the normal way through contributions to the ITU-T.

8.3 Policies for Updating This Document

This document is managed by the ITU-T Study Group 16 Question 12 Rapporteur's group. It can be revised at any recognized Q.12/16 Rapporteur's group meeting provided the proposed revisions are accepted by the members of the group. A revision history cataloguing the evolution of this document is included.

8.4 Defect Resolution Procedure

Upon discovering technical defects with any components of the H.310 Recommendations series, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q.12/16 Rapporteur. The template for a defect report is enclosed. Contact information for these parties is included in this 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.310-series Recommendations. Formal membership in the ITU is not required to participate in this process.

8.5 References

This document refers to the following H.310-series Recommendations:

1. "Recommendation H.310: Broadband Audiovisual Communication Systems and Terminals" *International Telecommunication Union, Telecommunication Standardization Sector, Study Group 15*, 1996.
2. "Draft Recommendation H.245 (Version 2): Control Protocol for Multimedia Communication" *International Telecommunication Union, Telecommunication Standardization Sector, Study Group 15*, 1996.
3. "Recommendation H.222.1: Multimedia Multiplex and Synchronization for Audiovisual Communication in ATM Environments" *International Telecommunication Union, Telecommunication Standardization Sector, Study Group 15*, 1996.
4. "ITU-T Recommendation H.222.0-ISO/IEC 13818-1 (1995) Generic Coding of Moving Pictures and Associated Audio Information: Systems" *International Organization for Standardisation, Organization Internationale De Normalisation (ISO/IEC JTC 1/SC 29/WG 11)*

8.6 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.

8.7 Technical and Editorial Corrections

8.7.1 Technical and Editorial Corrections to ITU-T Recommendation H.310

8.7.1.1 Error in Structured Data Transfer abbreviation

Description: Error in Structured Data Transfer abbreviation

[Begin Correction]

4 Abbreviations

...

SDT ~~Synchronous~~Structured Data Transfer

...

[End Correction]

8.7.1.2 Reference to DSS-2 Documents

Description: Reference to DSS-2 Q.2941.1 and Q.2961.2 added

[Begin Correction]

6.1 System Configuration

TABLE 1/H.310

Summary of DSS2 protocols

ITU-T Rec.	Title
....	
Q.2941.1	Recommendation Q.2941 Part 1 - DSS2 Generic Identifier Transport Capability
Q.2961.2	Broadband-integrated services digital network (B-ISDN) - Digital Subscriber Signalling System No. 2 (DSS 2), Section 2: ATM Transfer capability coding in the broadband bearer capability information element.
....	

[End Correction]

8.7.1.3 Logical Channel Signalling

Description: When a bi-directional channel is needed, a bi-directional logical channel should be setup up using H.245 signalling instead of setting up of a pair of unidirectional logical channels. Using this convention, there is no need to develop an association mechanism for a pair of unidirectional logical channels. The note identifying this for further study needs to be removed.

[Begin Correction]

6.4.3 Logical channel signalling

Each logical channel carries information from a transmitter to a receiver, and is identified by a logical channel number unique for each direction of transmission.

Logical channels are opened and closed using the **OpenLogicalChannel** and **CloseLogicalChannel** messages and procedures of H.245.

When a logical channel is opened, the **OpenLogicalChannel** message fully describes the content of the logical channel, including media type, algorithm in use and any options, and all other information needed for the receiver to interpret the contents of the logical channel. Logical channels may be closed when no longer needed.

Certain media types, including data protocols such as T.120, inherently require a bi-directional channel for their operation. In such cases a bi-directional logical channel, which is capable of information transmission in both directions, may be opened using the bi-directional channel opening procedures of H.245.

NOTE - A pair of uni-directional logical channels, each of which is opened by the open logical channel procedure, can be applied to a bi-directional operation of those media providing the transmission in each direction. However, the use of bi-directional logical channel is strongly recommended for bi-directional media in this Recommendation. ~~The association mechanism for a pair of uni-directional logical channels is for further study.~~

[End Correction]

8.7.1.4 H.310 CorrelationID

Description: The H.245 "resourceID" parameter is used to associate a VC to a logical channel. The use of the H.310 CorrelationID and its relationship to the H.245 resourceID is not clearly defined.

When multiple VCs are involved in an H.310 communication, H.245 needs to be able to indicate a particular logical channel in a particular VC. Conventions are necessary for the:

- Association of the multiple VCs originated at a particular terminal
- Identification of each VC

Since H.310 currently assumes that a single session exists between any two instances of H.310, the association of the multiple VCs can be uniquely identified by the Calling party number/Called party number and Calling party sub-address/Called party sub-address in the Q.2931 SETUP message. The identification of each VC is accomplished by using the H.245 resourceID (2 bytes). This is depicted in Figure 1.

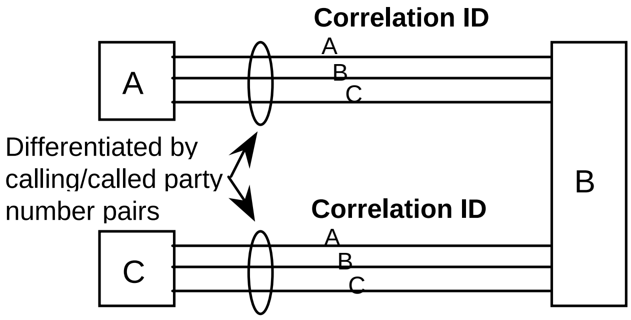


Figure 1: Single session case

To prepare for the eventual possibility that multiple H.310 sessions will exist between two terminals, as shown in Figure 2, the correlationID has been defined as follows:

correlationID= session ID; Identification of a session
+
resource ID (2 bytes) ; Identification of a logical channel

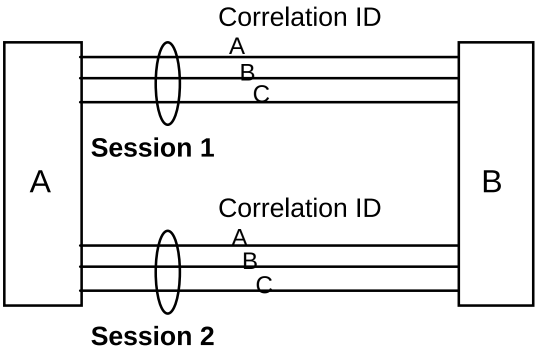


Figure 2: Multisession case

However, the current H.310 specification does not assume the multi-session case so the resourceID is the only valid element in the correlation ID for the moment.

[Begin Correction]

7.1 Native H.310 Communication Call Procedures

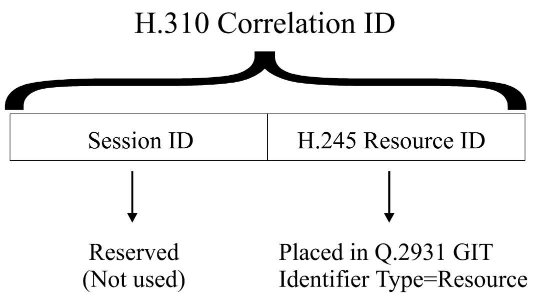
The H.310 correlationID is composed of a session identification field and the 2 octet H.245 "resourceID". The H.245 resourceID in the H222LogicalChannelParameters is used to indicate which ATM Virtual Channel the logical channel is associated. The session identification field is currently undefined and is reserved for future use.

H.310 CorrelationID	
Session ID (Reserved)	H.245 resourceID

In allocating H.245 resourceIDs, the side that set up the initial H.245 VC chooses values starting from the lowest possible value and the other side (the called side), chooses values starting from the highest to cope with the case that both sides may simultaneously set up a new VC.

The Generic Identifier Transport (GIT) Information Element shall be used for the transmission of the H.310 correlationID in the Q.2931 SETUP message. The GIT information element is specified in Q.2941.1.

Since the resourceID is the only valid element in the correlation ID at this time, only the resourceID is sent in the Q.2931 SETUP. This is pictorially shown in the following figure. When coding the GIT, the identifier related standard/application is coded as Recommendation H.310, the identifier type as Resource, and the identifier length as two octets.



Optionally, the H.310 correlationID may also be sent in the Broadband High Layer IE (B-HLI) as follows.

Broadband high layer information (B-HLI)

Information Element	Value
High layer information type	'000 0001' user -specific
High layer information (octets 6 & 7)	H.245 ResourceID

In the case where either of the terminals do not support the GIT information element, the B-HLI information element shall be used to convey the H.310 correlationID

[End Correction]

8.7.1.5 Use of multiple B-LLI

Description: Currently there is no guidance in H.310 on how to use the B-LLI Information Element.

When a terminal has multiple protocol parameters, it can send multiple B-LLI information elements to the called terminal in the Q.2931 SETUP message at a call establish stage. A description of this procedure is needed.

The Q.2931 Annex C defines the procedure for sending multiple B-LLI information element.

- Up to three B-LLI information element may be included in a SETUP message.
- The first B-LLI information element in the message is preceded by the Broadband repeat indicator information element in the case of multiple B-LLI information element transport.
- The called terminal indicates a single choice from among the options offered in the SETUP messages by including the B-LLI information element in the CONNECT message.

[Begin Correction]

7.1 Native H.310 Communication Call Procedures

The H.310 terminal type and multiplexing capability shall be conveyed using the Broadband Low Layer IE in the ATM Signalling SETUP message. Since multiple H.310 terminal types such as RAST and SOT, cannot be expressed by a single B-LLI information element, the calling terminal may send multiple B-LLI information elements according to Q.2931 Annex C. Each B-LLI shall indicate a different set of H.310 terminal capabilities.

Since the terminal capabilities of H.310 terminals are not symmetric, the following convention has been adopted.

> The terminal sending the SETUP (the calling terminal) shall identify it's own capabilities in the B- LLI's. The H.310 terminal receiving the SETUP (the called terminal) must determine if it can interoperate with the identified capabilities of the sending terminal. If a complementary mode of operation is present in the received B-LLIs, then the call is accepted. When multiple B-LLI's are present the called terminal shall select the preferred mode of operation that compliments its own capabilities.

For example, if the terminal sending the SETUP message indicates that it has the capability to be a SOT, then a ROT terminal knows that it can interoperate with it so it can accept the call.

To ensure maximum interoperability, a RAST terminal may indicate that it has the capability to be a RAST, SOT, and ROT as separate B-LLIs. The B-LLI's are included in the preferred order in the Q.2931 SETUP message. The called terminal shall choose one B-LLI information element and return it to the calling terminal in the CONNECT message.

When multiple B-LLI's are included in the SETUP that complement the capabilities of the receiving terminal , the procedure for selecting the preferred B-LLI at the called terminal is outside the scope of this specification.

[End Correction]

8.7.1.6 Update of Annex B: Use of Q.2931 Signalling by H.310 Terminals

Description: The current version of H.310 does not address the Q.2931 Signalling necessary for setting up of the video and audio virtual circuits.

This also includes various changes to align with the latest Q.2931 definition.

To ensure interoperation, the following section will be incorporated into Annex B "SETUP Message parameters".

This is a complete replacement for Annex B.

[Begin Correction]

ANNEX B

(This annex forms an integral part of this Recommendation)

Use of Q.2931 Signalling by H.310 Terminals

B.1 Q.2931 Signalling Information Elements

The following Information Elements are required in the SETUP message to establish the following types of connections for H.310 terminals.

TABLE 1

Q.2931 Signalling Information Elements

Information Element	**H.310 Control VC**	**H.310 RASTA/V VC**	**H.310 ROT/SOT A/V VC**
Protocol discriminator	M	M	M
Call Reference	M	M	M
Message type	M	M	M
Message length	M	M	M
AAL Parameters	M	M	M
ATM Traffic descriptor	M	M	M
Broadband bearer capability	M	M	M
Broadband repeat indicator	CM	O	O
Broadband low layer information	M	M	M
Generic Identifier Transport	M	M	M
Broadband High layer information	O	O	O
Notification Indicator	O	O	O
Called party number	M	M	M
Called party subaddress	C	C	C
Calling party number	M	M	M
Calling party subaddress	C	C	C
Connection identifier	O	O	O
Extended QOS parameters	O	O	O
End to end transit delay	O	O	O
QoS parameter	M	M	M
Broadband sending complete	C	C	C
Transit network selection	O	O	O
Endpoint reference	NA	NA	NA

NA - Not Applicable M - Required
O - Optional C - Conditional (if appropriate for the network being used)
CM - Conditionally Mandatory

The next sections summarize the use of the information elements that are unique to H.310 RAST/ROT/SOT. The elements that are not specifically described here are coded in accordance with the requirements of Q.2931 Specification.

TABLE 2

Q.2931 Signalling Information Elements Specific To H.310 RAST-1 and ROT/SOT-1

Information Element	H.310 Control VC	H.310 RAST-1 A/V VC	H.310 ROT/SOT-1 A/V VC
AAL Parameters	AAL Type = AAL-5 Forward CPCS SDU size Backward CPCS SDU size SSCS Type = Frame Relay	AAL Type = AAL-1 Subtype = Video signal transport CBR rate = n x 64 Multiplier = as negotiated by H.245 Source Clock Frequency Recovery = as negotiated by H.245 Error Correction = as negotiated by H.245 SDT Block Size = Impl. Specific	
ATM Traffic descriptor	Forward PCR and Backward PCR = 167 cells per second which corresponds to 64 kbit/s Sustainable Cell Rate = Impl. Specific Maximum burst size = 2048	Forward PCR = Implementation Specific Backward PCR = Implementation Specific Note: Forward and Backwards are set according to which terminal sends the original SETUP message.	
Broadband bearer capability	Bearer Class = BCOB-X Broadband Transfer Capability = BTC10 User Plane Connection = PtP	Bearer Class = BCOB-A Broadband Transfer Capability = CBR Timing Requirements = end-to-end timing required User Plane Connection = PtP	
Broadband repeat indicator	Set when more than one B-LLI element is present		
Broadband low layer information	User Layer 3 = H.310 Terminal type = ROT/SOT/RAST Forward Multiplexing = NO Backward Multiplexing = NO	User Layer 3 = H.310 Terminal type = RAST Forward Multiplexing = TS Backward Multiplexing = TS	User Layer 3 = H.310 Terminal type = ROT/SOT Forward Multiplexing = TS Backward Multiplexing = Null Note: Forward and Backwards are set according to which terminal sends the original SETUP message
Generic Identifier Transport		H.310 Correlation ID Note: if GIT is not available, see Appendix II.	

TABLE 3

Q.2931 Signalling Information Elements Specific To H.310 RAST-5 and ROT/SOT-5

Information Element	**H.310 Control VC**	**H.310 RAST-5 A/V VC**	**H.310 ROT/SOT-5 A/V VC**
AAL Parameters	AAL Type = AAL-5 Forward CPCS SDU size Backward CPCS SDU size SSCS Type = Frame Relay	AAL Type = AAL-5 Forward CPCS SDU size Backward CPCS SDU size SSCS Type = Null	
ATM Traffic descriptor	Forward PCR and Backward PCR = 167 cells per second which corresponds to 64 kbit/s Sustainable Cell Rate = Impl. specific Maximum burst size = 2048	Forward PCR = Implementation Specific Backward PCR = Implementation Specific Note: Forward and Backwards are set according to which terminal sends the original SETUP message.	
Broadband bearer capability	Bearer Class = BCOB-X Broadband Transfer Capability = BTC10 User Plane Connection = PtP	Bearer Class = BCOB-X Broadband Transfer Capability = CBR User Plane Connection = PtP	
Broadband repeat indicator	Set when more than one B-LLI element is present		
Broadband low layer information	User Layer 3 = H.310 Terminal type = ROT/SOT/RAST Forward Multiplexing = NO Backward Multiplexing = NO	User Layer 3 = H.310 Terminal type = RAST Forward Multiplexing = TS Backward Multiplexing = TS	User Layer 3 = H.310 Terminal type = ROT/SOT Forward Multiplexing = TS Backward Multiplexing = Null Note: Forward and Backwards are set according to which terminal sends the original SETUP message.
Generic Identifier Transport		H.310 Correlation ID Note: if GIT is not available, see Appendix II .	

B.2 ATM Signalling Required for the H.310 Control VC

The following tables define the ATM signalling information elements for the H.310 control channel. The AAL, Traffic Descriptor and Broadband Bearer Capabilities IEs are defined similarly for other applications.

The control VC Protocol stack can be assumed based on Terminal identification exchanged in B-LLI. No need for multiple B-LLIs for this purpose. The Quality of Service (QoS) parameters shall be coded in accordance with the requirements of the network and are not specifically defined here.

TABLE 4

AAL Parameters IE

IE Parameter	Value	Notes
AAL type	AAL-5.	
Forward Maximum AAL-5 CPCS SDU size		Value sufficient for transporting H.245 messages up to 2048 octets
Backward Maximum AAL-5 CPCS-SDU size		Value sufficient for transporting H.245 messages up to 2048 octets
SSCS Type	'00000100'	Frame Relay SSCS

TABLE 5

ATM Traffic Descriptor IE

IE Parameter	Value	Notes
Forward Peak Cell Rate	implementation and program selection specific	Set to the Peak Cell Rate value required for the control connection
Backward Peak Cell Rate	implementation and program selection specific	Set to the Peak Cell Rate value required for the control connection

TABLE 6

Broadband Bearer Capabilities IE

IE Parameter	Value	Notes
Bearer Class	BCOB-X	
Broadband Transfer Capability	BTC10='01010'	VBR with End-to-End Timing not required (Non-realtime VBR)
User Plane Connection configuration	Point-to-Point	

TABLE 7

Broadband Low Layer IE

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU-T Recommendations H.310
Terminal Type	'0001' = H.310 ROT '0010' = H.310 SOT '0011' = H.310 RAST	Specify the appropriate terminal type for the expected mode of operation
Forward Multiplexing Capability	'000'	No multiplex
Backward Multiplexing Capability	'000'	No multiplex

B.2 ATM Signalling Required for RAST-1 A/V VC

The following tables define the signalling elements required to establish a bi-directional (RAST) audiovisual virtual circuit from a terminal. The QoS parameters shall be coded in accordance with the requirements of the network.

TABLE 8

AAL Parameters IE

IE Parameter	**Value**	**Notes**
AAL type	AAL-1	
Subtype	Video signal transport	
CBR rate	n x 64 kbit/s	
Multiplier	m	as negotiated by H.245
Source Clock Frequency Recovery		as negotiated by H.245
Error Correction Method		as negotiated by H.245

The Peak Cell Rate for the video service component is calculated using the MPEG-2 encoded rate plus AAL-1 overhead and specified as shown in Table 9. The ATM Traffic Descriptor includes only the user plane information rate for the service components in that one VC.

The video service component PCR may be specified using Cell Loss Priority, CLP=0+1 and CLP=0 or CLP=0+1.

TABLE 9

ATM Traffic Descriptor IE

IE Parameter	**Value**	**Notes**
Forward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 SPTS).
Backward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 SPTS).

TABLE 10

Broadband Bearer Capabilities IE

IE Parameter	**Value**	**Notes**
Bearer Class	BCOB-A	
Broadband Transfer Capability	Constant Bit Rate	
Timing Requirements	end-to-end timing required	
Susceptibility to clipping	susceptible to clipping	
User Plane Connection configuration	Point-to-Point	

TABLE 11

Broadband Low Layer IE

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0011' = H.310 RAST	Receive and Send Terminal
Forward Multiplexing Capability	'001'	Transport Stream
Backward Multiplexing Capability	'001'	Transport Stream

* NOTE - Program Stream is also an option for Forward and Reverse Multiplexing Capability.

The Generic Identifier Transport Information Element is a generic parameter that indicates the correspondence of the VC to a certain previously established request carried outside ATM signalling. Table 12 identifies the parameters of the Generic Identifier Transport IE for use in an H.310 environment.

TABLE 12

Generic Identifier Transport IE

IE Parameter	**Value**	**Notes**
Identifier Related Standard	'00000010'	H.310
Resource	Identifier for the virtual circuit	H.310 Correlation ID

B.3 ATM Signalling Required for H.310 SOT/ROT-1

The following tables define the signalling elements required to establish an audiovisual virtual circuit for a ROT or SOT terminal.

The QoS parameters shall be coded in accordance with the requirements of the network.

TABLE 13

AAL Parameters IE

IE Parameter	**Value**	**Notes**
AAL type	AAL-1	
Subtype	Video signal transport	
CBR rate	n x 64 kbit/s	
Multiplier	m	as negotiated by H.245
Source Clock Frequency Recovery		as negotiated by H.245
Error Correction Method		as negotiated by H.245

The Peak Cell Rate for the video service component is calculated using the MPEG-2 encoded rate plus AAL-1 overhead and specified as shown in Table 14. The ATM Traffic Descriptor includes only the user plane information rate for the service components in that one VC.

The video service component PCR may be specified using Cell Loss Priority, CLP=0+1 and CLP=0 or CLP=0+1.

TABLE 14

ATM Traffic Descriptor IE

The Forward and Backwards Peak Cell Rates are set according to which terminal sends the original SETUP message.

IE Parameter	**Value**	**Notes**
Forward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component MPEG-2 SPTS).
Backward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 SPTS).

TABLE 15

Broadband Bearer Capabilities IE

IE Parameter	**Value**	**Notes**
	BCOB-A	
Broadband Transfer Capability	Constant Bit Rate	
Timing Requirements	end-to-end timing required	
Susceptibility to clipping	susceptible to clipping	
User Plane Connection configuration	Point-to-Point	

TABLE 16

Broadband Low Layer IE

The Forward and Backwards Multiplexing Capability parameters are set according to which terminal sends the original SETUP message.

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0010' = H.310 ROT '0001' = H.310 SOT	-
Forward Multiplexing Capability*	If ROT then '000' = No multiplex If SOT then '001' = Transport Stream	
Backward Multiplexing Capability*	If ROT then '001' = Transport Stream If SOT then '000' = No multiplex	

* NOTE - Program Stream is also an option for Forward and Reverse Multiplexing Capability.

The Generic Identifier Transport Information Element is a generic parameter that indicates the correspondence of the VC to a certain previously established request carried outside ATM signalling. Table 17 identifies the parameters of the Generic Identifier Transport IE for use in an H.310 environment.

TABLE 17

Generic Identifier Transport IE

IE Parameter	**Value**	**Notes**
Identifier Related Standard	'00000010'	H.310
Resource	Identifier for the virtual circuit	H.310 Correlation ID

B.4 ATM Signalling Required for RAST-5 A/V VC

The following tables define the signalling elements required to establish a bi-directional (RAST) audiovisual virtual circuit from a terminal. The QoS parameters shall be coded in accordance with the requirements of the network.

TABLE 18

AAL Parameters IE

IE Parameter	**Value**	**Notes**
AAL type	AAL-5	
Forward Maximum AAL-5 CPCS SDU size	N*188 bytes	Default value for the video service component in this specification is 376 bytes. N is an integer.
Backward Maximum AAL-5 CPCS-SDU size	N*188 bytes	Default value for the video service component in this specification is 376 bytes. N is an integer.
SSCS Type	Null	

The Peak Cell Rate for the video service component is calculated using the MPEG-2 encoded rate plus AAL-5 overhead and specified as shown in Table 19. The ATM Traffic Descriptor includes only the user plane information rate for the service components in that one VC.

The video service component PCR may be specified using Cell Loss Priority, LP=0+1 and CLP=0 or CLP=0+1.

TABLE 19

ATM Traffic Descriptor IE

IE Parameter	**Value**	**Notes**
Forward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 SPTS).
Backward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 SPTS).

TABLE 20

Broadband Bearer Capabilities IE

IE Parameter	**Value**	**Notes**
Bearer Class	BCOB-X	
Broadband Transfer Capability	Constant Bit Rate	
User Plane Connection configuration	Point-to-Point	

TABLE 21

Broadband Low Layer IE

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0011' = H.310 RAST	Receive and Send Terminal
Forward Multiplexing Capability	'001'	Transport Stream
Backward Multiplexing Capability	'001'	Transport Stream

* NOTE - Program Stream is also an option for Forward and Reverse Multiplexing Capability.

The Generic Identifier Transport Information Element is a generic parameter that indicates the correspondence of the VC to a certain previously established request carried outside ATM signalling. Table 22 identifies the parameters of the Generic Identifier Transport IE for use in an H.310 environment.

TABLE 22

Generic Identifier Transport IE

IE Parameter	**Value**	**Notes**
Identifier Related Standard	'00000010'	H.310
Resource	Identifier for the virtual circuit	H.310 Correlation ID

B.5 ATM Signalling Required for H.310 SOT-5

For an H.310 SOT terminal, the Forward and Backwards Maximum AAL-5 CPCS SDU sizes are set according to which terminal sends the original SETUP message. The following table shows the settings if the SOT terminal sends the SETUP message.

TABLE 23

AAL type IE

IE Parameter	**Value**	**Notes**
AAL type	AAL-5	
Forward Maximum AAL-5 CPCS SDU size	N*188 bytes	Default value for the video service component in this specification is 376 bytes. N is an integer.
Backward Maximum AAL-5 CPCS-SDU size	0 bytes if the video service component is unidirectional, otherwise implementation specific.	
SSCS Type	'00000000'	Null

The Peak Cell Rate for the video service component is calculated using the MPEG-2 encoded rate plus the AAL-5 overhead. The ATM Traffic Descriptor includes only the user plane information rate for the service components in that one VC.

The video service component PCR may be specified using Cell Lost Priority CLP=0+1 and CLP=0 or CLP=0+1.

The Forward and Backwards Peak Cell Rates are set according to which terminal sends the original SETUP message. The following table shows the settings if the SOT terminal sends the SETUP message.

TABLE 24

ATM Traffic Descriptor

IE Parameter	**Value**	**Notes**
Forward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 Single Program Transport Stream (SPTS))
Backward Peak Cell Rate	0 cells/sec if video service component is unidirectional, otherwise implementation specific	

TABLE 25

Broadband Bearer Capabilities IE

IE Parameter	**Value**	**Notes**
Bearer Class	BCOB-X	
Broadband Transfer Capability	Constant Bit Rate	
User Plane Connection configuration	Point-to-Point	

TABLE 26

Broadband Low Layer IE

IE Parameter	Value	Notes
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0010' = H.310 SOT	Send Only Terminal
Forward Multiplexing Capability	'001'	Transport Stream
Backward Multiplexing Capability	'000'	No multiplex

* NOTE - Program Stream is also an option for Forward Multiplexing Capability.

The QoS parameters shall be coded in accordance with the requirements of the network.

The Generic Identifier Transport Information Element is a generic parameter that indicates the correspondence of the VC to a certain previously established request carried outside ATM signalling. Table 27 identifies the parameters of the Generic Identifier Transport IE for use in an H.310 environment.

TABLE 27

Generic Identifier Transport IE

IE Parameter	Value	Notes
Identifier Related Standard	'00000010'	H.310
Resource	Identifier for the virtual circuit	H.310 Correlation ID

B.6 ATM Signalling Required for H.310 ROT-5

The Forward and Backwards Maximum AAL-5 CPCS SDU sizes are set according to which terminal sends the original SETUP message. The following table shows the settings if the ROT terminal sends the SETUP message.

TABLE 28

AAL Parameters IE

IE Parameter	**Value**	**Notes**
AAL type	AAL-5	
Forward Maximum AAL-5 CPCS SDU size	0 bytes if video service component is unidirectional; otherwise implementation specific	
Backward Maximum AAL-5 CPCS-SDU size	N*188 bytes	Default value for the video service component in this specification is 376 bytes. N is an integer.
SSCS Type	Null	

The Peak Cell Rate for the video service component is calculated using the MPEG-2 encoded rate plus the AAL-5 overhead. The ATM Traffic Descriptor includes only the user plane information rate for the service components in that one VC.

The video service component Peak Cell Rate may be specified using CLP=0+1 and CLP=0 or CLP=0+1.

The Forward and Backwards Peak Cell Rates are set according to which terminal sends the original SETUP message. The following table shows the settings if the ROT terminal sends the SETUP message.

TABLE 29

ATM Traffic Descriptor IE

IE Parameter	**Value**	**Notes**
Forward Peak Cell Rate	0 cells/sec if video service component is unidirectional, otherwise implementation specific	
Backward Peak Cell Rate	Implementation and program selection specific	Set to the Peak Cell Rate value required for the video service component (MPEG-2 Single Program Transport Stream)

TABLE 30

Broadband Bearer Capabilities IE

IE Parameter	**Value**	**Notes**
Bearer Class	BCOB-X	
Broadband Transfer Capability	Constant Bit Rate	
User Plane Connection configuration	Point-to-Point	

TABLE 31

Broadband Low Layer IE

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0001' = H.310 ROT	Receive Only Terminal
Forward Multiplexing Capability	'000'	No multiplex
Backward Multiplexing Capability	'001'	Transport Stream

* NOTE - Program Stream is also an option for Reverse Multiplexing Capability.

These QoS parameters shall be coded in accordance with the requirements of the network.

The Generic Identifier Transport Information Element is a generic parameter that indicates the correspondence of the VC to a certain previously established request carried outside ATM signalling. Table 32 identifies the parameters of the Generic Identifier Transport IE for use in an H.310 environment.

TABLE 32

Generic Transport Identifier IE

IE Parameter	**Value**	**Notes**
Identifier Related Standard	'00000010'	H.310
Resource	Identifier for the virtual circuit	H.310 Correlation ID

[End Correction]

8.7.2 Technical and Editorial Corrections to ITU-T Recommendation H.222.1

8.7.2.1 None identified at this time.

8.7.3 Technical and Editorial Corrections to ITU-T Recommendation H.245

8.7.3.1 None Identified at this time.

8.8 Implementation Clarifications

8.8.1 Implementation Guidance for ITU-T Recommendation H.310

8.8.1.1 Use of Release 1 Q.2931 Signalling (February, 1995) by H.310 Terminals

The following appendix will be added to the ITU-T Recommendation H.310:

[Begin Correction]

APPENDIX II

(This appendix does not form an integral part of this Recommendation)

Use of Release 1 Q.2931 (February, 1995) by H.310 Terminals

When using the latest Q.2931 changes, the Terminal Protocol , etc. is conveyed in the Broadband Low Layer Information Element, while the H.310 Correlation ID is carried by the Generic Identifier Transport Information Element. Neither of these are supported by Release 1 of Q.2931 (February, 1995).

When using Release 1 of Q.2931, the signalling parameters should be set in accordance with Annex B of the H.310 Recommendation (ATM Signalling SETUP Information Elements) with the following accommodations for and extensions to Release 1 of Q.2931.

- Set the Broadband Low Layer Information element with the new Q.2931 code points for H.310 terminal protocol and multiplexing scheme parameters.

IE Parameter	**Value**	**Notes**
User Information Layer 3 Protocol Field	'01100'	ITU Recommendation H.310
Terminal Type	'0001' = H.310 ROT '0010' = H.310 SOT '0011' = H.310 RAST	Receive Only Terminal Send Only Terminal Send and Receive
Forward Multiplexing Capability	'000' '001' '011'	No multiplex Transport Stream Program Stream
Backward Multiplexing Capability	'000' '001' '011'	No multiplex Transport Stream Program Stream

- Instead of the Generic Identifier Transport Information Element, use the B-HLI to convey the H.310 Correlation ID by setting the High Layer information type to "user specific"

B-HLI IE Parameter	**Value**	**Notes**
High Layer Information Type	'0000001'	User-specific
Octets 6 and 7	2-byte ID	H.245 Resource ID

- For the Quality of Service Parameter, use QoS Class in accordance with what your network provider has defined for carrying interactive multimedia services.

These accommodations are provisional, the Generic Identifier Transport Information Element should be used to convey the H.310 correlation ID when the relevant Q.2931 specifications have been implemented.

[End Correction]

8.8.1.2 Handling of Simultaneous Calls when Setting Up an H.245 Control Channel

Currently in H.310 or H.245, if two H.310 terminals call each other at exactly the same time to establish their control connection, no mechanism exists to detect this situation and to drop one of the connections.

ATM Q.2931 will allow two simultaneous calls between the same two terminals. Q.2931 believes that it is up to a higher layer protocol to detect "duplicate" connections and leaves it up to the ATM application to decide what to do about it (which call to drop, etc.).

Since H.245 is not involved in the actual ATM connection establishment between the terminals, this problem needs to be solved at the H.310 level.

The following appendix will be added to the ITU-T Recommendation H.310:

[Begin Correction]

APPENDIX III

(This appendix does not form an integral part of this Recommendation)

Handling of Simultaneous Calls when Setting Up an H.245 Control Channel

When H.310 terminals call each other at exactly the same time to establish their control connection (i.e. when an H.310 terminal detects simultaneous calls (call collisions)), the H.310 terminals will choose the call being setup by (originating from) the terminal with the higher ATM address as the winner. The other call will be dropped.

[End Correction]

H.310 Recommendation Series Defect Report Form

DATE:

CONTACT INFORMATION

NAME:
COMPANY:
ADDRESS:

TEL:
FAX:
E-MAIL:

AFFECTED RECOMMENDATIONS:

DESCRIPTION OF PROBLEM:

SUGGESTIONS FOR RESOLUTION:

NOTE - Attach additional pages if more space is required than is provided above

9 Implementor's Guide for H.323, H.225.0 and H.245

This document provides a text for correction and clarifications to H.323 (11/96), H.225.0 (11/96) and H.245 (3/96) which have been produced based on the discussions at three interregnum Rapporteurs meetings and discussions during the March 1997 SG 16 meeting. SG 16 is requested to review and determine it.

Editor: Jim Toga
Intel
+1 503 264 8816 (voice)
+1 503 264 3485 (fax)
jtoga@ibeam.intel.com

ABSTRACT: This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T H.323 (H.225.0 and H.245)-series Recommendations. It is intended to be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.323-series Recommendations.

Contact Information

ITU-T Study Group 16 / Question 13 Rapporteur	Dale Skran Lucent Technologies Rm. 2G432 200 Laurel Avenue Middletown, NJ 07748 USA	Tel: Fax: email:	+1.908.957.5988 +1.908.957.5627 dls@mtgbcs.lucent.com
Implementers Guide Editor	James Toga Intel Corporation 2111 NE 25th Avenue JF3-212 Hillsboro, OR. 97124-5961 USA	Tel: Fax: email:	+1.503.264.8816 +1.503.264.3485 jtoga@ibeam.intel.com
ITU-T Recommendation H.323 Editor	Gary Thom Delta Information Systems 300 Welsh Road, Bldg 3 Horsham, PA. 19044-2273 USA	Tel: Fax: email:	+1.215.657.5270 +1.215.657.5273 gthom@delta-info.com
ITU-T Recommendation H.225.0 Editor	Glen Freundlich Lucent Technologies 11900 N. Pecos St. Westminster, CO 80234 USA	Tel: Fax: email:	+1.303.538.2899 ggf@dr.lucent.com
ITU-T Recommendation H.245 Editor	Mark Reid PictureTel Corporation 222 Rosewood Drive M/S 634 Danvers, MA. 01923 USA	Tel: Fax: email:	+1.508.2925863 +1.508.749.2805 reid@pictel.com

Document History

Revision ~~Version~~	Date	Description
A~~1~~	27 September 1996	Initial version - completed at the ITU-T Study Group 15 Rapporteurs meeting, Eibsee Germany September 1996
B~~2~~	20 December 1996	Revised version - completed at the ITU-T Study Group 16 Rapporteurs meeting, Boulder Colorado December 1996
C	20 March 1997	Final version - completed at the ITU-T Study Group 16 meeting, Geneva Switzerland March 1997

9.1 Introduction

This document is a compilation of reported defects identified with the 1993-1996 editions of the ITU-T H.323-series Recommendations. It is intended to be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.323-series Recommendations.

The first version of the Guide was produced following the April 1997 ITU-T Study Group 15 meeting. Wide distribution of this document is expected and encouraged.

9.2 Scope

This Guide resolves defects in the following categories:

- editorial errors;
- technical errors such as omissions or inconsistencies;
- ambiguities.

In addition the 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 the normal way through contributions to the ITU-T.

9.3 Policies for Updating This Document

This document is managed by the ITU-T Study Group 15 Question 3 Rapporteur’s group. It can be revised at any recognized Q.3/15 Rapporteur's group meeting provided the proposed revisions are unanimously accepted by the members of the group. A revision history cataloguing the evolution of this document is included.

9.4 Defect Resolution Procedure

Upon discovering technical defects with any components of the H.323 Recommendations series, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q.3/15 Rapporteur. The template for a defect report is enclosed. Contact information for these parties is included in this 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.323-series Recommendations. Formal membership in the ITU is not required to participate in this process.

9.5 References

This document refers to the following H.323-series Recommendations:

- ITU-T Recommendation H.323 (1996), *Visual Telephone Systems and Equipment for Local Area Networks which provide a Non-Guaranteed Quality of Service*
- ITU-T Recommendation H.225.0 (1996), *Media Stream Packetization and Synchronization on Non-Guaranteed Quality of Service LANs*
- ITU-T Recommendation H.245 (1996), *Control Protocol for Multimedia Communication*

9.6 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.

9.7 Technical and Editorial Corrections

9.7.1 Technical and Editorial Corrections to ITU-T Recommendation H.323

9.7.1.1 RTP stream and Logical Channel Association

Description: An omission has been discovered in the H.323 specification concerning the correlation of multimedia streams and their sources within a multipoint conference.

This information will be contained in the revision 2 of H.323 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final H.323 document that was submitted for approval in 1996.

As this omission affects media stream correlation and presentation to endpoint users, failure to correct these errors would result in an incompatible implementation. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

Method to associate a logical channel with the RTP stream:

1. The sender sends the **OpenLogicalChannel** message to the MC. The MC shall add the **TerminalLabel** of the sender to the **destination** field in **H2250LogicalChannelParameters** before forwarding the message to the receiver. In the muli-unicast model, the sender shall enter the **TerminalLabel** of the receiver in the **destination** field and the MC shall replace the **destination** field with the **TerminalLabel** of the sender before forwarding the **OpenLogicalChannel** message to the receiver.

2. The receiver may associate the logical channel number with the RTP stream source by comparing the **TerminalLabel** obtained from the **OpenLogicalChannel** message with the lowest byte in the **SSRC** (H.323, Section 8.4.3.1, #A4c).

Restrictions:

This method allows the receiver to associate only one logical channel of the sender with the RTP stream within a RTP session. If the sender wants to send two streams (video 1 and video 2) in the same session then the receiver will not be able to map the two logical channels with their two corresponding streams. In practice this is not a real restriction because the sender will normally send one stream per session but this method does disallow multiple streams per session.

To allow for complete solution in revision 2 of H.323 the following correction will be followed

[Begin Correction]

8.3.4 Correlation of Media Streams in Multipoint Conferences

The following method shall be used to associate a logical channel with an RTP stream within a multipoint conference. The media stream source endpoint sends the **openLogicalChannel** message to the MC. The source endpoint shall place its **SSRC** value in the **H2250LogicalChannelParameters**. The MC shall add **TerminalLabel** of the source endpoint to the **destination** field in **H2250LogicalChannelParameters** before forwarding the message to the destination endpoint. In the multi-unicast model, the source endpoint shall enter the **TerminalLabel** of the receiving terminal in the **destination** field and the MC shall replace the **destination** field with the **TerminalLabel** of the source endpoint before forwarding the **OpenlogicalChannel** message to the destination endpoint.

The destination endpoint may associate the logical channel number with the RTP stream source, by comparing the **SSRC** obtained from the **OpenLogicalChannel** message with the **SSRC** in the RTP header. If the SSRC field is not present in the OpenLogicalChannel the receiver may associate the logical channel number with the RTP stream source by comparing the **TerminalLabel** obtained from the **OpenLogicalChannel destination** field with the lowest byte in the **SSRC** (H.323, Section 8.4.3.1, #A4c).

[End Correction]

9.7.1.2 Communication Mode Command Procedures

Description: An omission has been discovered in the H.323 specification concerning the usage of the **CommunicationModeCommand** command and its interpretation by the receiving endpoints.

This information will be contained in the revision 2 of H.323 Recommendation to be published by the ITU-T. However, this information is missing in the final H.323 document that was submitted for approval in 1996.

As this omission affects multipoint conferencing and MC(U)-endpoint coordination, failure to correct these errors would result in possibly incompatible implementations. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

There are a number of procedures associated with the **CommunicationModeCommand** which need to be documented in H.323. This section should be inserted as 8.4.4 and entitled **CommunicationModeCommand** Procedures.

[Begin Addition]

8.4.4 Communication Mode Command Procedures

The H.245 **CommunicationModeCommand** is sent by a H.323 MC to specify the communication mode for each media type: unicast or multicast. This command may cause a switch between a centralized and decentralized conference and therefore may involve closing all existing logical channels and opening new ones.

The **CommunicationModeCommand** specifies all the current sessions in the conference. For each session, the following data is specified: the RTP session identifier, the associated RTP session ID if applicable, a terminal label if applicable, a description of the session, the datatype of the sessions (e.g. G.711), and a unicast or multicast address for the media and media control channels as appropriate for the conference configuration and type.

The **CommunicationModeCommand** conveys the transmit modes which conference endpoints shall use in a conference. The command does not convey receive modes, as they are specified by **OpenLogicalChannel** commands which are sent from the MC to the endpoints.

It is presumed that the **CommunicationModeCommand** is defining the modes of a conference and is therefore sent after the **multipointConference** indication which notifies an endpoint that it must comply with the commands of the MC. Endpoints should wait for a **CommunicationModeCommand** before opening logical channels when they have received a **multipointConference** indication.

Endpoints receiving a **CommunicationModeCommand** use the **terminalLabel** field of each table entry to determine if the entry is applicable for its own processing. Entries which do not contain a **terminalLabel** apply to all endpoints in the conference. Entries which contain **terminalLables** are commands to specific endpoints and which match the **terminalLabel** in the entry. For example, when audio streams from all endpoints are placed on one multicast address (one session), the table entry for the audio mode, media address, and media control address will not contain a **terminalLabel**. When the table entry commands an endpoint to send its video to a multicast address, the MC will include that endpoint's **terminalLabel**.

8.4.4.1 Sample MC to Terminal Communication Mode Command

Sample Conference Scenario:

Endpoints A, B and C are in an audio and video distributed conference using multicast. The MC (which could be anyone of the nodes), has decided to place the media and media control channels on the following multicast addresses:

Stream	Multicast Address
Audio for all endpoints:	MCA1
Audio Control for all endpoints:	MCA2
Video from endpoint A:	MCA3
Video Control data about endpoint A:	MCA4
Video from endpoint B:	MCA5
Video Control data about endpoint B:	MCA6
Video from endpoint C:	MCA7
Video Control data about endpoint C:	MCA8

CommunicationModeTable sent to all Endpoints

All entries are commands for endpoints to open a logical channels for transmission. terminalLabel is only present when the entry is not specific to a single endpoint in the conference.

ENTRY ONE - AUDIO & AUDIO CONTROL FOR CONFERENCE

sessionID	1
sessionDescription	Audio
dataType	Audio Capability
mediaChannel	MCA1
mediaControlChannel	MCA2

ENTRY TWO - VIDEO FOR NODE A	
sessionID	2
associatedSessionID	1
terminalLabel	M/T for A
sessionDescription	Video for Node A
dataType	Video Capability
mediaChannel	MCA3
mediaControlChannel	MCA4

ENTRY THREE - VIDEO FOR NODE B	
sessionID	3
associatedSessionID	1
terminalLabel	M/T for B
sessionDescription	Video for Node B
dataType	Video Capability
mediaChannel	MCA5
mediaControlChannel	MCA6

ENTRY FOUR - VIDEO FOR NODE C	
sessionID	4
associatedSessionID	1
terminalLabel	M/T for C
sessionDescription	Video for Node C
dataType	Video Capability
mediaChannel	MCA7
mediaControlChannel	MCA8

[End Addition]

[Begin Correction]

8.3.3 Media Stream Address Distribution

...

In multicast, the multicast addresses are assigned by the MC and distributed to the endpoints in the **communicationModeCommand**. It is the responsibility of the MC to allocate and assign unique multicast addresses. The endpoint shall signal an **openLogicalChannel** to the MC with the assigned multicast address. The MC shall forward the **openLogicalChannel** to each receiving endpoint. In cases where media from multiple endpoints are transmitted on a single session (e.g. single multicast address), the MC shall open a logical channel to each endpoint receiving media from an endpoint in the conference.

In cases where an endpoint joins a conference after the initial **communicationModeCommand** has been transmitted, it is the responsibility of the MC to send an updated **communicationModeCommand** to the new endpoint and to open the appropriate logical channels for media sourced from the new endpoint. In cases where an endpoint leaves a conference after the initial **communicationModeCommand** has been transmitted, it is the responsibly of the MC to close the appropriate logical channels which were being sourced from the endpoint which left the conference.

...

[End correction]

9.7.2 Technical and Editorial Corrections to ITU-T Recommendation H.225.0

9.7.2.1 Coding the Facility IE

Description: An error exists in the documentation for encoding the Facility IE.

These encoding instructions will be printed in their corrected form in the H.225.0 revision 2 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final H225.0 document that was submitted for approval in 1996.

As these errors appear in the encoding instructions, failure to correct these errors would result in an incompatible implementations. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

The approach documented in H.225.0 for signalling a call deflection is not only erroneous but also unnecessarily complicated: a single value for a the Facility is defined that has to be encoded in order to signal to the recipient that the call is redirected to another destination and that further information is to be found in the Facility-UUIE part of the message. The same result can be achieved by sending a well-defined constant that cannot be accidentally misinterpreted to be a ROSE message. The approach is to send an empty FacilityIE to indicate the presence of Facility-UUIE that contains the actual information to be considered by the recipient. The following paragraphs contain the necessary modifications to H.225.0.

Also included in this section is some clarifying instructions as to the use of the Facility PDU in the endpoint initiated call forwarding scenario.

Corrections are required at three places in H.225.0.

9.7.2.1.1

[Begin Correction]

7.2.2.13 Facility

~~Encoded following Figure 8-2/Q.932 and Table 8-5/Q.932.~~

In order to signal call redirection specific to H.323 procedures (call forwarding, redirecting a call to the MC, or forcing a call to be routed to the gatekeeper), the User-to-User Information Element of the Facility is used. This particular case shall be indicated by coding a Facility IE of length zero. I.e. the Facility Information Element shall consist of exactly 2 octets as follows:

- Octet #1 (information element identifier) shall be set to ‘00011100’ (1CH) to indicate the Facility IE
- Octet #2 (information element length) shall be set to ‘0’ to indicate that no further octets belonging to this information element follow.

Any Facility IE that is used to indicate unmodified semantics as defined in Q.932 shall be encoded following Figure 8-2/Q.932 and Table 8-5/Q.932. In this case, ~~T~~the Facility IE PDU shall be formed according to ROSE (uses X.208 [Specification of ASN.1] and X.209 [Specification of basic encoding rules for ASN.1]) as defined in Q.932 and Q.952. Then the Facility IE shall consist of at least 8 octets as specified in Q.932.

The use of other Facility IEs as defined by Q.932 is for further study.

~~For the call forwarding case, the ROSE invoke component shall be completed as follows:~~

~~invokeIdentifier = sequence number~~
~~operationValue = callRerouting~~
~~argument =~~
~~{~~
~~reroutingReason = cd // cd = call deflection~~
~~calledAddress = forwarded-to E164 // get alias from Facility-UUIE~~
~~reroutingCounter = x~~
~~}~~

~~The **Facility-UUIE** will be encapsulated within the ROSE PDU as defined in Q.932. If the forwarded-to endpoint cannot be specified with an E.164 address, the forwarding terminal shall supply either the **alternativeAddress** or **alternativeAliasAddress.**~~

~~In cases unique to H.323 (i.e., the FacilityReason codes found under the **Facility-UUIE** description found in H.225.0), the *reroutingReason* will be **cd** and the *reason* of the **Facility-UUIE** will contain the actual reason for the deflection. This means that the receiver of the Facility message must always check the **Facility-UUIE** reason.~~

In order to indicate call forwarding, the Facility IE shall be empty and the **Facility-UUIE** shall indicate in the **alternativeAddress** or the **alternativeAliasAddress** the terminal to which the call is to be redirected. In this case, the **facilityReason** shall be set to **callForwarded**.

To instruct an endpoint to call a different endpoint because the calling endpoint wishes to join a conference and the called endpoint does not have the MC, the Facility IE would be left empty as well~~completed in nearly the same manner as for forwarding~~. The **conferenceID** shall indicate the conference to join and the reason in the **Facility-UUIE** shall be **routeCallToMC**.

Also, ~~T~~to instruct the calling endpoint to signal the called endpoint through the called endpoint’s gatekeeper, the Facility IE is left empty~~would be completed in nearly the same manner as for forwarding~~. The **conferenceID** in the Facility-UUIE shall indicate the conference to join and the reason in the **Facility-UUIE** shall be **routeCallToGatekeeper.**

Possible extensions for H.225.0 are for further study.

[End Correction]

9.7.2.1.2

[Begin Correction]

7.4.1 Facility

The FACILITY message may be used to request or acknowledge a supplementary service. It shall be used to provide information on where a call should be directed as part of call transfer or a terminal indicating that the incoming call must go through a gatekeeper.

In order to signal call redirection specific to H.323 procedures, the User-to-User Information Element of the Facility is used. This particular case shall be indicated by coding a Facility IE of length zero. In this case, the Facility Information Element shall consist of exactly 2 octets.

If a Facility IE carrying semantics of Q.932 and encoded as defined in Q.932 and Q.952 is present, it shall consist of at least 8 octets as required by Table 7-2/Q.932. The use of Facility IEs of that type is for further study.

An H.323 entity shall handle the empty (H.323-specific) Facility IE properly and shall be capable of skipping other Facility IEs that it does not understand.

Follow Table 7-2/Q.932. The following modifications apply:

TABLE 12/H.225.0

Facility

Information element	**H.225.0 status(M/F/O)**	**Length in H.225.0**
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Facility	M	2 or 8-*
Display	O	2-82
User-to-User	M	2-131

```
Facility-UUIE ::= SEQUENCE
{
        protocolIdentifier ProtocolIdentifier,
        alternativeAddress          TransportAddress OPTIONAL,
        alternativeAliasAddress     SEQUENCE OF AliasAddress OPTIONAL,
        conferenceID                ConferenceIdentifier OPTIONAL,
        reason                      FacilityReason,
        ...
}

        FacilityReason              ::=CHOICE
        {
                routeCallToGatekeeper   NULL,           -- call must use gatekeeper model
                                                        -- gatekeeper is alternativeAddress
                callForwarded           NULL,
                routeCallToMC           NULL,
                undefinedReason NULL,
                ...
        }
```

protocolIdentifier - set by the calling endpoint to the version of H.225 supported
alternativeAddress - this is a specific transport address to which the calling party should direct the call; if present **alternativeAliasAddress** is not needed.
alternativeAliasAddress - contains aliases that can be used to re-direct the call; if an alias is provided **alternativeAddress** is not needed.
conferenceID - unique conference identifier
reason - more information about the facility message.

[End Correction]

9.7.2.1.3

[Begin Correction]

7.3.10 Release Complete

This message shall be sent by a terminal to indicate release of the call if the reliable call signalling channel is open. Afterwards, the call reference value (CRV) is available for reuse.

The disconnect/release/release complete sequence is not used since the only added value is that a network-to-user information element can be appended to the release message. As this does not apply to the LAN environment, the single step method of sending only Release Complete is used.

Follow Table 3-11/Q.931. The following modifications apply:

TABLE 10/H.225.0

Release Complete

Information element	**H.225.0 status(M/F/O)**	**Length in H.225.0**
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Cause	CM (NOTE 1)	1
Display	O	2-82
Signal	O	2-3
User-to-User	M	2-131

NOTE 1 - Either the Cause IE or the **ReleaseCompleteReason** shall be present.

If this message is sent in response to a Facility message with an empty Facility IE, the ReleaseCompleteReason shall be set to **facilityCallDeflection**.

If this message is forwarded from a SCN by a gateway the cause value shall be set as specified in Q.931.

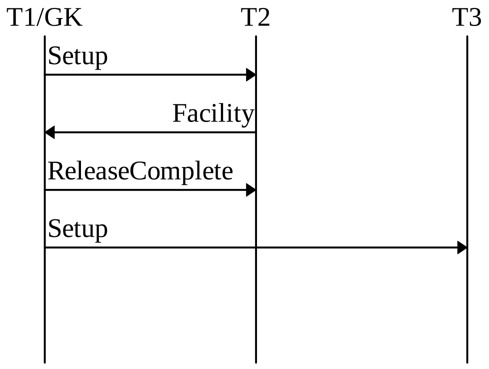
```
ReleaseComplete-UUIE ::= SEQUENCE
{
        protocolIdentifier   ProtocolIdentifier,
        reason                       ReleaseCompleteReason OPTIONAL,
        ...
}

ReleaseCompleteReason            ::=CHOICE
{
        noBandwidth                  NULL,              -- bandwidth taken away or ARQ denied
        gatekeeperResources          NULL,              -- exhausted
        unreachableDestination       NULL,              -- no transport path to the destination
        destinationRejection         NULL,              -- rejected at destination
        invalidRevision              NULL,
        noPermission                 NULL,              -- called party's gatekeeper rejects
        unreachableGatekeeper        NULL,              -- terminal cannot reach gatekeeper for
        ARQ                 gatewayResources            NULL,
        badFormatAddress             NULL,
        adaptiveBusy                 NULL,              --  call is dropping due to LAN crowding
        inConf                       NULL,              --  no address in AlternativeAddress
        undefinedReason              NULL,
        ...
        facilityCallDeflection       NULL,              - call was deflected using a Facility message
}
```

[End Correction]

9.7.2.1.3.1 Clarification of Release Complete

Consider the scenario where a terminal (T2) receives a Setup message, then instructs the originator of the Setup (T1 or GK) to forward the call to another terminal (T3). The message sequence appears as:



T2 responds to the Setup message with a Facility message. The Facility message contains an empty Facility IE.

The ReleaseComplete message would have the releaseCompleteReason in the ReleaseComplete-UUIE set to indicate facilityCallDflection.

A calling terminal may receive multiple Facility messages indicating *call forwarding* when trying to place a single call. For each call, the calling terminal should keep track of all the addresses which have been contacted due to call forwarding in order to avoid forwarding loops.

9.7.2.2 Setup message

Description: An error (missing value) in the defined fields of the **SETUP** message has been discovered.

The corrected **SETUP** structure will be entered in the corrected form in the H.225.0 revision 2 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final H225.0 document that was submitted for approval in 1996.

The receipt of a **SETUP** message will cause a corresponding **ARQ** message to be send to the receiver's Gatekeeper. The will exist situations in which the calling endpoint does not have any **AliasAddresses** assigned to it. In these cases, the only identifier for the caller is the **CallSignallingAddress**. Assuming this value is present, it should be taken from the **SETUP** message and placed in the **srcCallSignalAddress** so that a Gatekeeper can utilize it in any policy implementation for an **ARQ**/**ACF** response.

An additional missing value has been discovered which will disallow the calling to occur between two H.323 endpoints on separate LANS. In order to address an endpoint the calling endpoint may supply an 'extension address' for the far side of the intermediate Gateways.

[Begin Correction]

In the User-user field the following information shall be provided in the ASN.1 User-to-User IE:

```
Setup-UUIE                  ::=SEQUENCE
{
        protocolIdentifier ProtocolIdentifier,
        h245Address         TransportAddress OPTIONAL,
        sourceAddress       SEQUENCE OF AliasAddress OPTIONAL,
        sourceInfo          EndpointType,
        destinationAddress  SEQUENCE OF AliasAddress OPTIONAL,
        destCallSignalAddress TransportAddress OPTIONAL,            -- NOTE 1
        destExtraCallInfo   SEQUENCE OF AliasAddress OPTIONAL,      -- NOTE 1
        destExtraCRV        SEQUENCE OF CallReferenceValue OPTIONAL,-- NOTE 1
        activeMC            BOOLEAN,
        conferenceID        ConferenceIdentifier,
        conferenceGoal      CHOICE
        {
                create      NULL,
                join        NULL,
                invite      NULL,
                ...
        },
        callServices                QseriesOptions  OPTIONAL,
        callType              CallType,
        ...,
        sourceCallSignalAddress   TransportAddress OPTIONAL,
remoteExtensionAddress   AliasAddress OPTIONAL
}
```

NOTE 1 - If the **destExtraCallInfo** is present, a CRV for each call to be made may be supplied in **destExtraCRV**. These CRVs will be used to identify any response to each call launched. These procedures are for further study. If the destExtraCRV field is not present, a gateway shall aggregate all call information into ~~into~~ a single response, with the effect that if one call fails on the SCN side, the entire call is treated as a failure.

ProtocolIdentifier - set by the calling endpoint to the version of H.323 supported

...

sourceCallSignalAddress - contains the transport address for the source; this value shall be used in the ARQ message by the receiver of the SETUP**.** In all cases where the information is available to the sender of the SETUP message, this field shall be filled in.
remoteExtensionAddress - contains the alias address of a called endpoint in cases where this information is needed to traverse multiple Gateways. In all cases where the information is available to the sender of the SETUP message, this field shall be filled in.

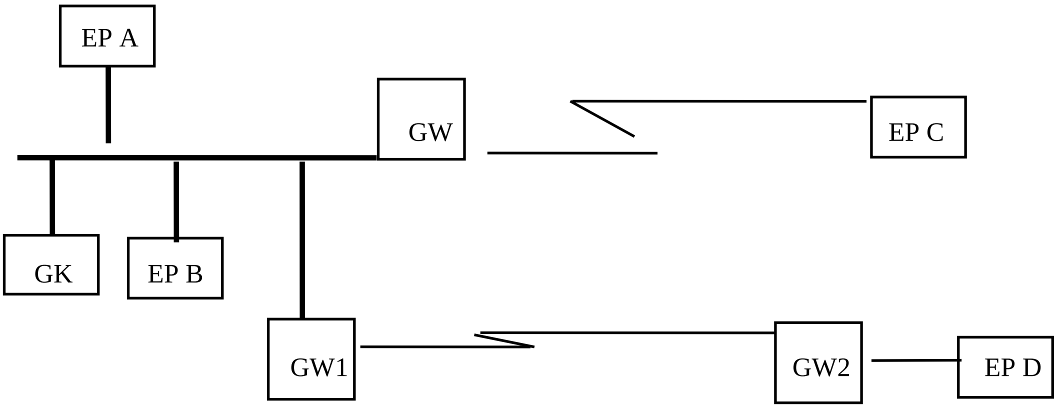
There are a number of dialing scenarios which are considered in the H.323 specification. The following list shows a subset of H.323 call types and the following table shows where each field shall be placed for each call.

There are two premises that are assumed:

1. The calling endpoint must know the type of call that it is making. What this means, is that it is able to signal the difference between an H.320, H.324 or POTS call when supplying an E.164 address. In addition, the endpoint will have to signal the desire to call out through an H.323 proxy (H.323-H.323 gateway).
2. There is a local implementation signalling element that is carried in-band with the address. In other words an arbitrary *prefix* may be added to the E.164 address. For example an “8xxxxx” indicates an H.324 call while a “9xxxx” indicates a H.320 call. This signalling element is what the GK may use to figure out which GW IP address to hand back in the ARQ. Optimally this is configurable at both the GW and the GK.

The following picture will be used to illustrate the subsequent tables.

The following list shows a subset of H.323 example call types and the table shows where each field should be placed for each call.



The following list shows a subset of H.323 example call types and the table shows where each field should be placed for each call.

Calling on the LAN EPA calls EPB

Msg	Field	Value	notes
ARQ	destinationAddress	alias(s) of EPB known by EPA	*sent to GK*
	destExtraCallInfo	NA	
ACF	destCallSignalAddress	EBP's IP *OR* GK IP if GK routed	*sent from GK*
Setup	Called Party Number	First E.164 of EPB alias if known by EPA([2])	*sent to ACF destCallSignalAddress*
	destinationAddress	any other alias(s) of EPB known by EPA	
	destCallSignalAddress	*from ACF*	
	destExtraCallInfo	NA	

Calling Circuit EP (H320, H.324, POTS) via a Gateway EPA calls EPC

Msg	Field	Value	notes
ARQ	destinationAddress	First E.164 with call prefix	*sent to G*
	destExtraCallInfo	all other E.164 with prefixes	
ACF	destCallSignalAddress	GW's IP ***OR*** GK IP if GK routed	*sent from GK*
Setup	Called Party Number	First E.164 with call prefix	*sent to ACF destCallSignalAddress*

[2] Note that the numbering plan value shall be set to '1001' (Private) as specified in H.225.0 section 7.2.2.6

	subAddress	<used if required[3]>	
	destinationAddress	NA	
	destCallSignalAddress	*from ACF*	
	destExtraCallInfo	all other E.164 with prefixes	

<u>Calling a LAN endpoint via two Gateways   EPA calls EPD</u>

(In other words - passing H.323 through H.320)

Msg	Field	Value	notes
ARQ	destinationAddress	First E.164 with call prefix (e.g. GW2's #)	*sent to GK*
	destExtraCallInfo	all other E.164 with prefixes (e.g. GW2's #s)	
ACF	destCallSignalAddress	GW1's IP ***OR*** GK IP if GK routed	*sent from GK*
Setup	Called Party Number	First E.164 with call prefix (e.g. GW2's #)	*sent to ACF destCallSignalAddress*
	subAddress	<used if required[4]>	
	destinationAddress	NA	
	destCallSignalAddress	*from ACF*	
	destExtraCallInfo	all other E.164 with prefixes (e.g. GW2's #s)	
	remoteExtensionAddress	alias of EPD known by EPA	

<u>Calling a LAN endpoint via two Proxies   EPA calls EPD</u>

(Proxy calls would have a prefix just as with any other gateway)

Msg	Field	Value	notes
ARQ	destinationAddress	alias(s) of EPD known by EPA with prefixes	*sent to GK*
	destExtraCallInfo	NA	
ACF	destCallSignalAddress	GW1's IP *OR* GK IP if GK routed	*sent from GK*

[3] Usage of subAddress is to signal the GW about far device which is on a real (physical) ISND sub-address. In most cases this field is redundant

[4] Usage of subAddress is to signal the GW about far device which is on a real (physical) ISDN sub-address. In most cases this field is redundant

Setup	Called Party Number	NA	*sent to ACF destCallSignalAddress*
	subAddress	NA	
	destinationAddress	alias(s) of GW2 known by EPA with prefixes	([5])
	destCallSignalAddress	*from ACF OR Known IP address of GW2 (proxy)*	([6])
	destExtraCallInfo	NA	
	remoteExtensionAddress	alias of EPD known by EPA	([7])

9.7.2.3 RAS Data Definitions

Description: A typographical error has been discovered in the defined values used in RAS messages.

The corrected value definitions will be entered in the corrected form in the H.225.0 revision 2 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final H225.0 document that was submitted for approval in May 1996.

The id values defined contain fixed length strings. This is incorrect as there is no manner in which the end of the string can be determined. The intent of these definitions in Section 7.6 was to contain length encoded values, as is used in all other cases.

[Begin Correction]

GatekeeperIdentifier ::= BMPString (SIZE(1..128))

EndpointIdentifier ::= BMPString (SIZE(1..128))

[End Correction]

[5] This is essentially a ‘routable’ alias in that there is some mechanism to map from the alias to the GW2 IP address (e.g. DNS). It may also include the actual endpoint alias (such as endpoint@proxy) as long as it is parse-able by the gateway. When parsing this field, GW1 should remove any references to itself before passing this to GW2 (i.e. GW1[endpoint@proxy] ➔ endpoint@proxy)

[6] If this address is not that of the local receiving proxy it should take precedence over destinationAddress. This would be the case when the terminal is not using a gatekeeper and knows the address of both proxies in line.

[7] This address is the H.323 alias of the endpoint on the ‘far side’ of the proxy. It should allow the far side proxy to eventually connect to the endpoint. It should map eventually to an IP address either via Gatekeeper resolution, DNS or some other means.

Description: A typographical error has been discovered in the defined values used in RAS messages when combined with the procedures as described in H.225.0 sections 7.11, 7.12 and 7.15.

Corrected ASN.1 text will be entered in the corrected form in the H.225.0 revision 2 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the final H225.0 document that was submitted for approval in May 1996.

The values ranges defined in ASN.1 for both BandWidth and CallReferenceValue are defined to be (1..4294967295) and (1..65535) respectively. This is incorrect as this encoding does not allow for a legal value of 0 to be passed. The intent of these definitions in Section 7.6 was to be used in structures defined in 7.12 and 7.15.

As used in 7.11 and 7.12 the Bandwidth field may be returned to an endpoint from a Gatekeeper with a value of 0. This would indicate that there is no bandwidth presently available. In section 7.15 the CallReferenceValue set to 0 indicates that the IRR request is to query about ALL calls.

[Begin Correction]

BandWidth ::= INTEGER (0~~1~~.. 4294967295) -- in 100s of bits
CallReferenceValue ::= INTEGER (0~~1~~..65535)

[End Correction]

9.7.3 Technical and Editorial Corrections to ITU-T Recommendation H.245

9.7.3.1 Correction of Errors UnicastAddress/Multicast Definitions

Description: Typographical omissions have been identified in the specifications for H.323 transport addresses as defined in H.245.

These omissions will be added in their corrected form in the H.245 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the H.245 (NCM6) document that was determined in May of 1996.

As these omissions appear in the H.225.0 specification, failure to correct these errors could result in an incompatible implementation. Transport addresses can be indicated in the H.225.0 signalling which have no counterpart in H.245 needed commands. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

[Begin Correction]

```
UnicastAddress          ::=CHOICE
{
        iPAddress       SEQUENCE
        {
                network OCTET STRING (SIZE(4)),
                tsapIdentifier  INTEGER(0..65535),
                ...
        },
        iPXAddress              SEQUENCE
        {
                node            OCTET STRING (SIZE(6)),
                netnum          OCTET STRING (SIZE(4)),
                tsapIdentifier          OCTET STRING (SIZE(2)),
                ...
        },
        iP6Address              SEQUENCE
        {
                network         OCTET STRING (SIZE(16)),
                tsapIdentifier          INTEGER(0..65535),
                ...
        },
        netBios                 OCTET STRING (SIZE(16)),
        iPSourceRouteAddress    SEQUENCE
        {
                routing         CHOICE
                {
                   strict       NULL,
                   loose        NULL
                },
                network         OCTET STRING (SIZE(4)),
                tsapIdentifier          INTEGER(0..65535),
                route           SEQUENCE OF OCTET STRING (SIZE(4)),
                ...
        },
        ...,
        nsap                    OCTET STRING (SIZE(1..20)),
        nonStandardAddress      NonStandardParameter
}

MulticastAddress                ::=CHOICE
{
        iPAddress               SEQUENCE
        {
                network         OCTET STRING (SIZE(4)),
                tsapIdentifier          INTEGER(0..65535),
                ...
        },
        iP6Address              SEQUENCE
        {
                network         OCTET STRING (SIZE(16)),
                tsapIdentifier          INTEGER(0..65535),
                ...
        },
        ...,
        nsap                    OCTET STRING (SIZE(1..20)),
        nonStandardAddress      NonStandardParameter

}
```

[End Correction]

9.7.3.2 Addition of OpenLogicalChannel Reject Definitions

Description: Typographical omissions have been identified in the specifications for H.323 logical channel operation as defined in H.245.

These omissions will be added in their corrected form in the H.245 Recommendation to be published by the ITU-T. However, this information appears incorrectly in the H.245 (NCM6) document that was determined in May of 1996.

As these omissions directly affect interoperability between endpoints, failure to correct these errors could result in an incompatible implementations. The listed error codepoint should be utilized by a master endpoint when it receives an OpenLogicalChannel command during the time it has an outstanding OpenLogicalChannel issued. For this reason, these corrections are highlighted in this document to ensure that implementers are aware that these corrections have been made.

[Begin Correction]

```
OpenLogicalChannelReject        ::=SEQUENCE
{
        forwardLogicalChannelNumber           LogicalChannelNumber,
        cause           CHOICE
        {
                unspecified   NULL,
                unsuitableReverseParameters     NULL,
                dataTypeNotSupported     NULL,
                dataTypeNotAvailable      NULL,
                unknownDataType NULL,
                dataTypeALCombinationNotSupported          NULL,
                ...,
                multicastChannelNotAllowed      NULL,
                insufficientBandwidth       NULL,
                separateStackEstablishmentFailed        NULL,
                invalidSessionID      NULL,
                masterSlaveConflict               NULL
        },
        ...
}
```

[End Correction]

9.7.3.2.1 Add entry to table in 7.3.3

[Begin Correction]

TABLE 7/H.245

Reasons for rejecting a OpenLogicalChannel

ASN.1 codepoint	Cause
unspecified	No cause for rejection specified.
UnsuitableReverseParameters	This shall only be used to reject a bi-directional logical channel request when the only reason for rejection is that the requested reverseLogicalChannelParameters are inappropriate. Such a rejection shall immediately be followed by initiating procedures to open a similar but acceptable bi-directional logical channel.

…

invalidSessionID	Attempt by slave to set SessionID when opening a logical channel to the master
masterSlaveConflict	*Attempt by slave to open logical channel in which the master has determined a conflict may occur. (See 8.4.1.3 and 8.5.1.3)*

[End Correction]

9.8 Implementation Clarifications

9.8.1 Timer Values for RAS messages

Description: These are recommended time-out values for the response to RAS messages and subsequent retry counts if a response is not received. (These values are subject to change with further implementation experience and input).

NOTE - If a RAS message is retried due to a timeout value being exceeded, the same sequence number as the previous timed-out message shall be used.

RAS Message	*Time-out value (sec)*	*Retry count*
GRQ	5	2
RRQ	3	2
URQ	3	1
ARQ	3	2
BRQ	3	2
IRQ	3	1
DRQ	3	2
LRQ (multi/uni)	5/3	2

9.8.2 Q.931 Call Issues

9.8.2.1 Q.931 Call Reference Value

There has been some confusion about the encoding and usage of the Call reference value from documentation in the Q.931 standard. Shown below is the pertinent section from the Q.931 standard as referenced in the H.225.0 document. The ‘octet 2’ as indicated by the highlighted field is actually the first octet of the CRV number itself.

Note also the CRV values as passed in RAS messages shall conform to the structure as specified in Q.931. Specifically, the *Flag* shall be included as the most significant bit of the **CallReferenceValue**. This restricts the actual CRV value to 0 through 32767.

7.2.1.2 Call reference (Q.931)

As defined in subclause 4.3/Q.931

A call reference value length of two octets shall be supported by any H.323 end point.

The call reference value is used to associate all messages (RAS and Q.931) relating to the same call, **between two entities**. An entity A sending the first message belonging to a call to another entity B shall choose a CRV which is unique for communication between A and B. In particular, an endpoint receiving a SETUP message might have to allocate another CRV for communication related to this call with its gatekeeper.

Entity A will be regarded as “origin” and B as “destination” in relation to the flag part of the CRV.

NOTE - This is in contradiction to H.323/7.4 which might lead to confusion in the case of a terminal receiving many SETUPs bearing the same CRV, trying, for instance, getting admission for them from a gatekeeper.

The value is encoded following Figure 4-5/Q.931 for a two-octet call reference value. The most significant octet of the reference value is always encoded in octet number 2.

				Bits				
8	7	6	5	4	3	2	1	Octets
0	0	0	0	Length of call reference value (in octets)				1
Flag								2
Call reference value								etc.

9.8.2.2 Q.931 Signalling channel

Implementers of H.323 endpoints have the option of controlling the Q.931 control channel under the following circumstances outlined in H.225.0

From H.225.0 Section 6.1

1. For terminal to terminal call signalling (Figure 9/H.323), either terminal may choose to close the reliable call signalling channel, or to leave it open.
2. For the gatekeeper mediated call signalling case (Figure 8/H.323), the terminals shall keep the reliable port active throughout the call. However, the gatekeeper may chose to close this signalling channel, but should keep the channel open for calls that involve gateways. This will allow the end-to-end transmission of Q.931 information elements such as display information.

The recommend mode of implementation is that endpoints should not close the Q.931 control channel for the duration of a call. Although it is not a strict requirement, this simplification will allow optional signalling to occur between the endpoints. This consistent mode of operation by all endpoints (keeping the Q.931 channel open) implies that all endpoints will issue the Q.931 ReleaseComplete when closing down the connection; after the H.245 EndSession.

9.8.2.3 Q.931 Message transmission

The usage of Q.931 mandates the operation of a reliable transport channel - in the IP environment, TCP. For IP implementations of H.323 however, the practicalities of decoding the Q.931 messages from the TCP stream require the usage of an individual TPKT header for each message. In other words Q.931 messages may be ‘pipelined’ into the TCP channel as long as each message boundary is clearly indicated with an associated TPKT header.

For the receiving entities, this implies that the section of the byte stream that is associated with a particular message should be derived from the TPKT header.

The following paragraph should be added to H.225.0 section 6.1:

“When messages are sent on the reliable H.225.0 call signalling channel, only one whole message shall be sent within the boundaries defined by the reliable transport; there shall be no fragmentation of H.225.0 messages across transport PDUs. (In IP implementations as outlined in Appendix D, this PDU is defined by TPKT).”

9.8.2.4 H.225.0 Cause IE encoding

The description of usage and encoding of ‘cause’ information as passed in ReleaseComplete has led to some confusion. The following text is presented in order to further clarify that which is presented in H.225.0 Sections 7.2.2.8 and 7.3.10. It in no manner, changes or supersedes the text as supplied in the H.225.0 recommendation.

1) The Cause IE and the ReleaseComplete.ReleaseCompleteReason are mutually exclusive; one OR the other shall be used.

2) Gateways will have to map FROM ReleaseComplete-UUIE TO Cause IE when sending ReleaseComplete *to* a circuit side entity *from* a LAN-based entity. (the reverse mapping is not needed as LAN entities are required to decode the Cause IE)

3) The Cause IE will be coded without the 'recommendation identifier' (octet 3a) as shown in Figure 1/Q.850

The overall outline of the information element is as follows:

8	7	6	5	4	3	2	1	Octet
0	0	0	0	1	0	0	0	#1
LENGTH = 2								#2
ext.	coding standard		spare	location				#3
1	0	0	0					
recommendation identifier = (NOT USED)								#3a
Cause value (see table 1 below for acceptable values)								#4
not applicable								#5*

(Figure 1/Q.850)

The mapping *from* ReleaseCompleteReasons *to* Cause IE are as follows:

ReleaseCompleteReason code	corresponding Q.931 / Q.850 cause value
noBandwidth	34 - no circuit/channel available
gatekeeperResources	47 - Resource Unavailable
unreachableDestination	3 - no route to destination
destinationReject	16 - normal call clearing
invalidRevision	88 - incompatible destination
noPermission	111 - Interworking, unspecified
unreachableGatekeeper	38 - network out of order
gatewayResources	42 - switching equipment congestion
badFormatAddress	28 - invalid number format
adaptiveBusy	41 - Temporary Failure
inConference	17 - user busy
undefined	31 - unspecified

9.8.3 Media Codecs

9.8.3.1 H.261 Packetization

The SBIT is the number of most significant bits that shall be ignored in the first data octet, EBIT is the number of least significant bits that shall be ignored in the last data octet.

9.8.3.2 G.711

Clarification to implementers for H.323 Section 6.2.1 on audio coding.

G.711 - Each G.711 octet shall be octet aligned in an RTP packet. The sign bit of each G.711 octet shall correspond to the most significant bit of the octet in the RTP packet (i.e. assuming the G.711 samples are handled as octets on the host machine, the sign bit shall be the most significant bit of the octet as defined by the host machine format).

The 56kbps and 48kbps modes are not applicable to H.323 endpoints, since G.711 shall always be transmitted on the LAN as 8-bit samples (regardless of the sample size on the SCN, i.e. 6 or 7 bit). The H.323 endpoint would always generate 8-bit samples and always expect to receive 8-bit samples. The Gateway shall resolve the truncation or zero filling issues.

Implementers should verify accordance with note 2 in table 1b of G.711 specification. For example, H.323 G.711 coding should occur in the same manner as G.711 is encoded by H.320 endpoints. ~~{need to add text relative inverted msb as per FT issues}~~

9.8.4 H.245 Signalling

H.225.0 Section 6.1 states:

“When messages are sent on the reliable H.245 control channel, more than one message may be sent in a single packet as long as whole messages are sent; there shall be no fragmentation of H.245 messages across packets.”

The usage of H.245 mandates the operation of a reliable transport channel - in the IP environment, TCP. In general there are no issues with overlapping multiple H.245 signalling elements. For IP implementations of H.323 however, the practicalities of decoding the messages boundaries from the TCP stream require the usage of TPKT headers. This is the ‘packet’ that is referred to in the above section; not the IP layer packet. In other words, multiple H.245 messages may be combined into one TPKT by accounting for their combined length in the header value. Each individual message boundary can be deduced as the TPKT is incrementally decoded by message sections.

For the receiving entities, this implies that the section of the TPKT associated with an individual H.245 messages should be derived from the ASN.1 decode.

H.225.0 Section 6.1 should be changed to:

“When messages are sent on the reliable H.245 control channel, more than one message may be sent within the boundaries defined by the reliable transport PDU as long as whole messages are sent messages; there shall be no fragmentation of H.245 messages across transport PDUs. (In IP implementations as outlined in Appendix 16, this PDU is defined by TPKT).”

9.8.5 Connection Establishment

9.8.5.1 SETUP Messages

There is no explicit synchronization or locking, during call establishment between two endpoints. This implies that an endpoint ‘A’ can send a SETUP to endpoint ‘B’ at the same instant that ‘B’ is sending a SETUP to ‘B’. Assuming that A and B only wanted to establish one call between them, implementations should provide some method to resolve this ‘overlap’ of calls.

SETUP overlaps will be handled in the application. In the event that an endpoint receives a SETUP message from a terminal to which it had previously sent its own SETUP, the endpoint may take any action suitable to the implementers. The recommended behavior is that the application go ‘off-hook’ (become busy) during the time that an outstanding SETUP message has not received any response. In the event that the terminal can support more than one concurrent connection, this action is still recommended in cases where the received SETUP is from the same endpoint to which the pending SETUP has been sent.

9.8.5.2 Response to SETUP Messages

Section 8 in H.323 shows a number of diagrams outlining message signalling for various call scenarios (Figures 13-23). The ALERTING message is mandatory; in terms of being able to be sent. However, if the endpoint receiving the SETUP message can respond with a CONNECT, CALLPROCEEDING or RELEASECOMPLETE within 4 seconds, it is not required to send an ALERTING.

The sender of a SETUP message can expect to get either an ALERTING, CALLPROCEEDING, CONNECT or RELEASECOMPLETE within 4 seconds after successful transmission. The meaning of the Alerting message is that the called party (the user) has been alerted of an incoming call. Alerting should never be generated by any system except the ultimate called terminal, and then only when it has actually alerted the user. (Of course, in a case of interworking with POTS, for example, the gateway would generate Alerting when it receives a ringing indication - the POTS equivalent of Alerting.)

A gateway should send CALLPROCEEDING when it receives the SETUP (or after it receives ACF) if it expects more than 4 seconds to elapse before it receives CALLPROCEEDING, ALERTING, CONNECT, or RELEASECOMPLETE from the remote system.

9.8.5.3 STATUS message values

Some call states (as indicated by a STATUS message) can’t be reached by other mandatory states. No endpoint should be required to support *sending* these call states - but if an endpoint is able to support an optional call state, it may signal it. In other words receivers of the STATUS message should be tolerant of state values that itself can not get into.

9.8.5.4 Addressing values in messages

SrcInfo (AliasAddress) in the ARQ is a required sequence, however there are conditions under which this value may not be known. Endpoints are not required to have an alias defined - in this case the SrcInfo sequence is present in the encoding of the ARQ message but it has a 0 length.

A calling endpoint must identify itself in the SETUP message. In order for a calling endpoint to identify itself, it may supply an alias (either an E.164 address and/or an h323-ID). If the source address is in the form of E.164 it should be placed in the calling party number of the SETUP message. (as per instructions in H.225.0 section 7.3.11) If the endpoint only has an h323-ID (or prefers to use this as its source address) it should place the h323-ID in the optional Setup-UUIE, sourceAddress field.

To facilitate crossing intranet boundaries, anytime both sourceAddress and sourceCallSignalAddress (as per section 7.2.2 in this document) are available to the sender of the SETUP message they should both be utilized.

The ARQ value of sourceCallSignalAddress shall be equal to the value in the received SETUP. The inclusion of the sourceCallSignalAddress in all three messages is needed for the association of the two ARQs by the GK, as they might not bear the same CRV. For this reason, both endpoints shall include destCallSignal in their ARQs if available.

H.323 Recommendation Series Defect Report Form

DATE:

CONTACT INFORMATION

NAME:
COMPANY:
ADDRESS:

TEL:
FAX:
E-MAIL:

AFFECTED RECOMMENDATIONS:

DESCRIPTION OF PROBLEM:

SUGGESTIONS FOR RESOLUTION:

NOTE - Attach additional pages if more space is required than is provided above