

ITU-T

H.323 System Implementors' Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(25 January 2013)

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

**Implementors' Guide for Recommendations of
the H.323 System (Packet-based multimedia
communications systems):**

***H.323, H.225.0, H.245, H.246, H.283, H.341,
H.450 Series, H.460 Series, and H.500 Series***

Summary

This document is a compilation of reported defects identified in the versions of ITU-T Recommendation H.323 and its related Recommendations currently in force. It must 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 H.323-series Recommendations.

This revision contains all updates submitted up to and including those at Study Group 16 meeting in Geneva, 14–25 January, 2013.

This Implementors' Guide was approved by ITU-T Study Group 16 on 24 January 2013 (TD 74R1/Plen) and it obsoletes the earlier version of this Implementors' Guide approved on 25 March 2011.

Contact Information

ITU-T Study Group 16 / Rapporteur Question 2/16	Paul E. Jones Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709, USA	Tel: +1 919 476 2048 Fax: E-mail: paulej@packetizer.com
Editor ITU-T Rec. H.323 Series Implementors' Guide	Simon Horne Spranto Australia Pty Ltd Australia	Tel: +61 412 984 323 E-mail: s.horne@spranto.com.au

Table of Contents

1	SCOPE.....	1
2	INTRODUCTION	1
3	REFERENCES.....	1
4	NOMENCLATURE.....	3
5	TECHNICAL AND EDITORIAL CORRECTIONS TO H.323 SERIES RECOMMENDATIONS	3
5.1	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.323 (2009).....	3
5.2	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.225.0 (2009).....	3
5.3	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.245 (2009).....	4
5.4	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.246 (2006).....	4
5.5	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.235 SERIES.....	4
5.6	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.450 SERIES.....	4
5.6.1	<i>Technical and Editorial Corrections to H.450.12 (2001).....</i>	<i>4</i>
5.7	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.341 (1999).....	8
5.7.1	<i>Corrections to H.341 Annex B-1 H225-MIB</i>	<i>8</i>
5.7.2	<i>Corrections to H.341 Annex B-2 RAS-MIB</i>	<i>14</i>
5.7.3	<i>Support for Expanded Country Code Values in T.35 in H.341 Annex B-3.....</i>	<i>16</i>
5.8	TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION H.460 SERIES.....	16
6	IMPLEMENTATION CLARIFICATION.....	16
6.1	TOKEN USAGE IN H.323 SYSTEMS.....	16
6.2	H.235 RANDOM VALUE USAGE IN H.323 SYSTEMS.....	17
6.3	GATEWAY RESOURCE AVAILABILITY MESSAGES	17
6.4	OPENLOGICALCHANNEL IN FASTSTART.....	17
6.5	CLARIFICATION IN Q.931 (1993).....	17
6.6	GRACEFUL CLOSURE OF TCP CONNECTIONS.....	17
6.7	RACE CONDITION ON SIMULTANEOUS CLOSE OF CHANNELS	17
6.8	ACCEPTANCE OF FAST CONNECT.....	18
6.9	SEMANTIC DIFFERENCES BETWEEN LIGHTWEIGHT RRQS AND IRQ/IRR MESSAGES	18
6.10	SPECIFYING THE PAYLOAD FORMAT FOR A CHANNEL.....	18
6.11	VERSION DEPENDENCIES IN ANNEXES.....	18
6.12	ROUTING THROUGH SIGNALING ENTITIES AND DETECTING LOOPS	19
6.13	PACKETIZATION FOR G.729, G.729A, G.711, AND G.723.1	20
6.14	CHECKING VERSIONS FOR T.38 AND V.150.1.....	21
7	ALLOCATED OBJECT IDENTIFIERS AND PORT NUMBERS	21
7.1	ALLOCATED OBJECT IDENTIFIERS.....	21
7.2	ALLOCATED PORT NUMBERS.....	22
8	USE OF E.164 AND ISO/IEC 11571 NUMBERING PLANS.....	23
8.1	E.164 NUMBERING PLAN.....	23
8.2	PRIVATE NETWORK NUMBER	25
9	ASN.1 USAGE, GUIDELINES, AND CONVENTIONS.....	27
9.1	NULL, BOOLEAN, AND NULL/BOOLEAN OPTIONAL.....	27
9.2	ASN.1 USAGE IN H.450-SERIES RECOMMENDATIONS	27
9.2.1	<i>ASN.1 version and encoding rules.....</i>	<i>28</i>
9.2.2	<i>Tagging.....</i>	<i>28</i>
9.2.3	<i>Basic ASN.1 Types.....</i>	<i>28</i>
9.2.4	<i>Value sets, subtyping and constraints used in H.450.x:.....</i>	<i>29</i>
9.2.5	<i>Object classes, parameterization, general constraints, and ROS.....</i>	<i>29</i>
9.2.6	<i>Extensibility and non-standard information</i>	<i>29</i>
9.2.7	<i>List of Operation and Error Codes.....</i>	<i>29</i>
	ANNEX: DEFECT REPORT FORM FOR RECOMMENDATIONS OF THE H.323 SYSTEM.....	33

IMPLEMENTORS' GUIDE FOR RECOMMENDATIONS OF THE H.323 SYSTEM (PACKET-BASED MULTIMEDIA COMMUNICATIONS SYSTEMS)

1 Scope

This guide resolves defects in the following categories:

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

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

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

2 Introduction

This document is a compilation of reported defects identified in the versions of ITU-T Recommendation H.323 and its related Recommendations currently in force. It must 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 H.323-series Recommendations.

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 Q2/16 Rapporteur. The template for a defect report is located at the end of the Guide. Contact information for these parties is included at the front of the document. Return contact information should also be supplied so a dialogue can be established to resolve the matter and an appropriate reply to the defect report can be conveyed. This defect resolution process is open to any interested party. Formal membership in the ITU is not required to participate in this process.

3 References

This document refers to the following ITU-T Recommendations:

- ITU-T Recommendation H.323 (2009), *Packet-Based multimedia communications systems*.
- ITU-T Recommendation H.225.0 (2009), *Call signaling protocols and media stream packetization for packet based multimedia communications Systems*.
- ITU-T Recommendation H.235.0 – H.235.9 (2005), *Security and encryption for H Series (H.323 and other H.245 based) multimedia terminals*.
- ITU-T Recommendation H.245 (2011), *Control protocol for multimedia communication*.
- ITU-T Recommendation H.246 (2006), *Interworking of H-Series multimedia terminals with H-Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN*.
- ITU-T Recommendation H.450.1 (2011), *Generic functional protocol for the support of supplementary services in H.323*.

- ITU-T Recommendation H.450.2 (2011), *Call transfer supplementary service for H.323.*
- ITU-T Recommendation H.450.3 (2011), *Call diversion supplementary service for H.323.*
- ITU-T Recommendation H.450.4 (2013), *Call hold supplementary service for H.323.*
- ITU-T Recommendation H.450.5 (2013), *Call park and call pickup supplementary services for H.323.*
- ITU-T Recommendation H.450.6 (1999), *Call waiting supplementary service for H.323.*
- ITU-T Recommendation H.450.7 (2013), *Message waiting indication supplementary service for H.323.*
- ITU-T Recommendation H.450.8 (2013), *Name identification supplementary service for H.323.*
- ITU-T Recommendation H.450.9 (2000), *Call Completion Supplementary Services for H.323.*
- ITU-T Recommendation H.450.10 (2001), *Call offer supplementary service for H.323.*
- ITU-T Recommendation H.450.11 (2001), *Call intrusion supplementary services.*
- ITU-T Recommendation H.450.12 (2001), *Call Information Additional Network Feature for H.323.*
- ITU-T Recommendation H.460.1 (2013), *Guidelines for the use of generic extensibility framework.*
- ITU-T Recommendation H.460.2 (2013), *Number Portability interworking between H.323 and SCN networks.*
- ITU-T Recommendation H.460.3 (2002), *Circuit status map within H.323 systems.*
- ITU-T Recommendation H.460.4 (2002), *Call priority designation for H.323 calls.*
- ITU-T Recommendation H.460.5 (2002), *H.225.0 transport of multiple Q.931 IE of the same type.*
- ITU-T Recommendation H.460.6 (2013), *Extended Fast Connect Feature.*
- ITU-T Recommendation H.460.7 (2013), *Digit Maps Within H.323 Systems.*
- ITU-T Recommendation H.460.8 (2002), *Querying for alternate routes within H.323 systems.*
- ITU-T Recommendation H.460.9 (2002), *Support for online QoS-Monitoring report.*
- ITU-T Recommendation H.460.10 (2004), *Call party category within H.323 systems.*
- ITU-T Recommendation H.460.11 (2004), *Delayed call establishment within H.323 systems.*
- ITU-T Recommendation H.460.12 (2004), *Glare control indicator within H.323 systems.*
- ITU-T Recommendation H.460.13 (2004), *Called user release control.*
- ITU-T Recommendation H.460.14 (2004), *Support for Multi-Level Precedence and Preemption (MLPP) within H.323 systems.*
- ITU-T Recommendation H.460.15 (2004), *Call signalling transport channel suspension and redirection within H.323 systems.*

- ITU-T Recommendation H.460.16 (2005), *Multiple message release sequence capability*.
- ITU-T Recommendation H.460.17 (2005), *Using H.225.0 call signalling connection as transport for H.323 RAS messages*.
- ITU-T Recommendation H.460.18 (2013), *Traversal of H.323 signalling across network address translators and firewalls*.
- ITU-T Recommendation H.460.19 (2013), *Traversal of H.323 media across network address translators and firewalls*.
- ITU-T Recommendation H.460.20 (2005), *Location number within H.323 systems*.
- ITU-T Recommendation H.283, *Remote device control logical channel transport*.
- ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.
- ISO/IEC 11571 (1998), *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Networks – Addressing*.

4 Nomenclature

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

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

5 Technical and Editorial Corrections to H.323 Series Recommendations

5.1 Technical and Editorial Corrections to ITU-T Recommendation H.323 (2009)

None for this version of the H.323 System IG.

5.2 Technical and Editorial Corrections to ITU-T Recommendation H.225.0 (2009)

None for this version of the H.323 System IG.

5.3 Technical and Editorial Corrections to ITU-T Recommendation H.245 (2009)

None for this version of the H.323 System IG.

5.4 Technical and Editorial Corrections to ITU-T Recommendation H.246 (2006)

None for this version of the H.323 System IG.

5.5 Technical and Editorial Corrections to ITU-T Recommendation H.235 Series

Corrections to H.235 series Recommendations are specified in H.235 Series Implementors' Guide.

5.6 Technical and Editorial Corrections to ITU-T Recommendation H.450 Series

5.6.1 Technical and Editorial Corrections to H.450.12 (2001)

5.6.1.1 Technical Correction

Description:	The receipt of a CmnInform APDU at User A's Endpoint is not described. Therefore add the text below at the end of section 7.1.1.1 ANF-CMN invocation.
---------------------	---

[Begin Correction]

7.1.1.1 ANF-CMN invocation

...

Upon receipt of a CmnInform invoke APDU in any message, the Originating endpoint shall remain in the current state.

[End Correction]

5.6.1.2 Add definition of the states CMN-Wait-Response and CMN-Wait-Answer-Response

Description:	The states CMN-Wait-Response and CMN-Wait-Answer-Response are used only in the SDL diagrams but are not defined anywhere. To avoid confusion, a definition of their meaning is added in section 13.
---------------------	---

[Begin Correction]

13. Specification and Description Language (SDL) Diagrams for ANF-CMN

...

In the following SDLs the states CMN-Wait-Response and CMN-Wait-Answer-Response are used to describe the behavior of the Endpoints using explicit primitive exchange.

The state CMN-Wait-Response is entered at the Endpoint after a primitive CMNRequest indication is received and the previous state was CMN-Idle.

The state CMN-Wait-Answer-Response is entered at the Endpoint after a primitive CMNRequest indication is received and the previous state was CMN-Wait-Answer.

[End Correction]

5.6.1.3 Redesign the SDL Diagrams, add two missing collision branches and delete an erroneous message symbol

Description:	<p>Two collision branches are missing: add in section 13.1 Figure 8/H.450.12 the possible receipt of a CMNInform request from the application in state CMN-Wait-Answer and in Figure 9/H.450.12 the possible receipt of a CMNRequest request in state CMN-Wait-Response.</p> <p>In Figure 9/H.450.12 the receipt of a CMNInform Request in state CMN-Wait-Response shall be ignored and the message with CMNInform invoke APDU shall not be forwarded to endpoint B.</p>
---------------------	--

[Begin Correction]

Editorial - Replace the indicated diagrams by the following:

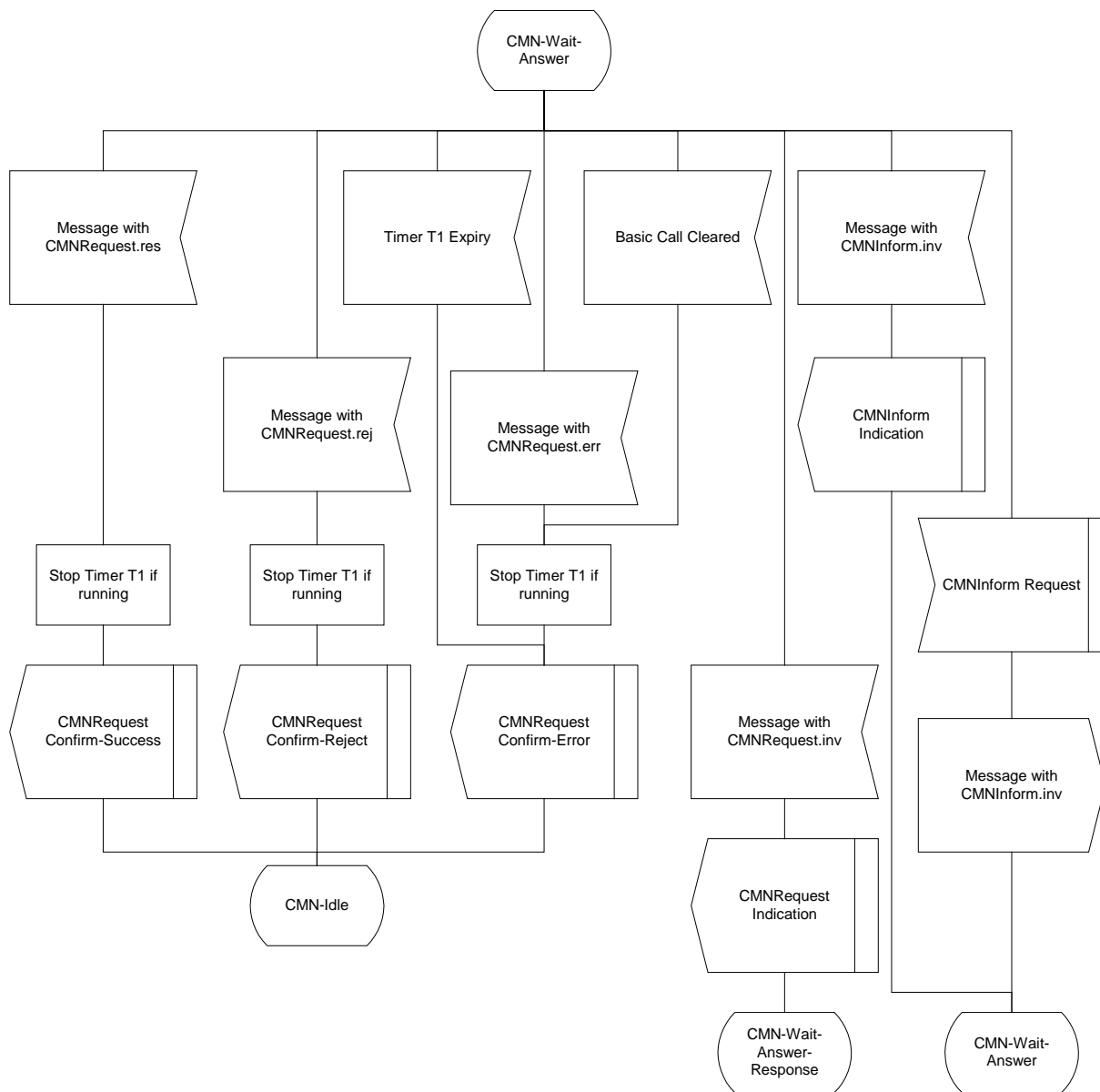


Figure 8/H.450.12 – SDL Representation of ANF-CMN at Endpoint A (Part 3)

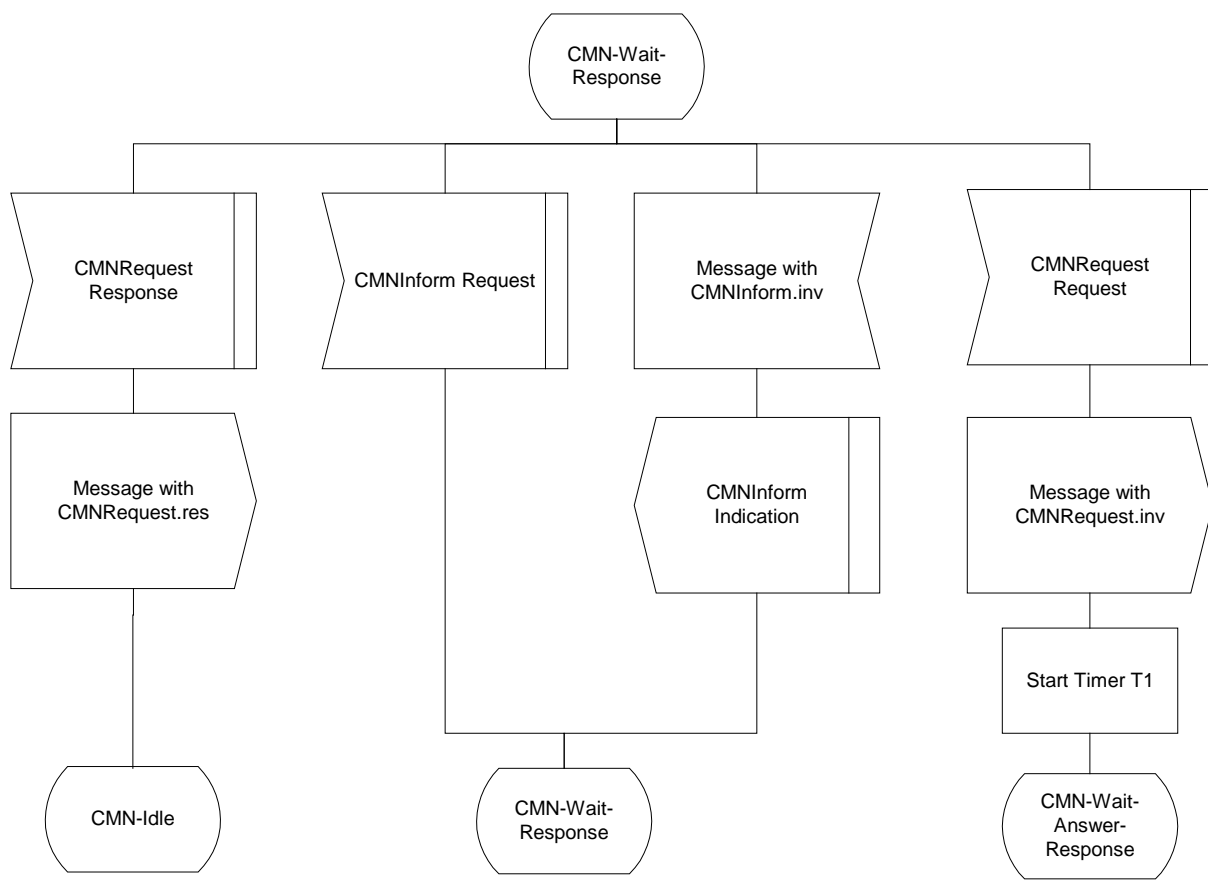


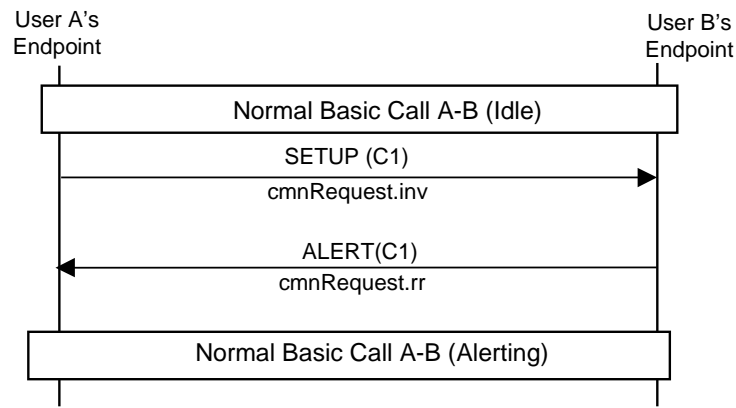
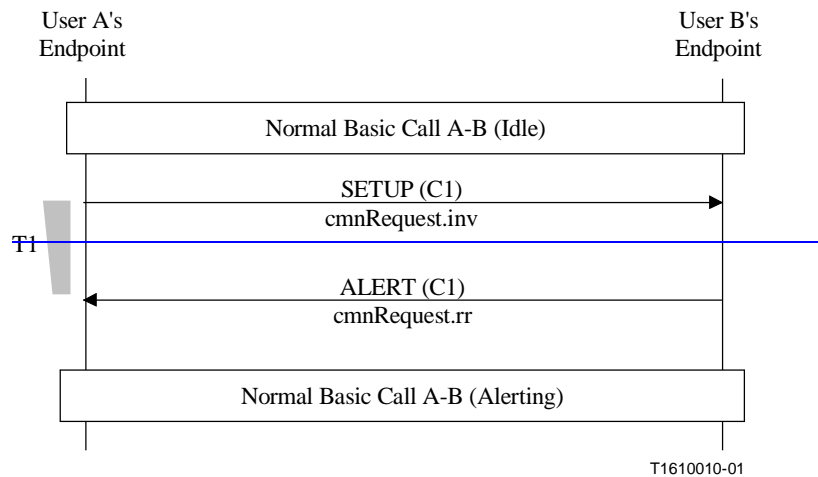
Figure 9/H.450.12 – SDL Representation of ANF-CMN at Endpoint A (Part 3)

[End Correction]

5.6.1.4 Message Flow of ANF-CMN

Description:	Timer T1 is started if cmnRequest invoke is sent in FACILITY message, but not if it is sent in a SETUP message. However, the message flow diagram in Figure 2/H.450.12 erroneously contains timer T1. The erroneous diagram should be replaced by the corrected diagram as below.
---------------------	--

[Begin Correction]



[End Correction]

5.7 Technical and Editorial Corrections to ITU-T Recommendation H.341 (1999)

5.7.1 Corrections to H.341 Annex B-1 H225-MIB

Description:	Each field in CallSignalStatsEntry SEQUENCE referred to the number of messages received ("In") and the number of messages transmitted ("Out"). These counters shall be combined.
---------------------	---

[Begin Correction]

```

CallSignalStatsEntry ::= SEQUENCE {
    callSignalStatsCallConnectionsIn
        Counter32,
    callSignalStatsCallConnectionsOut
        Counter32,
    callSignalStatsAlertingMsgsIn
        Counter32,
    callSignalStatsAlertingMsgsOut
        Counter32,
    callSignalStatsCallProceedingsIn
        Counter32,
    callSignalStatsCallProceedingsOut
        Counter32,
    callSignalStatsSetupMsgsIn
        Counter32,
    callSignalStatsSetupMsgsOut
        Counter32,
    callSignalStatsSetupAckMsgsIn
        Counter32,
    callSignalStatsSetupAckMsgsOut
        Counter32,
    callSignalStatsProgressMsgsIn
        Counter32,
    callSignalStatsProgressMsgsOut
        Counter32,
    callSignalStatsReleaseCompleteMsgsIn
        Counter32,
    callSignalStatsReleaseCompleteMsgsOut
        Counter32,
    callSignalStatsStatusMsgsIn
        Counter32,
    callSignalStatsStatusMsgsOut
        Counter32,
    callSignalStatsStatusInquiryMsgsIn
        Counter32,
    callSignalStatsStatusInquiryMsgsOut
        Counter32,
    callSignalStatsFacilityMsgsIn
        Counter32,
    callSignalStatsFacilityMsgsOut
        Counter32,
    callSignalStatsInfoMsgsIn
        Counter32,
    callSignalStatsInfoMsgsOut
        Counter32,
    callSignalStatsNotifyMsgsIn
        Counter32,
    callSignalStatsNotifyMsgsOut
        Counter32,
    callSignalStatsAverageCallDuration
        Integer32,

```

```

        callSignalStatsCallConnections
        Counter32,
        callSignalStatsAlertingMsgs
        Counter32,
        callSignalStatsCallProceedings
        Counter32,
        callSignalStatsSetupMsgs
        Counter32,
        callSignalStatsSetupAckMsgs
        Counter32,
        callSignalStatsProgressMsgs
        Counter32,
        callSignalStatsReleaseCompleteMsgs
        Counter32,
        callSignalStatsStatusMsgs
        Counter32,
        callSignalStatsStatusInquiryMsgs
        Counter32,
        callSignalStatsFacilityMsgs
        Counter32,
        callSignalStatsInfoMsgs
        Counter32,
        callSignalStatsNotifyMsgs
        Counter32
    }

```

```

callSignalStatsCallConnectionsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      deprecatedcurrent
    DESCRIPTION
        "The number of successful connections in which this entity
         has been a callee."
    ::= { callSignalStatsEntry 1 }
callSignalStatsCallConnectionsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      deprecatedcurrent
    DESCRIPTION
        "The number of successful connections in which this entity
         has been a caller."
    ::= { callSignalStatsEntry 2 }
callSignalStatsAlertingMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      deprecatedcurrent
    DESCRIPTION
        "The number of alerting messages received by this entity."
    ::= { callSignalStatsEntry 3 }
callSignalStatsAlertingMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of alerting messages sent by this entity."
    ::= { callSignalStatsEntry 4 }
callSignalStatsCallProceedingsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of call proceeding messages received by this
         entity."
    ::= { callSignalStatsEntry 5 }

```

```

callSignalStatsCallProceedingsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of call proceeding messages sent by this entity."
    ::= { callSignalStatsEntry 6 }

callSignalStatsSetupMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of setup messages received by this entity."
    ::= { callSignalStatsEntry 7 }
callSignalStatsSetupMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of setup messages sent by this entity."
    ::= { callSignalStatsEntry 8 }
callSignalStatsSetupAckMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of setupAck messages received by this entity."
    ::= { callSignalStatsEntry 9 }
callSignalStatsSetupAckMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of setupAck messages sent by this entity."
    ::= { callSignalStatsEntry 10 }
callSignalStatsProgressMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of progress messages received by this entity."
    ::= { callSignalStatsEntry 11 }
callSignalStatsProgressMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of progress messages sent by this entity."
    ::= { callSignalStatsEntry 12 }

callSignalStatsReleaseCompleteMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of release complete messages received by this
        entity."
    ::= { callSignalStatsEntry 13 }
callSignalStatsReleaseCompleteMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated

```

```

DESCRIPTION
    "The number of release complete messages sent by this
    entity."
    ::= { callSignalStatsEntry 14 }
callSignalStatsStatusMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of status messages received by this entity."
        ::= { callSignalStatsEntry 15 }
callSignalStatsStatusMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of status messages sent by this entity."
        ::= { callSignalStatsEntry 16 }
callSignalStatsStatusInquiryMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of status inquiry messages received by this
        entity."
        ::= { callSignalStatsEntry 17 }

callSignalStatsStatusInquiryMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of status inquiry messages sent by this
        entity."
        ::= { callSignalStatsEntry 18 }
callSignalStatsFacilityMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of connect messages received by this entity."
        ::= { callSignalStatsEntry 19 }
callSignalStatsFacilityMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of connect messages sent by this entity."
        ::= { callSignalStatsEntry 20 }
callSignalStatsInfoMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of info messages received by this entity."
        ::= { callSignalStatsEntry 21 }
callSignalStatsInfoMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      currentdeprecated
    DESCRIPTION
        "The number of info messages sent by this entity."
        ::= { callSignalStatsEntry 22 }

```

```

callSignalStatsNotifyMsgsIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current deprecated
    DESCRIPTION
        "The number of notify messages received by this entity."
        ::= { callSignalStatsEntry 23 }
callSignalStatsNotifyMsgsOut OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current deprecated
    DESCRIPTION
        "The number of notify messages sent by this entity."
        ::= { callSignalStatsEntry 24 }

callSignalStatsAverageCallDuration OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The average duration of the call in minutes since
        system boot time. "
        ::= { callSignalStatsEntry 25 }

callSignalStatsCallConnections OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of successful connections ."
        ::= { callSignalStatsEntry 26 }

callSignalStatsAlertingMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of alerting messages."
        ::= { callSignalStatsEntry 27 }

callSignalStatsCallProceedings OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of call proceeding messages."
        ::= { callSignalStatsEntry 28 }

callSignalStatsSetupMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of setup messages."
        ::= { callSignalStatsEntry 29 }

callSignalStatsSetupAckMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of setupAck messages."
        ::= { callSignalStatsEntry 30 }

```


<u>callSignalStatsProgressMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of progress messages."</u>
	<u>::= { callSignalStatsEntry 31 }</u>

<u>callSignalStatsReleaseCompleteMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of release complete messages."</u>
	<u>::= { callSignalStatsEntry 32 }</u>

<u>callSignalStatsStatusMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of status messages."</u>
	<u>::= { callSignalStatsEntry 33 }</u>

<u>callSignalStatsStatusInquiryMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of status inquiry messages."</u>
	<u>::= { callSignalStatsEntry 34 }</u>

<u>callSignalStatsFacilityMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of connect messages."</u>
	<u>::= { callSignalStatsEntry 35 }</u>

<u>callSignalStatsInfoMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of info messages."</u>
	<u>::= { callSignalStatsEntry 36 }</u>

<u>callSignalStatsNotifyMsgs</u>	<u>OBJECT-TYPE</u>
<u>SYNTAX</u>	<u>Counter32</u>
<u>MAX-ACCESS</u>	<u>read-only</u>
<u>STATUS</u>	<u>current</u>
<u>DESCRIPTION</u>	
	<u>"The number of notify messages."</u>
	<u>::= { callSignalStatsEntry 37 }</u>

[End Correction]

5.7.2 Corrections to H.341 Annex B-2 RAS-MIB

Description:	A few editorial errors have been identified in the RAS MIB in H.341. The rasAdmissionCallIdentifier field is inserted twice in the RasAdmissionTableEntry SEQUENCE. The ASN type of rasRegistrationEndpointType field in RasRegistrationTableEntry SEQUENCE should be changed to MmH323EndpointType .
---------------------	--

[Begin Correction]

```

RAS-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        Counter32, Integer32, OBJECT-TYPE,
        MODULE-IDENTITY, NOTIFICATION-TYPE
            FROM SNMPv2-SMI
        TAddress, RowStatus, TruthValue, DateAndTime
            FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP
            FROM SNMPv2-CONF
        ifIndex
            FROM IF-MIB
        MmGatekeeperID, MmTAddressTag, MmEndpointID, MmGlobalIdentifier,
        MmAliasTag, MmAliasAddress, mmH323Root, MmH323EndpointType,
        MmH225Crv, MmCallType
            FROM MULTI-MEDIA-MIB-TC;

```

...

```

RasRegistrationTableEntry ::=
    SEQUENCE {
        rasRegistrationCallSignallingAddressTag
            MmTAddressTag,
        rasRegistrationCallSignallingAddress
            TAddress,
        rasRegistrationSrcRasAddressTag
            MmTAddressTag,
        rasRegistrationSrcRasAddress
            TAddress,
        rasRegistrationIsGatekeeper
            TruthValue,
        rasRegistrationGatekeeperId
            MmGatekeeperID,
        rasRegistrationEndpointId
            MmEndpointID,
        rasRegistrationEncryption
            TruthValue,
        rasRegistrationWillSupplyUUIE
            TruthValue,
        rasRegistrationIntegrityCheckValue
            TruthValue,
        rasRegistrationTableNumberOfAliases
            Integer32,
        rasRegistrationTableRowStatus
            RowStatus,
        rasRegistrationEndpointType
            MmH323EndpointTypeInteger32,
        rasRegistrationPregrantedARQ
            TruthValue,
        rasRegistrationIsregisteredByRRQ
            TruthValue
    }

```

...

```

rasRegistrationEndpointType OBJECT-TYPE
    SYNTAX MmH323EndpointTypeInteger32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Terminal type represents the type of H.323 terminal:

```

...

```

RasAdmissionTableEntry ::=
    SEQUENCE {
        rasAdmissionSrcCallSignallingAddressTag
            MmTAddressTag,
        rasAdmissionSrcCallSignallingAddress
            TAddress,
        rasAdmissionDestCallSignallingAddressTag
            MmTAddressTag,
        rasAdmissionDestCallSignallingAddress
            TAddress,
        rasAdmissionCallIdentifier
            MmGlobalIdentifier,
        rasAdmissionConferenceId
            MmGlobalIdentifier,
        rasAdmissionRasAddressTag
            MmTAddressTag,
        rasAdmissionRasAddress
            TAddress,
        rasAdmissionCRV
            MmH225Crv,
        rasAdmissionIsGatekeeper
            TruthValue,
        rasAdmissionSrcAliasAddressTag
            MmAliasTag,
        rasAdmissionSrcAliasAddress
            MmAliasAddress,
        rasAdmissionDestAliasAddressTag
            MmAliasTag,
        rasAdmissionDestAliasAddress
            MmAliasAddress,
        rasAdmissionAnswerCallIndicator
            INTEGER,
        rasAdmissionTime
            DateAndTime,
rasAdmissionCallIdentifier
MmGlobalIdentifier,
        rasAdmissionEndpointId
            MmEndpointID,
        rasAdmissionBandwidth
            Integer32,
        rasAdmissionIRRFrequency
            Integer32,
        rasAdmissionCallType
            MmCallType,
        rasAdmissionCallModel
            INTEGER,
        rasAdmissionSrcHandlesBandwidth
            TruthValue,
        rasAdmissionDestHandlesBandwidth
            TruthValue,
        rasAdmissionSecurity
            TruthValue,
        rasAdmissionSrcWillSupplyUUIE
            TruthValue,
        rasAdmissionDestWillSupplyUUIE
            TruthValue,

```

```

rasAdmissionTableRowStatus
    RowStatus
}

```

[End Correction]

5.7.3 Support for Expanded Country Code Values in T.35 in H.341 Annex B-3

Description:	T.35 (1999) expanded the available country codes from one octet to two octets. In order to support the expanded country codes going forward, it is recommended that implementers make the following changes to these definitions in H.341 Annex B-3 H323TERMINAL-MIB.
---------------------	---

[Begin Correction]

```

h323TermSystemt35CountryCode OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Country code, per T.35 Annex A."
 ::= { h323TermSystemEntry 5 }
h323TermSystemt35CountryCodeExtention OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Assigned nationally, unless the country code
is 255, in which case this value shall contain
the country code found in T.35 Annex B."
 ::= { h323TermSystemEntry 6 }

```

[End Correction]

5.8 Technical and Editorial Corrections to ITU-T Recommendation H.460 Series

None for this version of the H.323 System IG.

6 Implementation Clarification

6.1 Token Usage in H.323 Systems

There has been some confusion on the usage of individual **CryptoH323Tokens** as passed in RAS messages. There are two main categories of **CryptoH323Tokens**; those used for H.235 procedures and those used in an application specific manner. The use of these tokens should be according to the following rules:

- All H.235 defined (e.g. **cryptoEPPwdHash**, **cryptoGKPwdHash**, **cryptoEPPwdEncr**, **cryptoGKPwdEncr**, **cryptoGKCert**, and **cryptoFastStart**). shall be utilized with the procedures and algorithms as described in H.235.
- Application specific or proprietary use of tokens shall utilize the **nestedcryptoToken** for their exchanges.

- Any **nestedcryptoToken** used should have a **tokenOID** (object identifier) which unambiguously identifies it.

6.2 H.235 Random Value Usage in H.323 Systems

The random value that is passed in xRQ/xCF sequence between endpoints and Gatekeepers may be updated by the Gatekeeper. As described in section 4.2 of H.235 this random value may be refreshed in any xCF message to be utilized by a subsequent xRQ messages from the endpoint. Due to the fact that RAS messages may be lost (including xCF/xRJ) the updated random value may also be lost. The recovery from this situation may be the reinitializing of the security context but is left to local implementation.

Implementations that require the use of multiple outstanding RAS requests will be limited by the updating of the random values used in any authentication. If the updating of this value occurs on every response to a request, parallel requests are not possible. One possible solution, is to have a logical "window" during which a random value remains constant. This issue is a local implementation matter.

6.3 Gateway Resource Availability Messages

The Resources Available Indication (RAI) is a notification from a gateway to a gatekeeper of its current call capacity for each H-series protocol and data rate for that protocol. The gatekeeper responds with a Resources Available Confirmation (RAC) upon receiving a RAI to acknowledge its reception. A Gatekeeper should ignore any RAI notifications (e.g. send no RAC) upon receiving a RAI which contains bogus information (i.e. a bad endpointIdentifier).

6.4 OpenLogicalChannel in fastStart

In the H.225.0 ASN.1, **fastStart** is defined as SEQUENCE OF OCTET STRING OPTIONAL. The text definition states "This uses the **OpenLogicalChannel** structure defined in H.245..." Each OCTET STRING in **fastStart** is to contain the **OpenLogicalChannel** structure, not an entire request message.

6.5 Clarification in Q.931 (1993)

Table 4-3/Q.931 (1993) (Information Element Identifier Coding) shows that the Progress Indicator IE identifier is 0x1e, but Figure 4-29/Q.931 (octet layout of Progress Indicator IE) shows the identifier as 0x1f. Note that the identifier should be 0x1e.

6.6 Graceful Closure of TCP Connections

When a TCP connection is closed, the graceful closure procedure documented in section 3.5 of RFC 793 should always be used.

6.7 Race Condition on Simultaneous Close of Channels

Section 8.5 of H.323 describes the procedures that an endpoint follows to terminate a call. It should be noted that as prescribed in Step 6, both endpoints shall issue a Release Complete simultaneously. Endpoints should be prepared for this potential race condition.

6.8 Acceptance of Fast Connect

When an endpoint accepts the Fast Connect procedure, it may select from the proposed channels as specified in section 8.1.7.1/H.323. The Recommendation clearly specifies what fields shall be modified by the endpoint to accept both the forward and the reverse channels. An endpoint shall not modify any fields other than those specified in 8.1.7.1/H.323 when returning the proposed channels.

Newer versions of H.245 may introduce new fields into the **OpenLogicalChannel** sequence or one of the structures contained therein, as well as new procedures. An older endpoint is obviously not required to decode such new fields or to return such new fields when accepting any proposal. Implementers should consider the consequences of transmitting a newer H.245 OLC to an older endpoint. For the purposes of Fast Connect, the calling endpoint shall assume that the called endpoint's version of H.245 is the minimum version of H.245 necessary to be complaint with an H.323 device that advertises the version of H.225.0 transmitted in the messages from the called endpoint (refer to the "Summary" section of H.323).

6.9 Semantic Differences between Lightweight RRQs and IRQ/IRR Messages

The lightweight RRQ and the IRR message serve two different functions with an H.323 system. While both are a means of allowing the Gatekeeper to discover that an endpoint is alive, they also each serve separate, unique functions.

The lightweight RRQ is intended to prevent a registration with a Gatekeeper from expiring. The message is generated by the endpoint and does not require the Gatekeeper to poll each endpoint on a regular interval. This message is also a means of allowing the Gatekeeper to provide updated registration information, such as a new list of Alternate Gatekeepers, after the initial registration.

Version 1 of H.323 did not have the concept of a lightweight RRQ, so the IRQ/IRR exchange is the only mechanism available to determine endpoint status of Version 1 devices. However, the lightweight RRQ may be a better choice for determining endpoint status for Version 2 and higher devices.

The IRQ/IRR exchange allows the Gatekeeper to poll the endpoint periodically to discover if the endpoint is still alive. However, an IRR is also intended to convey details about current active calls. This can be used by the Gatekeeper to discover calls that have terminated, which may happen if the endpoint fails to properly send a DRQ message for a call. The IRR message also provides specific details about active calls.

6.10 Specifying the Payload Format for a Channel

Implementers should be conscientious of the fact that there are possibly multiple payload formats defined for media formats. For example, two payload formats are defined for H.263—one is defined for the Recommendation H.263 (1996) and one for Recommendation H.263 (1998). Other payload formats may be defined for existing codecs or revisions of those codecs. For interoperability, it is strongly advised that implementers provide the **mediaPacketization** element of the **h2250LogicalChannelParameters** sequence in the **OpenLogicalChannel** message so that there is no ambiguity as to which payload format is being used.

6.11 Version Dependencies in Annexes

It was noted that the Annexes to H.323 often fail to indicate the minimum version of H.323 and H.245 required for the Annex. This table is an attempt to clarify the version relationships:

<i>H.323 Annex</i>	<i>Minimum H.323 Version</i>	<i>Minimum H.245 Version</i>
<i>Annex Dv1 (1998)</i>	1998 (Version 2)	1998 (Version 4)
<i>Annex Dv2 (2000)</i>	2000 (Version 4)	2000 (Version 7)
<i>Annex Dv3 (2005)</i>	2000 (Version 4)	2005 (Version 11)
<i>Annex E</i>	1998 (Version 2)	N/A
<i>Annex F</i>	1998 (Version 2)	N/A
<i>Annex G</i>	1998 (Version 2)	1998 (Version 4)
<i>Annex Gv2 (2006)</i>	1998 (Version 2)	2000 (Version 7)
<i>Annex J</i>	1998 (Version 2)	N/A
<i>Annex K</i>	1998 (Version 2)	N/A
<i>Annex L</i>	1998 (Version 2)	N/A
<i>Annex M.1</i>	2000 (Version 4)	N/A
<i>Annex M.2</i>	2000 (Version 4)	N/A
<i>Annex M.3 (2001)</i>	2000 (Version 4)	N/A
<i>Annex M.4 (2004)</i>	2000 (Version 4)	N/A
<i>Annex O</i>	2000 (Version 4)	N/A
<i>Annex P</i>	2000 (Version 4)	2003 (Version 9)
<i>Annex Q</i>	1998 (Version 2)	2000 (Version 7)
<i>Annex R</i>	2000 (Version 4)	N/A

6.12 Routing through Signaling Entities and Detecting Loops

In some call scenarios, a call may be routed through a signaling entity multiple times. For example, a call from Endpoint 1 (EP1) may be routed through Gatekeeper 1 (GK1) and Gatekeeper 2 (GK2) to Endpoint 2 (EP2) as shown in Figure 1.

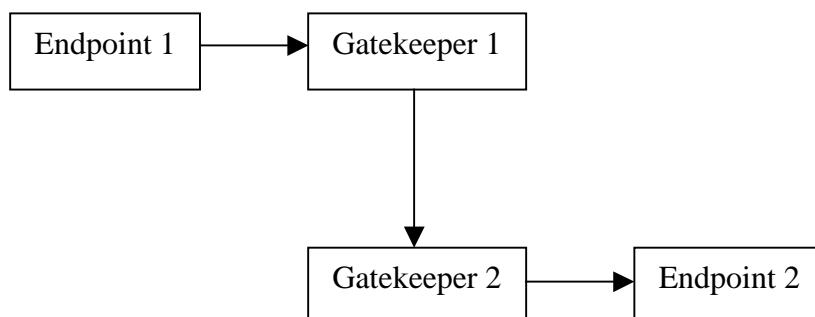


Figure 1 - Call placed through multiple gatekeepers

If EP2 redirects the call to a third endpoint, such as Endpoint 3 (EP3), signaling entities such as GK1 and GK2 should be prepared to handle such call rerouting. For this example, assume that EP2 returned a Facility message with a **reason** of **callForwarded** upon receiving a Setup message.

Rather than propagate that response back to EP1, GK2 may choose to handle the call forward operation. GK2 would send a Release Complete to EP2 and begin rerouting the call. Suppose that GK2 sends an LRQ message to GK1 for EP3 and that GK1 replies with its address so that that calls routed to EP3 are routed through it. GK2 would then send a Setup message for this call to GK1 as shown in Figure 2.

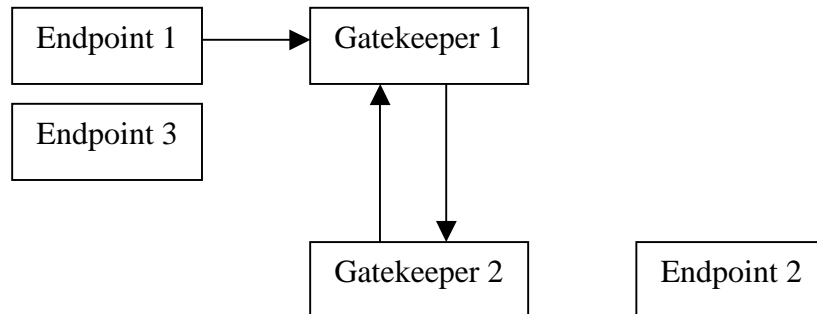


Figure 2 - Gatekeeper 2 re-routes call back to Gatekeeper 1

When GK1 receives the Setup message from GK2, it may inadvertently mistake the call as "bogus", since the Call Identifier will match an already existing call within the Gatekeeper. Implementers should consider this type of call scenario and be prepared to receive incoming calls that contain Call Identifiers for calls that are already being routed through the routing entity. The routing entity should examine not only the Call Identifier, but also the destination address of the call (the call signaling address, aliases, or Called Party Number of the destination). In this case, the call is routed through GK1 with a destination address of EP2 is rerouted by GK2 to GK1, but with a destination address of EP3. In this way, the GK1 will properly handle call routing and rerouting, as well as prevent loops in the call signaling path.

In this example, there was a dependency on the H.323v2 Call Identifier. Unfortunately, H.323 version 1 systems did not have Call Identifiers. For this reason, these loop detection and rerouting procedures are not possible. Nonetheless, it is advisable for routing entities to make an effort to prevent loops properly. For example, if the entities in Figure 2 were version 1 devices, the GK1 may examine the source address, destination address, and Conference Identifier (CID) of the call. The first time the call is presented to the Gatekeeper, the destination address is EP2, just as before. However, when GK re-routes the call back to GK1, the destination address is EP3. In this way, GK1 may allow proper rerouting of the call to EP3.

The logic for Version 1 devices seems similar to that for Version 2 and higher devices, but there are issues when EP2 and EP3 are MCUs, for example. Suppose that EP2 is an MCU that is directing all calls to EP3. The first time a call is redirected to GK1, GK1 may realize that this is, indeed, a call redirection as described above. However, when the second call is redirected, GK1 has no means of distinguishing between the first redirected call and the second: the source address *may* be the same, the destination address is the same as the previously rerouted call (EP3), and the Conference ID is the same. So in this case, GK1 may have no choice but to assume that a loop has occurred and release the offending call. Although this is unfortunate, H.323v2 and higher systems do not suffer from this problem. What is important, though, is that loop detection is possible—even with version 1 systems.

6.13 Packetization for G.729, G.729a, G.711, and G.723.1

The delay associated with codec processing and packetization should be kept as short as possible. To accomplish this objective when G.729 or G.729A is used, two frames per packet should be

considered as the maximum packet size. Similarly, G.711 may be used with packet sizes of 10 ms (80 frames) or 20 ms (160 frames) to achieve this objective. Finally, when G.723.1 is used, only one frame should be included in each packet. The 30 ms frame size of G.723.1 results in speech collection and coding delay of at least 60 ms, contributing to difficulty of interactive communications.

6.14 Checking versions for T.38 and V.150.1

It is important that devices properly negotiate the version of the T.38 or V.150.1 to be used and agree to use the same version. At the present time there are few guidelines for version negotiation. Until the guidelines are developed the following note applies:

Devices supporting multiple versions of T.38 and V.150.1 may offer multiple proposals in Fast Connect, each with a different version specified. A device shall not accept a proposal for a version that it does not support.

7 Allocated Object Identifiers and Port Numbers

Information in this section is provided for informational purposes and convenience. This section does not supercede nor replace proper references in H.225.0, H.225, H.235, or other Recommendations.

7.1 Allocated Object Identifiers

The following object identifiers have been allocated for protocols associated with H.323. Any future object IDs that are allocated should be indexed here to prevent duplication.

Note that object IDs below that are allocated below the arc { itu-t(0) recommendation(0) } are show with an abbreviated prefix of "0 0" below.

{ 0 0 h(8) 2250 version(0) [v] }	H225.0 version numbers
Assigned values of v: 1-4	
{ 0 0 h(8) 2250 annex(1) g(7) version(0) [v] }	H225.0 Annex G version numbers
Assigned values of v: 1-2	
{ 0 0 h(8) 2250 annex(1) g(7) usage(1) [u] }	H225.0 Annex G usage tags
Assigned values of u: none	
{ 0 0 h(8) 245 version(0) [v] }	H245 version numbers
Assigned values of v: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) video(0) [c] }	Generic video capabilities
Assigned values of c: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) audio(1) [c] }	Generic audio capabilities
Assigned values of c: Please refer to Table D.1/H.245	

{ 0 0 h(8) 245 generic-capabilities(1) data(2) [c] }	Generic data capabilities
Assigned values of <i>c</i> : Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) control(3) [c] }	Generic control capabilities
Assigned values of <i>c</i> : Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) multiplex(4) [c] }	Generic multiplex capabilities
Assigned values of <i>c</i> : Please refer to Table D.1/H.245	
{ 0 0 h(8) 283 generic-capabilities(1) 0 }	H.283 Capability
{ iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9) }	Identifies QSIG as the tunneled protocol within an H.225.0 Call Signalling Channel

7.2 Allocated Port Numbers

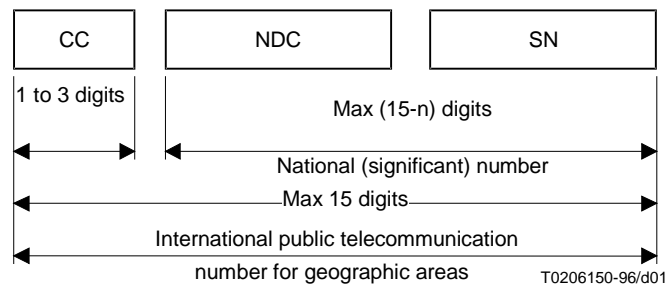
The following IP port numbers have been allocated for various components of H.323:

1300	TLS secured call signalling
1718	Multicast RAS Signalling
1719	Unicast RAS Signalling
1720	TCP call signalling
2099	Annex G/H.225.0 Signalling
2517	Annex E/H.323 Signalling

8 Use of E.164 and ISO/IEC 11571 Numbering Plans

8.1 E.164 numbering plan

ITU-T Recommendation defines E.164 numbers the following way for geographic areas:



CC Country Code for geographic areas
NDC National Destination Code (optional)
SN Subscriber Number
n Number of digits in the country code

NOTE – National and international prefixes are not part of the international public telecommunication number for geographic areas.

Figure – International public telecommunication number structure for geographic areas

Similar descriptions are also defined for non-geographic areas. Recommendation E.164 further defines country codes (CC) for all the countries and regions of the world.

An international E.164 number always starts with a country code and its total length is always 15 digits or less. More importantly, it does not include any prefixes that are part of a dialing plan (for example, "011" for an international call placed in North America, or "1" for a long-distance call), nor does it include "#" or "*". The number "49 30 345 67 00" is an E.164 number with CC=49 for Germany. A national number is the international number stripped of the country code, "30 345 67 00" in this case. The subscriber number is the national number stripped of the national destination code, "345 67 00" in this case.

An E.164 number has global significance: any E.164 number can be reached from any location in the world. A "dialed digit sequence", however, only has significance within a specific domain. Within a typical private numbering plan in an enterprise, for example, a prefix, such as "9", may indicate that a call goes "outside", at which point the local telephone company's dialing plan takes over. Each telephone company or private network is free to choose its own dialing plan. It is also free to change it as it pleases—and frequently does so (adding new area codes, for example).

In a typical geographically determined network where users input telephone numbers manually and where users do not travel too much, having different dialing plans everywhere is usually a problem. However, when a user travels, the user must determine the other network's numbering plan in order to place calls. When computer systems perform the dialing automatically, the user is usually required to customize the dialing software for every region or network.

Because of these issues with varying dialing plans and automated dialing, it is essential to be able to refer to an absolute "telephone number" instead of "what you have to dial to reach it from a specific location." Proper usage of E.164 numbers can resolve these issues. Many systems use E.164

numbers instead of dialed digits: for example, a PBX may gather the dialed digits from a user on a telephone and then initiate a call to the local phone company using an E.164 number in the Called Party Number information element in Q.931. When completing the Called Party Number IE, specifying the numbering plan as "ISDN/telephony numbering plan (Recommendation E.164)" indicates an E.164 number. Specifying the type of number as "unknown" and the specifying the numbering plan as "unknown" indicates dialed digits.

The following are a set of definitions from E.164:

number

A string of decimal digits that uniquely indicates the public network termination point. The number contains the information necessary to route the call to this termination point.

A number can be in a format determined nationally or in an international format. The international format is known as the International Public Telecommunication Number which includes the country code and subsequent digits, but not the international prefix.

numbering plan

A numbering plan specifies the format and structure of the numbers used within that plan. It typically consists of decimal digits segmented into groups in order to identify specific elements used for identification, routing and charging capabilities, e.g. within E.164 to identify countries, national destinations, and subscribers.

A numbering plan does not include prefixes, suffixes, and additional information required to complete a call.

The national numbering plan is the national implementation of the E.164 numbering plan.

dialing plan

A string or combination of decimal digits, symbols, and additional information that define the method by which the numbering plan is used. A dialing plan includes the use of prefixes, suffixes, and additional information, supplemental to the numbering plan, required to complete the call.

address

A string or combination of decimal digits, symbols, and additional information which identifies the specific termination point(s) of a connection in a public network(s) or, where applicable, in interconnected private network(s).

prefix

A prefix is an indicator consisting of one or more digits, that allows the selection of different types of number formats, networks and/or service.

international prefix

A digit or combination of digits used to indicate that the number following is an International Public Telecommunication Number.

country code (CC) for geographic areas

The combination of one, two or three digits identifying a specific country, countries in an integrated numbering plan, or a specific geographic area.

national (significant) number [N(S)N]

That portion of the number that follows the country code for geographic areas. The national (significant) number consists of the National Destination Code (NDC) followed by the Subscriber Number (SN). The function and format of the N(S)N is nationally determined.

national destination code (NDC)

A nationally optional code field, within the E.164 number plan, which combined with the Subscriber's Number (SN) will constitute the national (significant) number of the international public telecommunication number for geographic areas. The NDC will have a network and/or trunk code selection function.

The NDC can be a decimal digit or a combination of decimal digits (not including any prefix) identifying a numbering area within a country (or group of countries included in one integrated numbering plan or a specific geographic area) and/or network/services.

national (trunk) prefix

A digit or combination of digits used by a calling subscriber, making a call to a subscriber in his own country but outside his own numbering area. It provides access to the automatic outgoing trunk equipment.

subscriber number (SN)

The number identifying a subscriber in a network or numbering area.

8.2 Private Network Number

Private Network Numbers are used in private or virtual private telephony networks, e.g., a corporate network of PBXs and virtual private lines.

ISO/IEC 11571 defines Private Network Number (PNP) as having up to three regional levels.

A PNP Number shall comprise a sequence of x decimal digits (0,1,2,3,4,5,6,7,8,9) with the possibility that different PNP Numbers within the same PNP can have different values of x. The maximum value of x shall be the same as for the public ISDN numbering plan, see ITU-T Recommendation E.164.

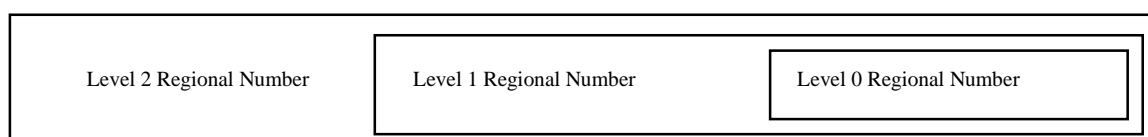


Figure – H.323 - Structure of a PNP Number with three levels of regions

A level n Regional Number (RN) shall have significance only within the level n region to which it applies. When that number is used outside that level n region, it shall be in the form of an RN of level greater than n. Only a Complete Number shall have significance throughout the entire PNP.

A typical example in North America would be a 4-digit "extension" as the Level 0 Regional Number: a 3-digit "location code" combined with the 4 digit "extension" would form the Level 1 Regional Number. The Level 2 Regional Number would be nil.

A prefix could also be used to signal which regional number is used, and would not be part of the regional number per se, but only part of the dialing plan. Again, a typical example would be the use of digit "6" to access a Level 1 Regional Number, and no digit for a Level 0 Regional Number.

The following are a set of definitions from ISO/IEC 11571:

Private Numbering Plan (PNP)

The numbering plan explicitly relating to a particular private numbering domain, defined by the PISN Administrator of that domain.

PNP Number

A number belonging to a PNP.

Region

The entire domain or a sub-domain of a PNP. A region does not necessarily correspond to a geographical area of a PISN.

Region Code (RC)

The leading digits of a PNP Number which identify a region. The RC may be omitted to yield a shortened form of a PNP Number for use internally to that region.

Regional Number (RN)

A particular form of a PNP Number which is unambiguous in the region concerned.

Complete Number

A number which is unambiguous in the entire PNP, i.e. which corresponds to the highest regional level employed in that PISN.

9.1 NULL, BOOLEAN, and NULL/BOOLEAN OPTIONAL

Throughout the ASN.1 used in H.323-series documents, the reader will see the types NULL and BOOLEAN used, along with the modifier OPTIONAL in some cases. People have questioned when NULL should be used or when BOOLEAN should be used and what the semantic differences are.

The BOOLEAN type allows a TRUE or FALSE value to be conveyed in the protocol. When used in conjunction with OPTIONAL, it actually allows three values to be conveyed through the protocol: TRUE, FALSE, and *absent*. The question is what does *absent* mean? In some instances, the absence of a BOOLEAN OPTIONAL means should be interpreted as FALSE, while in other cases, it should be interpreted as "I don't care" or "I don't know"—but not always. For example, the **additiveRegistration** field in the RRQ of H.225.0 Version 4 is defined as a BOOLEAN OPTIONAL. When present, it clearly indicates that the endpoint supports the feature or does not support the feature. However, absence of this field shall also be interpreted as FALSE. The reason is that an older endpoint would not know anything about the field and would obviously not be able to include it. Moreover, they certainly do not support the feature. Another example is the **originator** field in the **perCallInfo** sequence. When present, the meaning is quite clear: the caller is the originator or the terminator of the call. However, if the field is not present, it may mean that the endpoint does not know or cannot supply this information for some reason.

The NULL type is often used to select one of several CHOICE options. NULL carries no particular value, as it merely indicates presence. In selecting the conference goal in a Setup message, for example, the goal CHOICEes are simply NULL types to allow the endpoint to indicate a selection. Another common use of NULL is with the OPTIONAL modifier. A NULL OPTIONAL type allows an endpoint to indicate support for a feature, for example. It is similar in semantics to a BOOLEAN in that the presence of a NULL field indicates TRUE and absence of the NULL field indicates a FALSE. As an example, the **fastConnectRefused** field in the Alerting message is a NULL OPTIONAL. Absence of the field is interpreted as FALSE—Fast Connect is not (yet) refused. Presence of the field, though, clearly indicates refusal of Fast Connect. So why was BOOLEAN not used as the type for this field? It would not have made the encoding any clearer, because the field is past the extension marker (ellipsis). A version 1 and 2 device, for example, would not know to send this field, so there would be three values to consider if BOOLEAN were used: TRUE, FALSE, and *absent*.

Ideally, a field will convey no more values than makes sense. In most cases, these types indicate only two possible values: TRUE/present or FALSE/absent. However, there may be cases where three values are intended and the reader should refer to the appropriate Recommendation to determine if, indeed, there is significance in tri-state fields.

9.2 ASN.1 Usage in H.450-Series Recommendations

This section summarizes the use of ASN.1 in the current H.450.x Recommendations. This information is provided for implementers of the H.450.x protocols, as well as authors of new H.450.x Recommendations.

9.2.1 ASN.1 version and encoding rules

The ASN.1 code in H.450.x is based on the 1994 version of X.680-683, including the amendments on “*Rules of extensibility*”.

The *basic aligned variant of packed encoding rules* (PER) is used as specified in X.691 (1995).

9.2.2 Tagging

All modules defined in Recommendations H.450.x use the *tag default* AUTOMATIC TAGS.

The ROS APDUs (see below) are defined in H.450.1 as *tagged types* within the CHOICE type ROS. No other type defined in H.450.x is a *tagged type*, i.e. all *sets*, *sequences* and *choices* (except ROS) are automatically tagged.

9.2.3 Basic ASN.1 Types

The following types occur in ASN.1 definitions of H.450.x:

BMPString, NumericString	NULL
BOOLEAN	OBJECT IDENTIFIER
CHOICE	OCTET STRING
CLASS (see below)	Open type (see below)
ENUMERATED	SEQUENCE
GeneralizedTime	SEQUENCE OF
INTEGER	SET OF

No use is currently foreseen for the following basic types (needs consideration on a case-by-case basis):

CHARACTER STRING	ObjectDescriptor
EMBEDDED PDV	REAL
EXTERNAL	UTCTime
GeneralString, GraphicString, PrintableString, TeletexString (T61String), UniversalString, VideotexString, VisibleString (ISO646String)	

Use of the following basic types in future Recommendations H.450.x should not be precluded (needs consideration on a case-by-case basis):

BIT STRING	Selection Type (out of a CHOICE)
IA5String	SET
INSTANCE OF	TYPE-IDENTIFIER (see X.681)

Note: Some of these types are already used by other Recommendations in the H.323 universe, e.g. BIT STRING and TYPE-IDENTIFIER in H.235.

9.2.4 Value sets, subtyping and constraints used in H.450.x:

H.450.x Recommendations use *size constraints* (strings, set-of and sequence-of) and *value range constraints* (integers). In H.450.1 *inner subtyping* (“WITH COMPONENTS”) is used occasionally.

The use of *value sets*, *single values*, *contained subtypes* and *permitted alphabets* should be possible if needed by future services. The *type constraint* (for restricting an *open type*) may be useful, too.

Explicit set arithmetic (UNION, INTERSECTION, EXCEPT, ALL EXCEPT) is currently not used on subtype specifications.

9.2.5 Object classes, parameterization, general constraints, and ROS

H.450.1 defines a *remote operations service* (ROS) based on X.880. ROS uses *object classes* (X.681), *parameterization* (X.683) and *constraints* (X.682) for its generic part.

Two object classes OPERATION and ERROR are defined and then used to define four PDU types (*Invoke*, *ReturnResult*, *ReturnError* and *Reject*) as sequences containing individual parts of these classes. The first three PDU types contain an optional *open type* component which is tied by a *table constraint* (“at (@)” notation) to the code value identifying the particular operation or error.

For each supplementary service the actual operations and errors are then defined as *object instances* of the generic classes OPERATION and ERROR in the corresponding Rec. H.450.x. Each operation and error is identified uniquely (within the context of the H.450.x series) by a code value (type INTEGER). A list of currently assigned operation and error values is contained in section 10.8 below.

Each supplementary service defines an *object set* containing all operations defined for that service.

9.2.6 Extensibility and non-standard information

Wherever meaningful, an *extension marker* (ellipsis “...”) is included in the definitions.

All operations, and some errors, include placeholders for non-standard (e.g. manufacturer-specific) information. This non-standard information can either be of type *NonStandardParameter* (imported from H.225.0) or of type *Extension*, which is defined in H.450.1 and consists of an *object identifier* followed by an *open type*. The definition of the Extension type uses an *object class* (EXTENSION) with *parameterization* and *constraints* similar to the ROS definition.

Usually there is space for more than one addition of non-standard information in an operation. Additions of both types (*NonStandardParameter* and *Extension*) can be mixed in any order.

9.2.7 List of Operation and Error Codes

Table 10.1: ASN.1 Operation values used in H.450 series

Value number	Value name	Defined in standard:
0	callingName	H.450.8
1	called alertingName	H.450.8
2	connectedName	H.450.8
3	busyName	H.450.8
7	callTransferIdentity callTransferIdentify	H.450.2
8	callTransferAbandon	H.450.2

9	callTransferInitiate	H.450.2
10	callTransferSetup	H.450.2
11	callTransferActive	H.450.2
12	callTransferComplete	H.450.2
13	callTransferUpdate	H.450.2
14	subaddressTransfer	H.450.2
15	activateDiversionQ	H.450.3
16	deactivateDiversionQ	H.450.3
17	interrogateDiversionQ	H.450.3
18	checkRestriction	H.450.3
19	callRerouting	H.450.3
20	divertingLegInformation1	H.450.3
21	divertingLegInformation2	H.450.3
22	divertingLegInformation3	H.450.3
23	cfnrDivertedLegFailed	H.450.3
27	ccnrRequest	Draft H.450.9
28	ccCancel	Draft H.450.9
29	ccExecPossible	Draft H.450.9
31	ccRingout	Draft H.450.9
32	ccSuspend	Draft H.450.9
33	ccResume	Draft H.450.9
34	callOfferRequest	H.450.10
40	ccbsRequest	Draft H.450.9
43	callIntrusionRequest	H.450.11
44	callIntrusionGetCIPL	H.450.11
45	callIntrusionIsolate	H.450.11
46	callIntrusionForcedRelease	H.450.11
47	callIntrusionWOBRequest	H.450.11
49	cfbOverride	H.450.10 (re-used in H.450.11)
80	mwiActivate	H.450.7
81	mwiDeactivate	H.450.7
82	mwiInterrogate	H.450.7
84	cmnRequest	H.450.12
85	cmnInform	H.450.12
100	divertingLegInformation4	H.450.3
101	holdNotific	H.450.4
102	retrieveNotific	H.450.4
103	remoteHold	H.450.4
104	remoteRetrieve	H.450.4

105	callWaiting	H.450.6 (re-used in H.450.10, H.450.11)
106	cpRequest	H.450.5
107	cpSetup	H.450.5
108	groupIndicationOn	H.450.5
109	groupIndicationOff	H.450.5
110	pickrequ	H.450.5
111	pickup	H.450.5
112	pickExe	H.450.5
113	cpNotify	H.450.5
114	cpickupNotify	H.450.5
115	remoteUserAlerting	H.450.10 (re-used in H.450.11)
116	callIntrusionSilentMonitor	H.450.11
117	callIntrusionNotification	H.450.11

Table 10.2: ASN.1 Error Values used in H.450 series

Value number	Value name	Defined in standard:
0	userNotSubscribed	H.450.1
1	rejectedByNetwork	H.450.1
2	rejectedByUser	H.450.1
3	notAvailable	H.450.1
5	insufficientInformation	H.450.1
6	invalidServedUserNumber	H.450.1
7	invalidCallState	H.450.1
8	basicServiceNotProvided	H.450.1
9	notIncomingCall	H.450.1
10	supplementaryServiceInteractionNotAllowed	H.450.1
11	resourceUnavailable	H.450.1
12	invalidDivertedNumber	H.450.3
14	specialServiceNumber	H.450.3
15	diversionToServedUserNumber	H.450.3
24	numberOfDiversionExceeded	H.450.3
25	callFailure	H.450.1
31	notActivated	H.450.7
43	proceduralError	H.450.1
1000	temporarilyUnavailable	H.450.3, H.450.11
1004	invalidReroutingNumber	H.450.2
1005	unrecognizedCallIdentity	H.450.2
1006	establishmentFailure	H.450.2

1007	notAuthorized	H.450.3, H.450.11
1008	unspecified	H.450.2, H.450.3
1009	notBusy	H.450.11
1010	shortTermRejection	Draft H.450.9
1011	longTermRejection	Draft H.450.9
1012	remoteUserBusyAgain	Draft H.450.9
1013	failureToMatch	Draft H.450.9
1018	invalidMsgCentreId	H.450.7
2000	callPickupIdUnvalid	H.450.5
2001	callAlreadyPickedUp	H.450.5
2002	undefined	H.450.4, H.450.5, H.450.7, (re-used in H.450.9, H.450.11, H.450.12)

Annex: Defect Report Form for Recommendations of the H.323 System
--

DATE:	
CONTACT INFORMATION NAME: COMPANY: ADDRESS: TEL: FAX: EMAIL:	
AFFECTED RECOMMENDATIONS:	
DESCRIPTION OF PROBLEM:	
SUGGESTIONS FOR RESOLUTION:	

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