Packetizer®

# H.323 Protocol Overview

Paul E. Jones

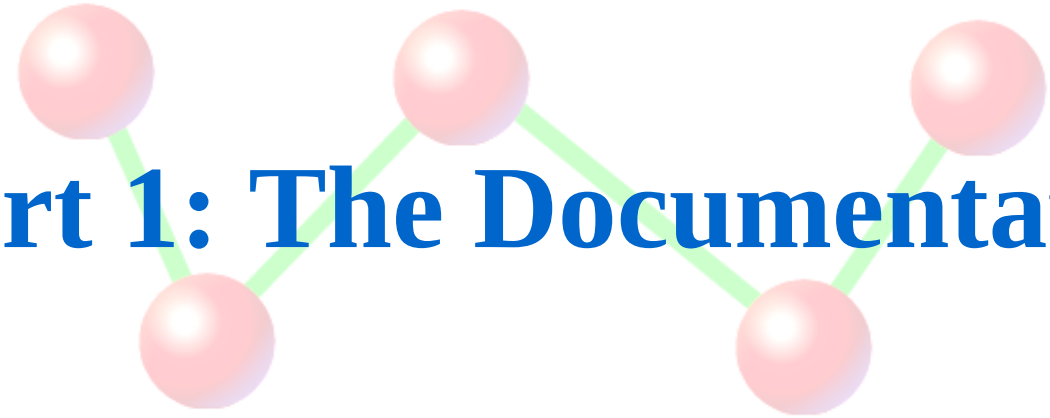
(October 2007)

# Assumptions

- Familiarity with audio, video, and data conferencing
- Familiarity with the role of Coders/Decoders (codecs)
- Familiarity with RTP/RTCP

# Sections

- Part 1: The Documentation
- Part 2: High Level Overview
- Part 3: ASN.1 Overview
- Part 4: RAS
- Part 5: Annex G/H.225.0
- Part 6: H.225.0 Call Signaling
- Part 7: H.245
- Part 8: Fast Connect
- Part 9: Extensibility
- Part 10: TCS=0
- Part 11: Odds and Ends



# Part 1: The Documentation

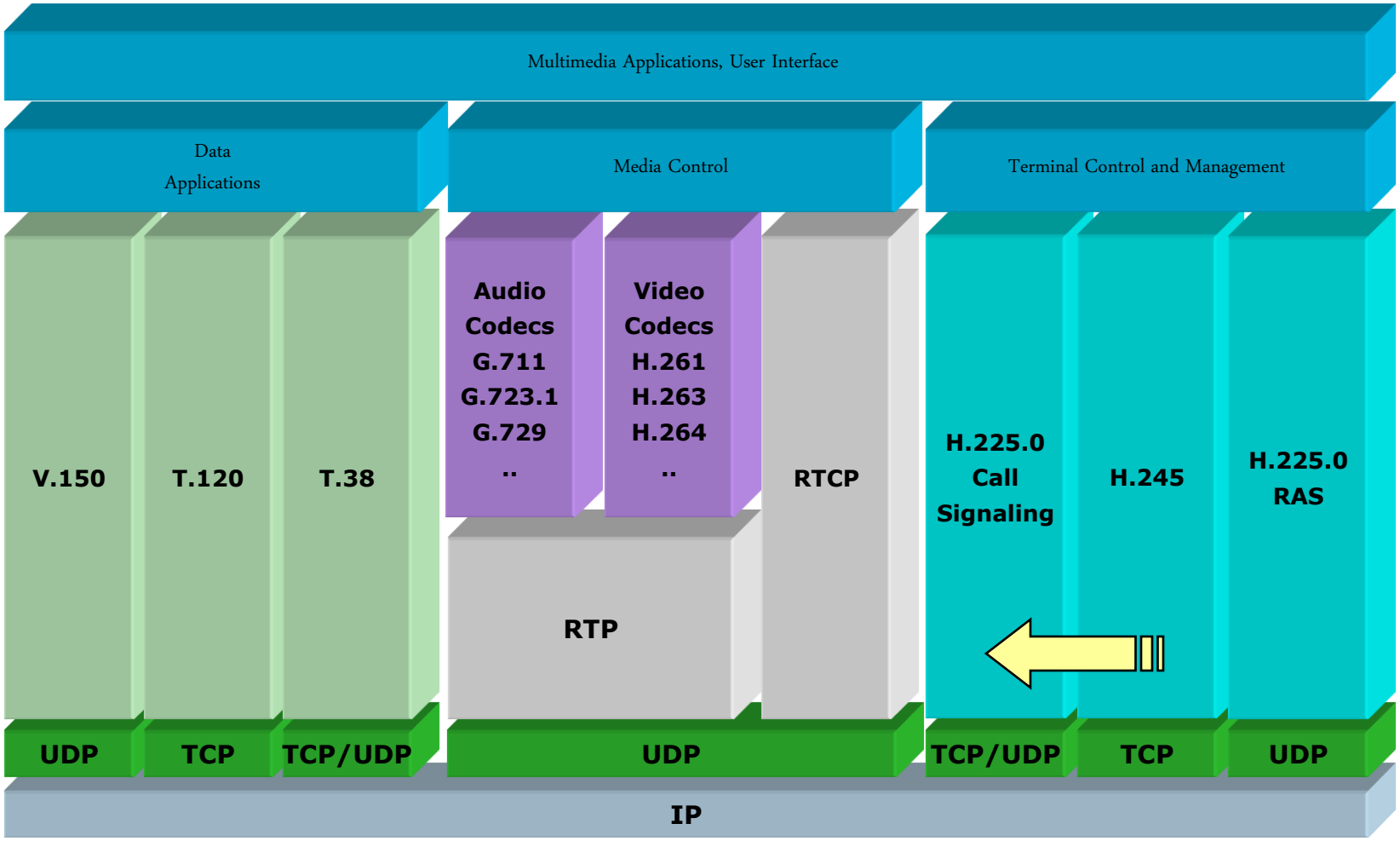
# Who Defined H.323?

- Recommendation H.323 is a standard published by the International Telecommunications Union Telecommunications Sector (ITU-T)
  - Formerly known as CCITT
  - Refer to http://www.itu.int/ITU-T/
  - A permanent organ of the United Nations System (refer to http://www.unsystem.org/)

# Base H.323 Documents

- H.323 – “Umbrella” document that describes the usage of H.225.0, H.245, and other related documents for delivery of packet-based multimedia conferencing services
- H.225.0 – Describes three signaling protocols (RAS, Call Signaling, and “Annex G”)
- H.245 – Multimedia control protocol (common to H.310, H.323, and H.324)

# Typical H.323 Stack



# Additional Documents

- H.235 – Security within H.245-based systems
- H.245 – Interworking with the PSTN
- H.450.x – Supplementary services
- H.460.x – Various H.323 protocol extensions
- H.501 – Protocol for mobility management and inter/intra-domain communication
- H.510 – User, terminal, and service mobility
- H.530 – Security specification for H.510

# Implementers Guide

- H.323-Series Implementers Guide
  - Addresses issues not adequately covered within the aforementioned recommendations
  - Contains corrections to the aforementioned recommendations
  - Updated every 9 months or so

**Must** be read in conjunction with the various H.323-related Recommendations

# Annex versus Appendix

- Often Recommendations will have Annexes and appendices
- An Annex is considered a “normative” part of the Recommendation (i.e., it is part of the standard)
- An Appendix is considered a “non-normative” part of the Recommendation (i.e., it is informational only)

# H.323 Annexes

- Annex A – Mandatory H.245 messages
- Annex B – Procedures for layered video codecs
- Annex C – H.323 on ATM
- Annex D – Fax
- Annex E – UDP for Call Signaling
- Annex F – Simple Endpoint Type (SET)
- Annex G – Text telephony
- Annex I – Error prone channels *(work in progress)*
- Annex J – Secure SET

# H.323 Annexes

- Annex K – HTTP-based service control
- Annex L – Stimulus control protocol
- Annex M.x – Tunneling of various protocols within H.323
- Annex N – QoS *(work in progress)*
- Annex O – Use of DNS *(work in progress)*
- Annex P – Modem over IP
- Annex Q – Far-end camera control
- Annex R - Robustness

# H.323 Appendices

- Appendix I – Sample MC/terminal communications
- Appendix II – Usage of RSVP
- Appendix III – Gatekeeper based user location
- Appendix IV - Signalling prioritized alternative logical channels in H.245
- Appendix V - Use of E.164 and ISO/IEC 11571 numbering plans

# H.225.0 Annexes

- Annex A – RTP/RTCP (RFC 1889)
- Annex B – RTP profile (RFC 1890)
- Annex C – RTP payload for H.261
- Annex D – RTP payload for H.261A
- Annex E – Video packetization
- Annex F – Audio and multiplexed packetization
- Annex G – Communication between and within Administrative Domains
- Annex H – ASN.1 Syntax
- Annex I – H.263+ packetization

# H.225.0 Appendices

- Appendix I – RTP/RTCP algorithms (reference to RFC 1889)
- Appendix II – RTP profile (reference to RFC 1890)
- Appendix III – H.261 packetization (reference to RFC 2032)
- Appendix IV – TCP/IP/UDP usage
- Appendix V – ASN.1 usage

# H.245 Annexes

- Annex A – ASN.1 syntax
- Annex B – Semantic definition of messages
- Annex C – Procedures
- Annex D – Object identifier assignments
- Annex E to M – Various “generic capability” definitions, including some codecs

# H.245 Appendices

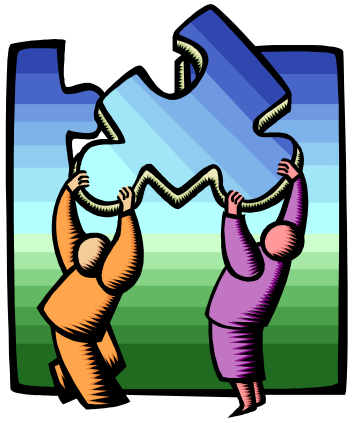
- Appendix I – Overview of ASN.1
- Appendix II – Example of H.245 procedures
- Appendix III – Timers and counters
- Appendix IV – H.245 extension procedure
- Appendix V – Using “replacementFor”
- Appendix VI – Example H.263 capabilities
- Appendix VII – Procedures and template for generic capabilities
- Appendix VIII – List of generic capabilities for H.245 defined in other Recommendations
- Appendix IX – Usage of ASN.1 in H.245

# Part 2: High Level Overview

# Elements of an H.323 System

- Terminals
- Multipoint Control Units (MCUs)
- Gateways
- Gatekeeper
- Border Elements / Peer Elements

Referred to as "endpoints"



# Terminals

- Telephones
- Video phones
- IVR devices
- Voicemail Systems
- “Soft phones” (e.g., NetMeeting®)

T

# MCUs

- Responsible for managing multipoint conferences (two or more endpoints engaged in a conference)

- The MCU contains a Multipoint Controller (MC) that manages the call signaling and may optionally have Multipoint Processors (MPs) to handle media mixing, switching, or other media processing

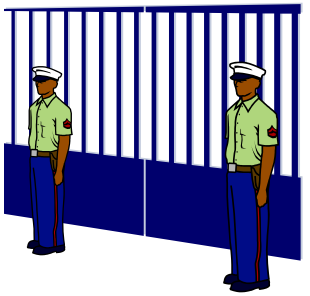
MCU

# Gateways

- The Gateway is composed of a “Media Gateway Controller” (MGC) and a “Media Gateway” (MG), which may co-exist or exist separately
- The MGC handles call signaling and other non-media-related functions
- The MG handles the media
- Gateways interface H.323 to other networks, including the PSTN, H.320 systems, other H.323 networks (proxy), etc.

GW

# Gatekeeper



- The Gatekeeper is an *optional* component in the H.323 system which is used for admission control and address resolution
- The gatekeeper may allow calls to be placed directly between endpoints or it may route the call signaling through itself to perform functions such as follow-me/find-me, forward on busy, etc.

GK

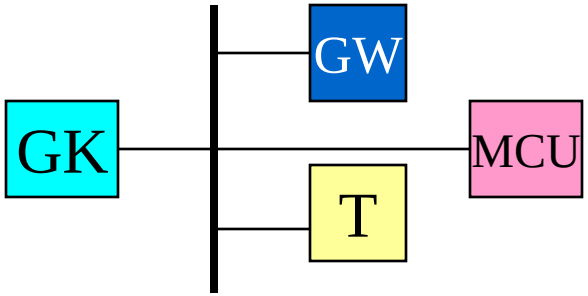
# Border Elements

- Border Elements, which are often co-located with a Gatekeeper, exchange addressing information and participate in call authorization between administrative domains
- Border Elements may aggregate address information to reduce the volume of routing information passed through the network
- Border elements may assist in call authorization/authentication directly between two administrative domains or via a clearinghouse
- Peer elements are like “border elements”, but reside within the interior of the administrative domain

BE

# Zone

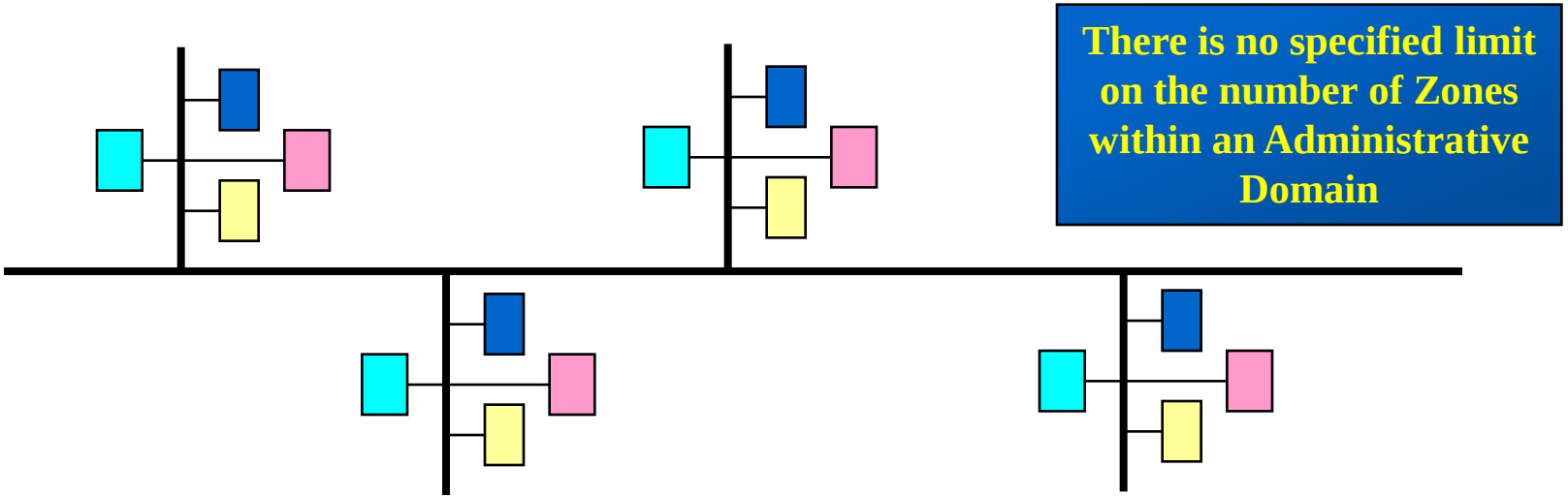
- A single Gatekeeper and all of the devices connected to it
- The physical location of the Gatekeeper with respect to its endpoints is immaterial
- There may be more than one physical Gatekeeper device that provides the logical Gatekeeper functionality for a zone




**There is no imposed limit on the number or types of devices in a zone**

# Administrative Domain

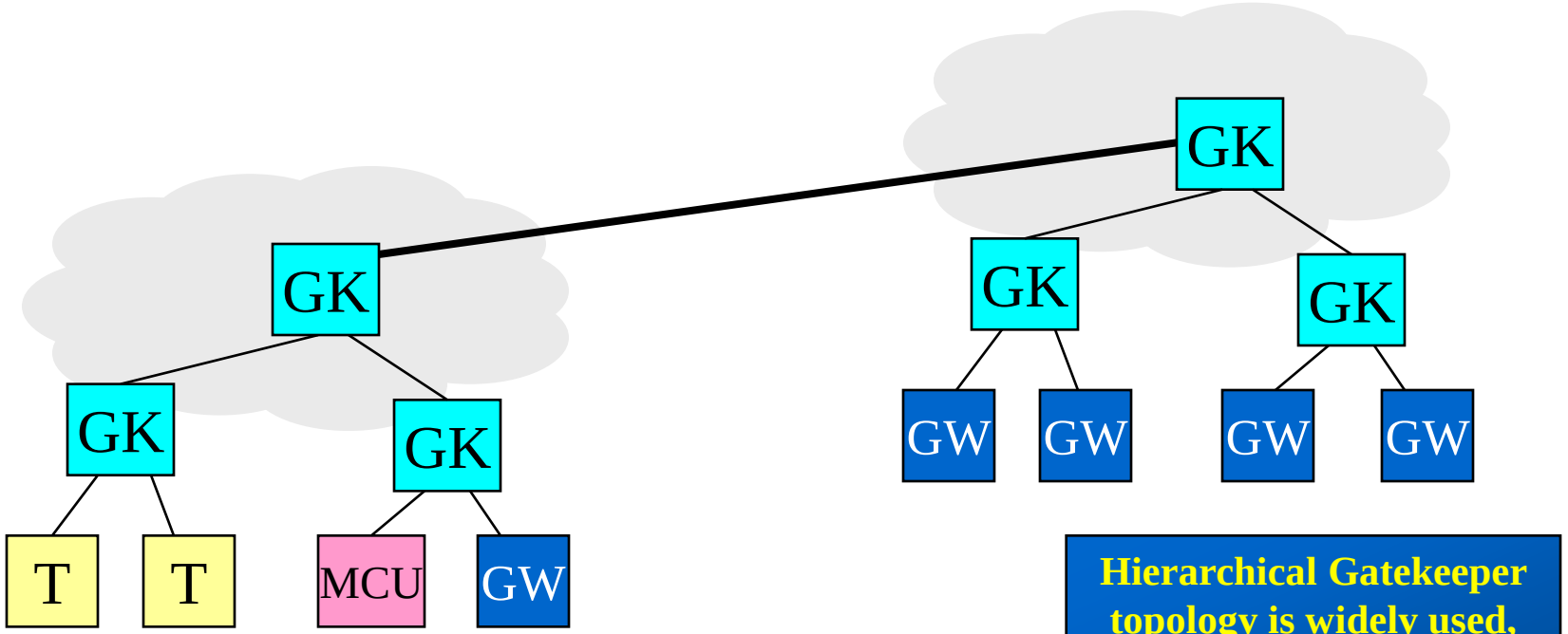
- A collection of Zones that are under a single administrative control (e.g., a service provider or enterprise network)



# Communication Between Administrative Domains

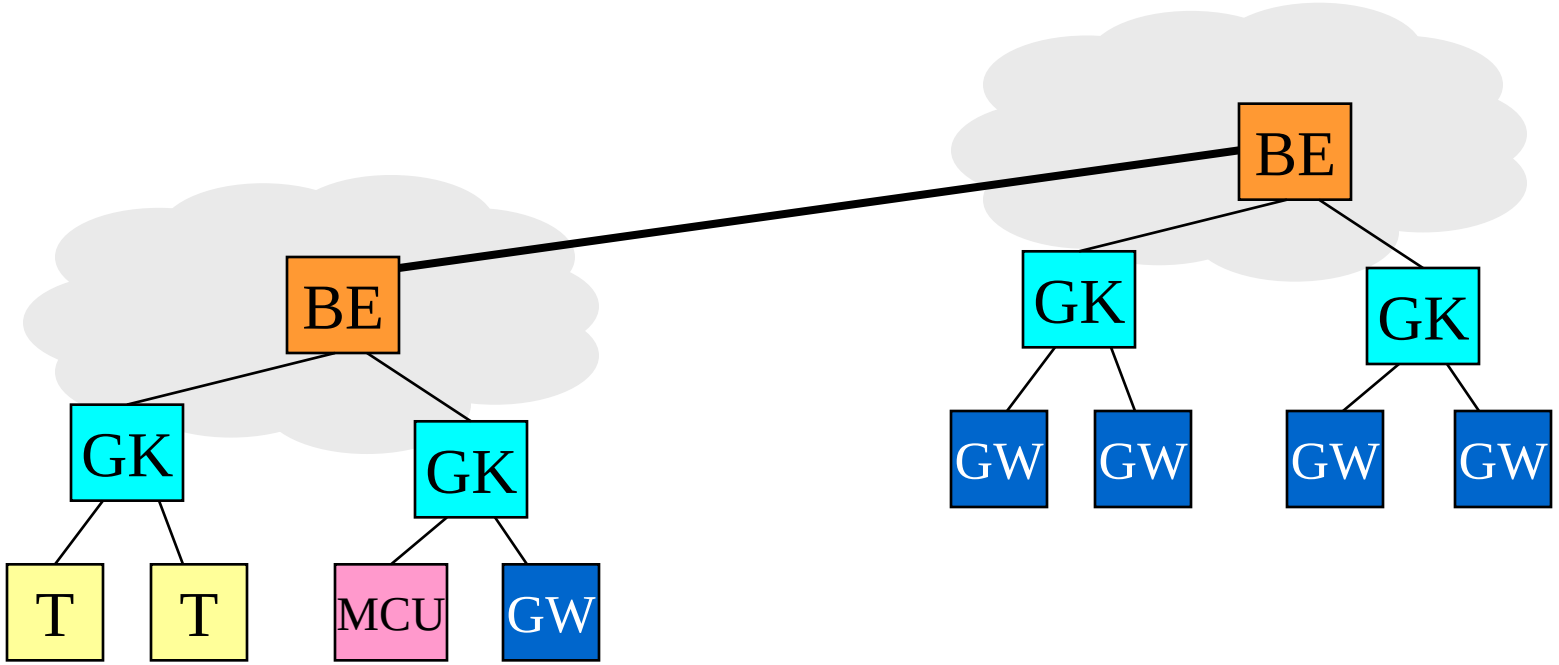
- Service providers or enterprise users have two options for communicating between Administrative Domains
  - RAS “Location Request” (LRQ) messages
  - Annex G/H.225.0
- Choice of protocols and topology is dictated by customer requirements
- Initial communication is generally for address resolution only
- Subsequent communication utilizes normal H.323 signaling between various elements to setup the call

# Topology with RAS

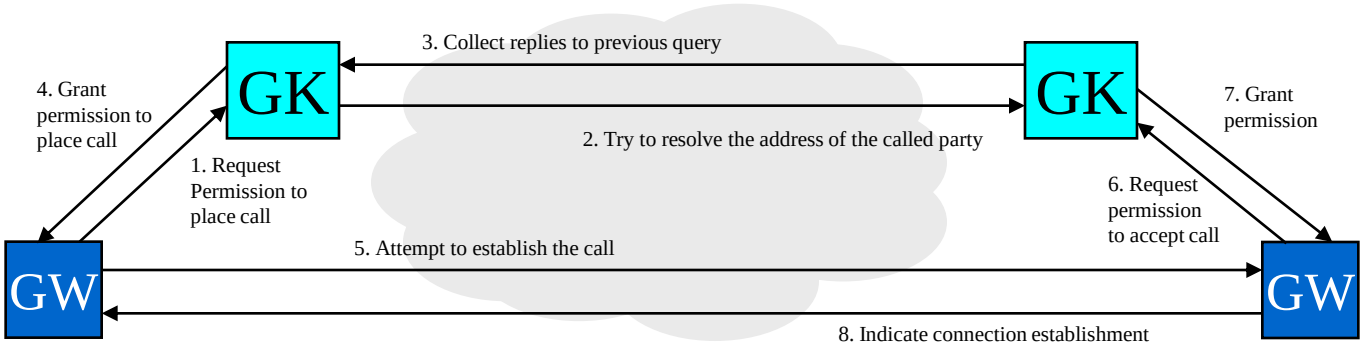



**Hierarchical Gatekeeper topology is widely used, but explicitly discussed in H.323**

# Topology with Annex G/H.225.0



# High Level Call Flow



# Part 3: ASN.1 Overview

# What is ASN.1?

- “Abstract Syntax Notation One”
- Protocol syntax language defined in ITU-T Recommendations X.680 – X.683
- Provides various encoding rules, including Packed Encoding Rules and XML encoding rules

# Why ASN.1?

- Separates the protocol syntax from the encoding of the protocol that is transmitted on the wire
- Facilitate development of protocols that are backward compatible
- Allows developers to focus on the program logic, rather than wasting time on parsing tasks

# Packed Encoding Rules

- The H.323 family of protocols use the Packed Encoding Rules (PER), as specified in ITU-T Recommendation X.691
- PER is a very efficient, binary encoding

# Modules

- Modules are syntax definitions for protocols
- The syntax in Annex H/H.225.0 is one “module” and the syntax for H.245 is one “module”
- One module can borrow definitions from another module

# The ASN.1 Module in H.225.0

```
H323-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
        SIGNED{},
        ENCRYPTED{},
        HASHED{},
        ChallengeString,
        TimeStamp,
        RandomVal,
        Password,
        EncodedPwdCertToken,
        ClearToken,
        CryptoToken,
        AuthenticationMechanism
    FROM H235-SECURITY-MESSAGES
        DataProtocolCapability,
        T38FaxProfile
    FROM MULTIMEDIA-SYSTEM-CONTROL;
```

```
--
-- The message and type definitions have been
-- omitted for brevity
--

END             -- Of ASN.1
```

# Common Integral Types

- INTEGER
- OCTET STRING
- NULL
- BOOLEAN
- IA5String
- OBJECT IDENTIFER
- BMPString
- GeneralString
- NumericString

# Object Identifiers

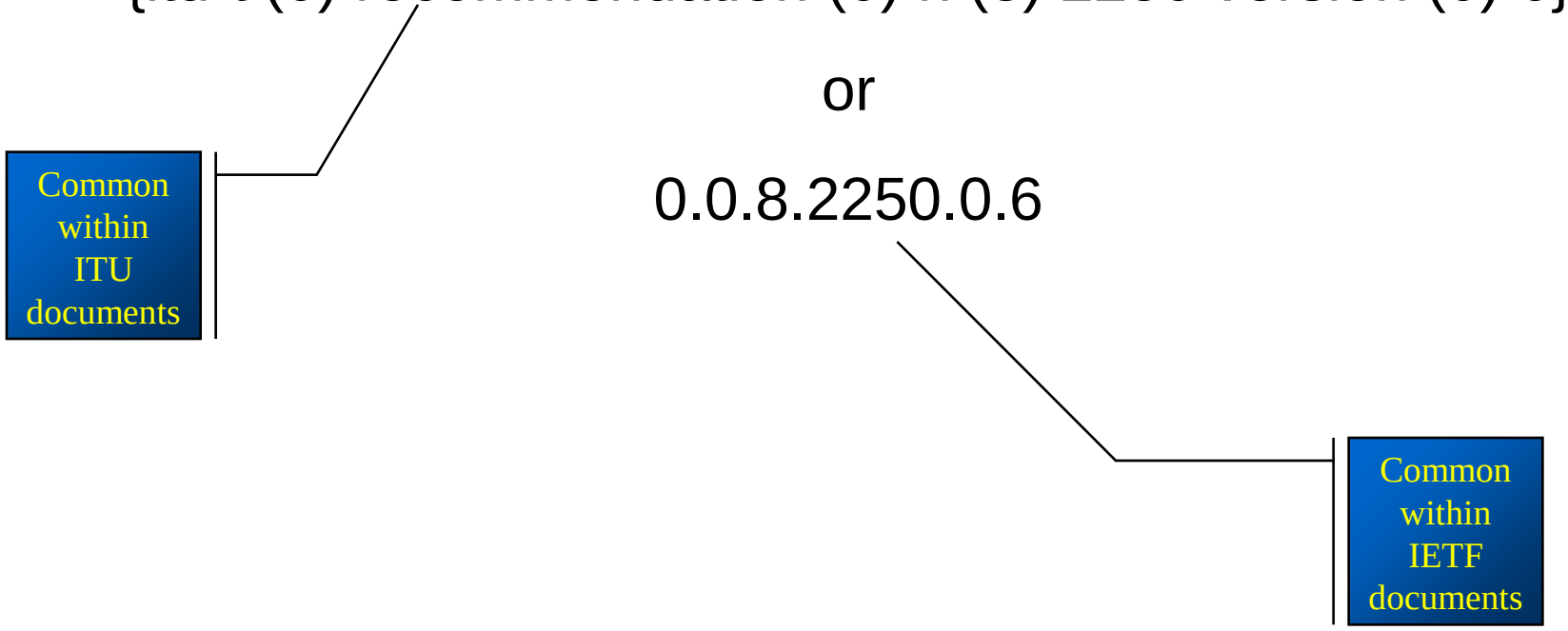
- Object Identifiers (OIDs) are a hierarchical arrangement of integers that refer to some object
- Used within LDAP and SNMP
- Used within H.323-based systems to refer to codecs, standard and non-standard protocol extensions, etc.
- Used to identify the particular version of H.225.0 or H.245 be transmitted (“protocol identifier”)

# Anatomy of an Object Identifier

{itu-t (0) recommendation (0) h (8) 2250 version (0) 6}

or

0.0.8.2250.0.6



# Common Syntax Structures

- SEQUENCE – think of C “struct”
- SET – think of C “struct”, but without order for the members of the structure
- CHOICE – think of a C “union”

# Samples from H.225.0

```
H323-UserInformation ::= SEQUENCE       -- root for all Q.931 related ASN.1
{
    h323-uu-pdu                     H323-UU-PDU,
    user-data SEQUENCE
    {
        protocol-discriminator      INTEGER  (0..255),
        user-information            OCTET STRING (SIZE(1..131)),
        ...
    } OPTIONAL,
    ...
}
```

# Samples from H.225.0 (cont.)

```
H323-UU-PDU ::= SEQUENCE
{
    h323-message-body   CHOICE
    {
        setup                   Setup-UUIE,
        callProceeding          CallProceeding-UUIE,
        connect                 Connect-UUIE,
        alerting                Alerting-UUIE,
        information             Information-UUIE,
        releaseComplete         ReleaseComplete-UUIE,
        facility                Facility-UUIE,
        ...,
        progress                Progress-UUIE,
        empty                   NULL,
        status                  Status-UUIE,
        statusInquiry           StatusInquiry-UUIE,
        setupAcknowledge        SetupAcknowledge-UUIE,
        notify                  Notify-UUIE
    },
    nonStandardData             NonStandardParameter OPTIONAL,
    ...,
    h4501SupplementaryService   SEQUENCE OF OCTET STRING OPTIONAL,
    h245Tunneling               BOOLEAN,
    h245Control                 SEQUENCE OF OCTET STRING OPTIONAL,
    nonStandardControl          SEQUENCE OF NonStandardParameter OPTIONAL,
    callLinkage                 CallLinkage OPTIONAL,
    tunnelledSignallingMessage  SEQUENCE
    {
        tunnelledProtocolID     TunnelledProtocol,
        messageContent          SEQUENCE OF OCTET STRING,
        tunnellingRequired      NULL OPTIONAL,
        nonStandardData         NonStandardParameter OPTIONAL,
        ...
    } OPTIONAL,
    provisionalRespToH245Tunneling    NULL OPTIONAL,
    stimulusControl             StimulusControl OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}
```

# Samples from H.225.0 (cont.)

Definition of new types

```
GloballyUniqueID        ::=    OCTET STRING (SIZE(16))
ConferenceIdentifier    ::=    GloballyUniqueID
RequestSeqNum           ::=    INTEGER (1..65535)
GatekeeperIdentifier    ::=    BMPString (SIZE(1..128))
BandWidth               ::=    INTEGER (0..4294967295)
CallReferenceValue      ::=    INTEGER (0..65535)
EndpointIdentifier      ::=    BMPString (SIZE(1..128))
ProtocolIdentifier      ::=    OBJECT IDENTIFIER
```

```
BandwidthRequest ::= SEQUENCE --(BRQ)
{
    requestSeqNum           RequestSeqNum,
    endpointIdentifier      EndpointIdentifier,
    conferenceID            ConferenceIdentifier,
    callReferenceValue      CallReferenceValue,
    callType                CallType OPTIONAL,
    bandWidth               BandWidth,
    nonStandardData         NonStandardParameter OPTIONAL,
    ...,
    callIdentifier          CallIdentifier,
    gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    answeredCall            BOOLEAN,
    callLinkage             CallLinkage OPTIONAL,
    capacity                CallCapacity OPTIONAL,
    usageInformation        RasUsageInformation OPTIONAL,
    bandwidthDetails        SEQUENCE OF BandwidthDetails OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL
}
```

# Using an ASN.1 Compiler and PER Encoder/Decoder

- Compile the “root” ASN.1 module, along with “dependent” modules (so that “IMPORTS” can be taken), and link the resulting source files with your application
- To send a message
  - Populate the data structures generated by the compiler
  - Call “encode()” and then send the resulting data
- When receiving a message
  - Call “decode()” to get appropriate data structures

# ASN.1 Resource Sites

- Electronic books on ASN.1:
  http://www.asn1.org/books/index.htm
- ASN.1 Information Site:
  http://asn1.elibel.tm.fr/
- ASN.1 Consortium:
  http://www.asn1.org/

# Part 4: RAS

# RAS

- Registration, Admission, and Status
- Used between the endpoint and its Gatekeeper in order to
  - Allow the Gatekeeper to manage the endpoint
  - Allow the endpoint to request admission for a call
  - Allow the Gatekeeper to provide address resolution functionality for the endpoint
- RAS signaling is required when a Gatekeeper is present in the network (i.e., the use of a Gatekeeper is conditionally mandatory)

# General Format of RAS

- RAS messages generally have three types
  - Request (xRQ)
  - Reject (xRJ)
  - Confirm (xCF)
- Exceptions are
  - Information Request / Response / Ack / Nak
  - The “nonStandardMessage”
  - The “unknownMessage” response
  - Request in Progress (RIP)
  - Resource Available Indicate / Confirm (RAI/RAC)
  - Service Control Indication / Response

# RAS Port

- Typically, RAS communications is carried out via UDP through port 1719 (unicast) and 1718 (multicast)
  - For backward compatibility sake, an endpoint should be prepared to receive a unicast message on port 1718 or 1719
  - Only UDP is defined for RAS communications
- GRQ and LRQ may be send multicast, but are generally sent unicast
- All other RAS messages are sent unicast

# Gatekeeper Request - GRQ

- When an endpoint comes to life, it should try to “discover” a gatekeeper by sending a GRQ message to a Gatekeeper
  - Address of a Gatekeeper may be provisioned
  - The endpoint may send a multicast GRQ
  - Address of a Gatekeeper may be found through DNS queries (Annex O/H.323)
- There may be multiple Gatekeepers that could service an endpoint, thus an endpoint should look through potentially several GCF/GRJ messages for a reply

# Contents of a GRQ

```
GatekeeperRequest ::= SEQUENCE
{
    requestSeqNum            RequestSeqNum,
    protocolIdentifier       ProtocolIdentifier,
    nonStandardData          NonStandardParameter OPTIONAL,
    rasAddress               TransportAddress,
    endpointType             EndpointType,
    gatekeeperIdentifier     GatekeeperIdentifier OPTIONAL,
    callServices             QseriesOptions  OPTIONAL,
    endpointAlias            SEQUENCE OF AliasAddress OPTIONAL,
    ...,
    alternateEndpoints       SEQUENCE OF Endpoint OPTIONAL,
    tokens                   SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
    authenticationCapability SEQUENCE OF AuthenticationMechanism OPTIONAL,
    algorithmOIDs            SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
    integrity                SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue      ICV OPTIONAL,
    supportsAltGK            NULL OPTIONAL,
    featureSet               FeatureSet OPTIONAL,
    genericData              SEQUENCE OF GenericData OPTIONAL
}
```

# Gatekeeper Reject - GRJ

- If a Gatekeeper does not wish to provide service to the endpoint, it will generally send a GRJ message to the endpoint
  - As a security consideration to avoid DoS attacks, one might want to consider ignoring requests from unknown endpoints
- The GRJ message will carry one of several rejection reasons

**As mentioned, most "request" messages have "reject" and "confirm" counterparts. We will discuss GRJ/GCF for illustration, but no special attention is given to other xRJ messages.**

# Contents of a GRJ

```
GatekeeperReject ::= SEQUENCE
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    gatekeeperIdentifier   GatekeeperIdentifier  OPTIONAL,
    rejectReason           GatekeeperRejectReason,
    ...,
    altGKInfo              AltGKInfo OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    featureSet             FeatureSet OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}
```

```
GatekeeperRejectReason ::= CHOICE
{
    resourceUnavailable        NULL,
    terminalExcluded           NULL,
    invalidRevision            NULL,
    undefinedReason            NULL,
    ...,
    securityDenial             NULL,
    genericDataReason          NULL,
    neededFeatureNotSupported  NULL
}
```

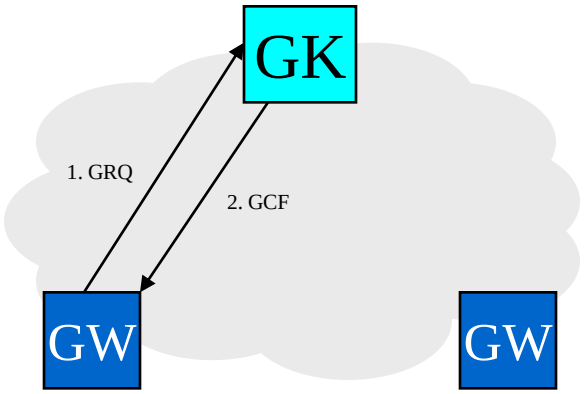
# Gatekeeper Confirm - GCF

- If the Gatekeeper wishes to provide service to the endpoint, it will return a GCF message
- The GCF message will contain a number of data elements that will later be used by the endpoint

# Contents of a GCF

```
GatekeeperConfirm ::= SEQUENCE
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    gatekeeperIdentifier   GatekeeperIdentifier  OPTIONAL,
    rasAddress             TransportAddress,
    ...,
    alternateGatekeeper    SEQUENCE OF AlternateGK OPTIONAL,
    authenticationMode     AuthenticationMechanism OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    algorithmOID           OBJECT IDENTIFIER OPTIONAL,
    integrity              SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    featureSet             FeatureSet OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}
```

# Gatekeeper Discovery




**Communication is over ports 1718 (multicast) and, most commonly, 1719 (unicast)**

# Gatekeeper Registration - RRQ

- Once a Gatekeeper has been “discovered”, the endpoint will then register with the Gatekeeper in order to receive services
- Communication is exclusively via port 1719 (unicast)
- Endpoint will send an RRQ and expect to receive either an RCF or RRJ
- Reception of an RRJ simply means that the endpoint will not receive services from the Gatekeeper, not that the endpoint cannot communicate on the network

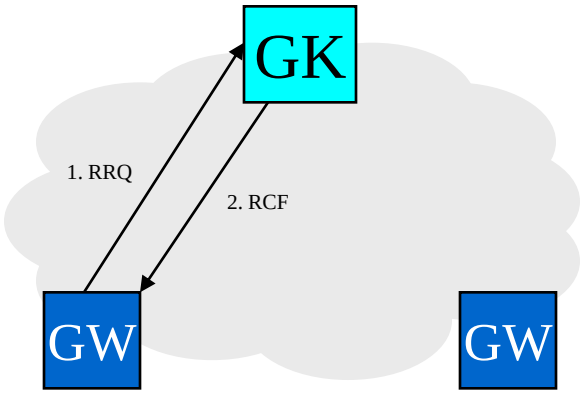
# Gatekeeper Registration (cont.)

- During the registration process, the Gatekeeper will assign an “endpoint identifier” to the endpoint, which is to be used during subsequent communications with the Gatekeeper
- The endpoint will supply a list of endpoint alias addresses and the Gatekeeper will indicate which ones it accepts
- The Gatekeeper may grant the endpoint permission to place calls without using the ARQ/ACF exchange (called “pre-granted ARQs”)
- The endpoint will indicate a “time to live” and the Gatekeeper may accept that or a lower TTL value

# Lightweight RRQs

- The “time to live” indicated in the RRQ tells the Gatekeeper when it may freely unregister the endpoint due to inactivity
- The endpoint may renew its registration by sending either a full RRQ message or a “lightweight RRQ” (LW RRQ)
- The LW RRQ message only contains a few elements and is only intended to refresh the endpoint’s registration

# Gatekeeper Registration (cont.)




**Communication is exclusively over port 1719 (unicast)**

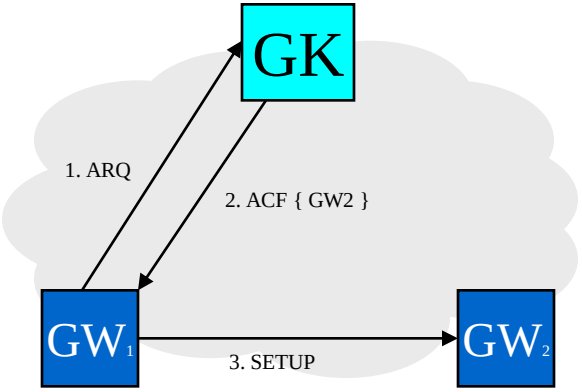
# Admission Request - ARQ

- Once registered with a Gatekeeper, the endpoint may only initiate or accept a call after first requesting “admission” to the Gatekeeper via the ARQ message (except in the case that “pre-granted ARQs” is in use)
- The Gatekeeper may may accept (ACF) or reject (ARJ) the request to place or accept a call
- The endpoint will indicate the destination address(es) and the Gatekeeper may (if “canMapAlias” is true) return an alternate set of destination addresses
- The endpoint uses a unique “call reference value” (CRV) between itself and the GK to refer to this call (link significant)

# Admission Request (cont.)

- The endpoint will provide a Call Identifier (CallID), which is a globally unique value
- The endpoint will indicate a conference ID (CID), or 0 if the conference ID is not known
  - This is unique if the call is point to point
  - This value is shared by all participants in the same multipoint conference
  - Some devices do not properly handle CID=0
- The endpoint will indicate the desired bandwidth and the Gatekeeper may adjust that value to a lower value
- The endpoint will indicate whether it is originating or answering a call

# Admission Request (cont.)



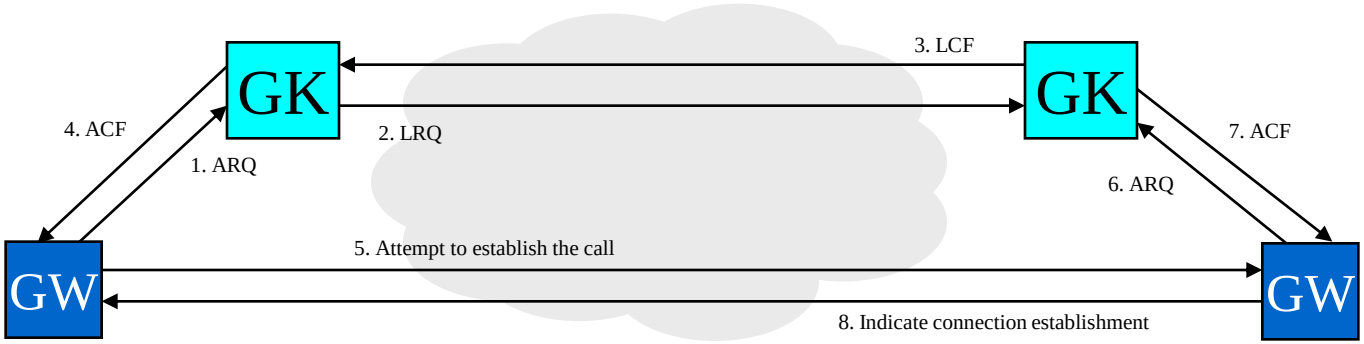
# Reference Value Soup

- CRV – Call Reference Value
  - 16 bit integer that is link significant
  - most significant bit indicates the direction of the call (0 = originator, 1 = terminator)'
  - CRV used in the ARQ to originate a call must be the subsequent SETUP message (interoperability with version 1)
  - CRV used in ARQ to answer a call does *not* have to be the same as that received in SETUP
- CID – Conference Identifier
  - A Globally unique (UUID or GUID) identifier that is shared by all participants in the same conference
- CallID – Call Identifier
  - A globally unique (UUID or GUID) identifier that is unique to that particular call

# Location Request - LRQ

- The LRQ message is sent by either an endpoint or a Gatekeeper to a Gatekeeper in order to resolve the address of an alias address (e.g., to turn a telephone number into an IP address)
- While LRQs may be sent by endpoints, they are almost exclusively sent by Gatekeepers

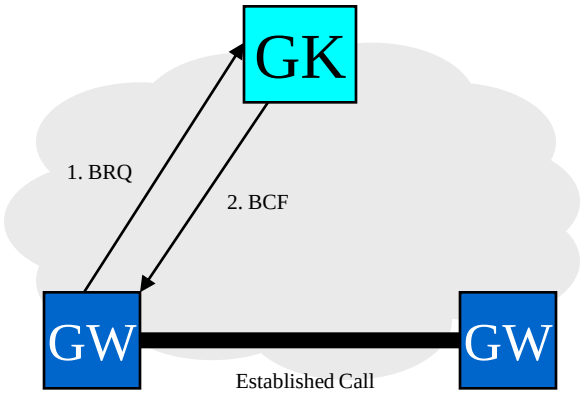
# Location Request (cont.)



# Bandwidth Request - BRQ

- Subsequent to initial call setup, the endpoint may wish to to use more or less bandwidth than previously indicated via the BRQ
  - Note that, while it is syntactically legal for the GK to send a BRJ to a request asking for less bandwidth, this makes no sense and should not be done
- An endpoint must send a BRQ subsequent to initial call establishment if the actual bandwidth utilized is less than initially requested

# Bandwidth Request (cont.)



# Disengage Request - DRQ

- Once a call completes, the endpoint sends a DRQ message to the Gatekeeper
  - The Gatekeeper may send a DRJ, but this is strongly discouraged… if an endpoint is sending a DRQ, it means the call is over and cannot be “rejected”!
- The DRQ is an opportunity for the endpoint to report information useful for billing
- The Gatekeeper may also send a DRQ to force the endpoint to disconnect the call

# Information Request - IRQ

- The IRQ is sent by the Gatekeeper to the endpoint to request information about one or all calls
- There are many details about each call that are reported to the Gatekeeper in the Information Response (IRR) message
- There are provisions in H.323 to allow the endpoint to provide call information periodically and unsolicited
- The Gatekeeper may acknowledge or provide negative acknowledgement to an unsolicited IRR

# Request In Progress - RIP

- A RIP message may be sent by the endpoint or the Gatekeeper to acknowledge receive of a RAS message that cannot be responded to in normal processing time

# Resource Availability - RAI

- The “Resource Available Indicate” (RAI) message is sent by an endpoint to indicate when it has neared resource limits or is no longer near a resource limit
- The Gatekeeper replies with “Resource Available Confirm” (RAC)

# Service Control Indication - SCI

- This message is sent by either the endpoint or the Gatekeeper to invoke some type of service
- The responding entity replies with “Service Control Response” (SCR)
- The SCI/SCR messages are used for specific services that are and will be defined for H.323, including Gatekeeper requested tones and announcements and “stimulus control” (Annex K/H.323)

# Miscellaneous Messages

- “Unknown Message Response” is sent to an unrecognized message
- “Non-Standard Message” is used to allow Gatekeepers and endpoints to exchange messages that are not standard

# RAS Timers and Retries

| RAS message | Time-out value (s) | Retry count |
|---|---|---|
| GRQ | 5 | 2 |
| RRQ | 3 | 2 |
| URQ | 3 | 1 |
| ARQ | 5 | 2 |
| BRQ | 3 | 2 |
| IRQ | 3 | 1 |
| IRR | 5 | 2 |
| DRQ | 3 | 2 |
| LRQ | 5 | 2 |
| RAI | 3 | 2 |
| SCI | 3 | 2 |

# Part 5: Annex G/H.225.0

# Standards

- Annex G/H.225.0 – “Communication Between And Within Administrative Domains”
- The actual protocol specification exists in H.501
- Annex Gv1 (1999) focused on communication between administrative domains
- Annex Gv2 (2002) expanded the scope to include communication within administrative domains

# Role

- Annex G was originally designed with the clearinghouse model in mind, wherein calls originate in one "administrative domain" and terminate in another
- That role was then expanded to include generally useful address resolution functions, propagation of routing information, etc.
- Two new functional entities were introduced into the H.323 network
  - Peer Element
  - Border Element

# RAS LRQ versus Annex G

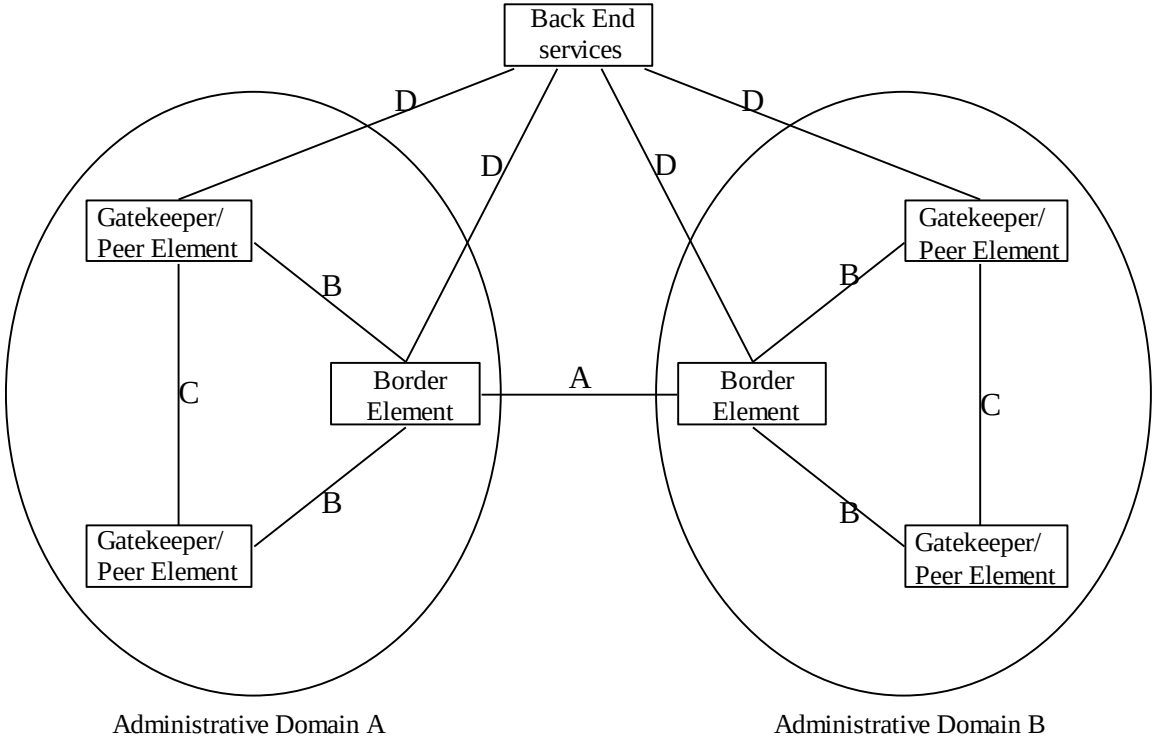
RAS

- Address resolution

Annex G/H.225.0

- Address resolution
- Propagation of routing information
- Usage reporting
- Access authorization (useful for clearinghouses and non-clearinghouse environments)
- A Border Element or Peer Element can act as an aggregation point (usually the role of a Border Element)

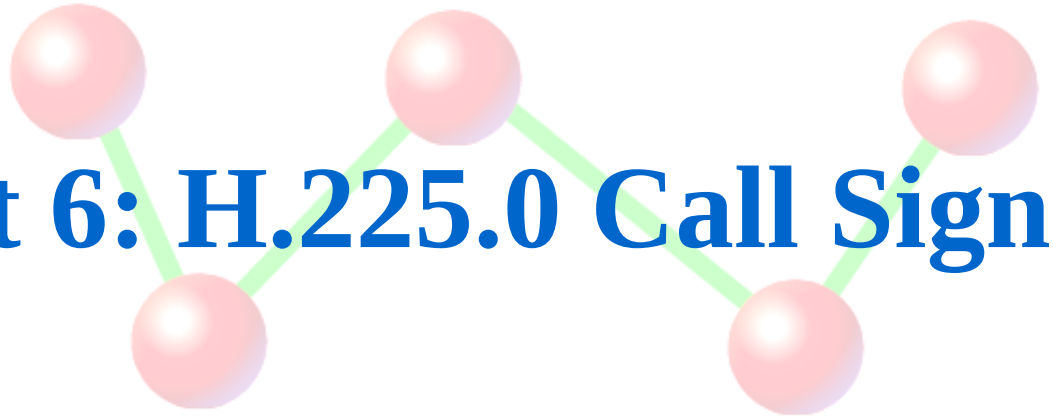
# Architecture




**Interface "D" is out of Scope**

# H.501 Messages Supported by Annex G/H.225.0 Entities

- ServiceRequest
- ServiceConfirmation
- ServiceRejection
- ServiceRelease
- DescriptorRequest
- DescriptorConfirmation
- DescriptorRejection
- DescriptorIDRequest
- DescriptorIDConfirmation
- DescriptorIDRejection
- DescriptorUpdate
- DescriptorUpdateAck
- AccessRequest
- AccessConfirmation
- AccessRejection
- RequestInProgress
- NonStandardRequest
- NonStandardConfirmation
- NonStandardRejection
- UnknownMessageResponse
- UsageRequest
- UsageConfirmation
- UsageRejection
- UsageIndication
- UsageIndicationConfirmation
- UsageIndicationRejection
- ValidationRequest
- ValidationConfirmation
- ValidationRejection



# Part 6: H.225.0 Call Signaling

# Introduction

- H.225.0 Call Signaling is used to establish calls between two H.323 entities
- It was derived from Q.931 (ISDN call signaling), but was modified to be suitable for use on a packet based network
- ASN.1 was added to augment to Q.931 information and is stored in the “User to User” Information Element from Q.931
- H.225.0 also borrows messages from Q.932

# H.225.0 Call Signaling Message


The UUIE refer to the "User-User Information Element". It should be the last octet in the chain, but some implementations do not properly order IEs. It is comprised of 0x7E, HH, LL, PD, and DATA. 0x7E is the identifier for the User-User IE, HH and LL are the length of DATA in network byte order, PD is a protocol discriminator for ASN.1 (0x05) and DATA is the ASN.1 PER encoded "H323-UserInformation".

Various Information Elements (IEs) that are appropriate for the message type. These are listed in H.225.0, but note that any valid Q.931 IE may be transmitted and should not result in a protocol failure by the endpoint.

All messages have a Q.931 header that includes a single octet called the "protocol discriminator" (0x08), three octets for the CRV (0x02, HH, LL, where 0x02 is the length of the CRV and HH and LL are the two octets of the CRV in network byte order), and single octet for the message type (specified in respective sections in Q.931).

Four octets that separate messages on the wire (necessary for TCP). They are defined in section 6 of RFC 1006. There are 0x03, 0x00, HH, LL. HH and LL represent the entire message length, including the TPKT header, in network byte order.

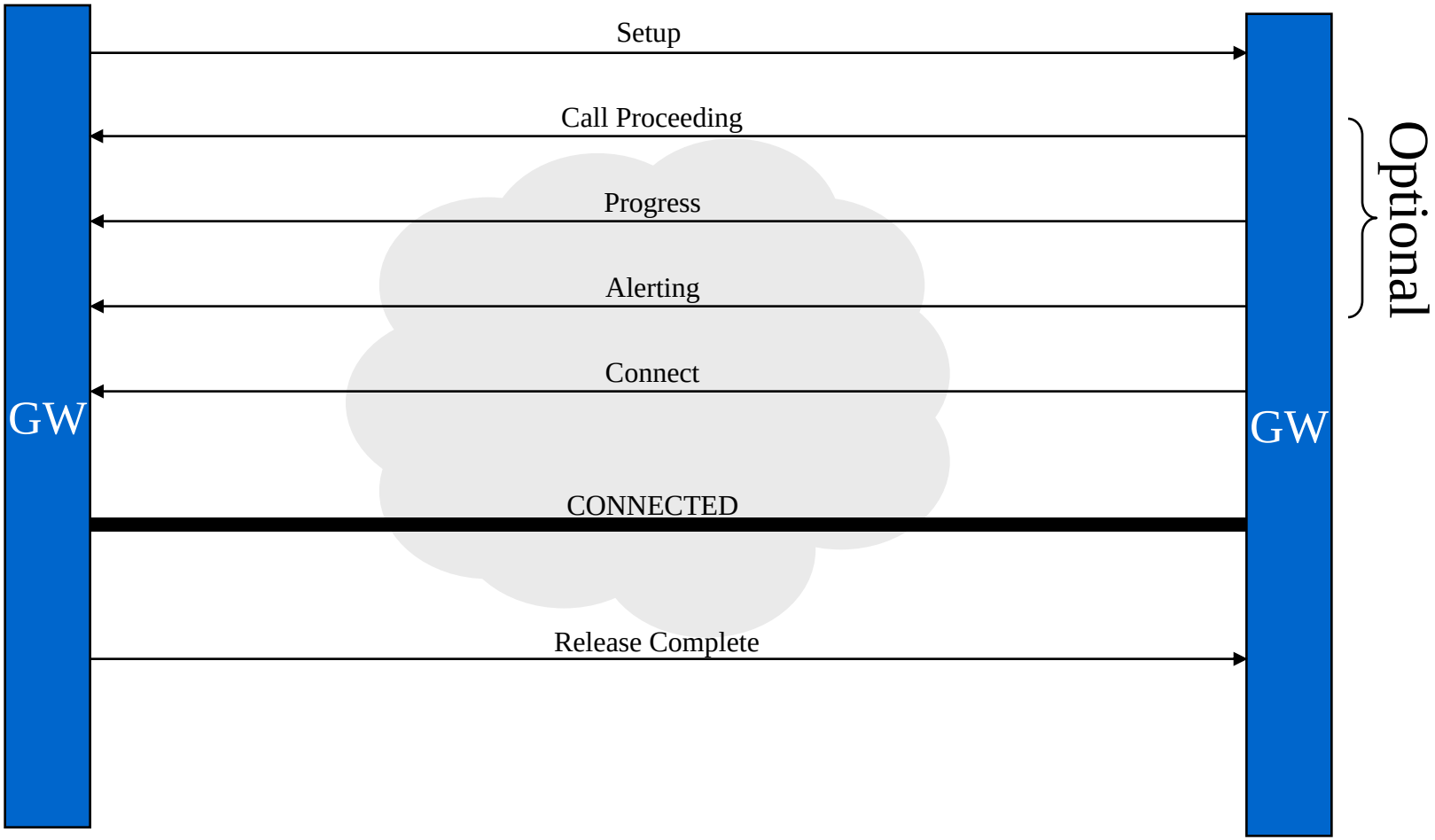
# Information Elements

- Information elements carry additional information related to the specific message
- For example, SETUP has, among other things, a “Calling Party Number” IE, “Called Party Number” IE, “Display” IE, etc.
- Every H.225.0 message has a UUIE, though this is not true of Q.931
- H.225.0 made a number of changes to Q.931 and should be the guiding document

# H.225.0 Call Signaling Messages

- Setup
- Call Proceeding
- Alerting
- Information
- Release Complete
- Facility
- Progress
- Status
- Status Inquiry
- Setup Acknowledge
- Notify
- Connect

# Basic Call Setup Signaling



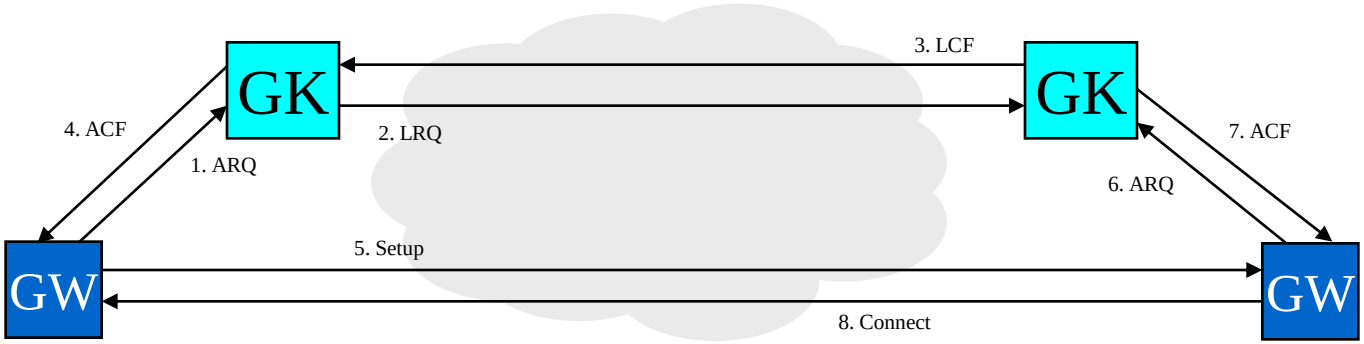
# Comments on Call Establishment

- The basic call setup procedures are pretty straight forward
- The setup procedure can be as simple as “Setup” and “Connect”
- Intermediate messages (labeled as optional on the previous slide) are generally useful to prevent timeout errors and to provide in-band tones and announcements

# Progress Message and Progress Indicator

- When a user places a call, he or she expects to hear a ringing tone or an announcement providing some information about why the call failed
- These “in-band tones and announcements” are provided by using the Progress message and the Progress Indicator IE (PI)
- Section 8.1.7.4/H.323 describes this more fully

# RAS and H.225.0 Call Signaling



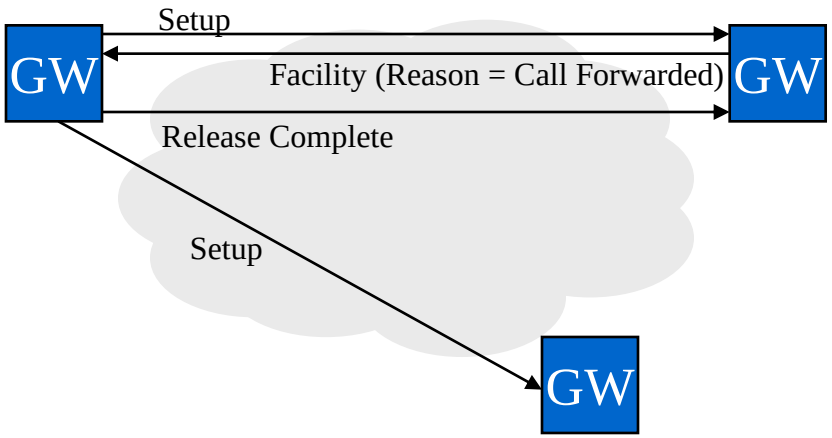
# Overlapped Sending

- In some cases, the user may not have entered a complete telephone number
- Overlapped sending allows the calling endpoint to provide additional dialed digits to the called endpoint during the call establishment procedure
- Overlapped sending is generally most useful in H.225.0 Call Signaling, but RAS also allows for overlapped sending (refer to 8.1.12/H.323 for details)

# Call Forwarding

- A Facility message with reason “callForwarded” allows for simple call redirection
- The H.323 standard states that this shall only be used for forwarding a call prior to “connect”
- In reality, many vendors use it as a lightweight means of performing a call transfer operation
- H.450.2 more fully describes a call transfer mechanism for H.323 systems

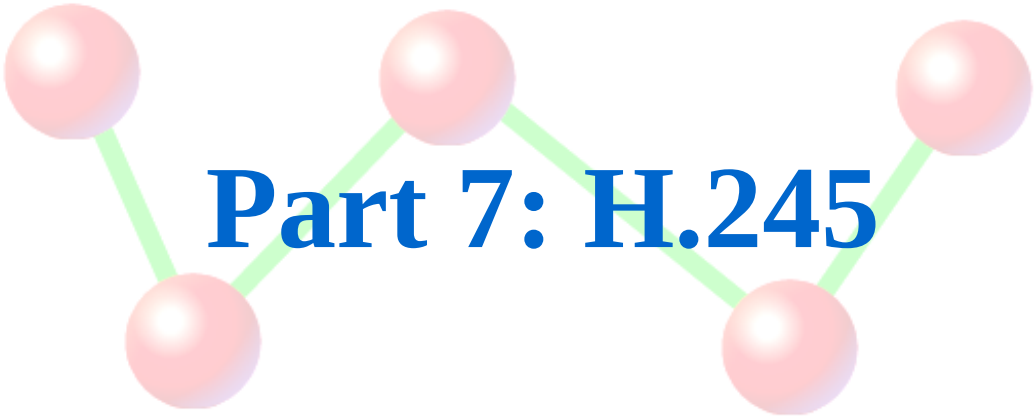
# Call Forwarding (cont.)




**Sometimes called "Facility Call Transfer"**

# H.450.x

- H.450.x is a whole series of supplementary service specifications for H.323
- H.450.1 is not a “service”, per se, but is the document upon which all services are based
- This series of document provide services like call transfer, call park, call hold, message waiting indication, etc.



# Part 7: H.245

# Comments on H.245

- H.245 is a protocol shared by a number of H.32x series protocols, including H.324M, which is used for multimedia conferencing within 3GPP wireless networks
- Like Q.931, not everything inside H.245 is applicable to H.323
- Refer to Annex A/H.323 for H.245 messages used by H.323 endpoints
- There are a *lot* of H.245 messages… but don't let that scare you
- H.245 signaling is intended to be carried out *in parallel* to H.225.0 signaling and preferably before the CONNECT message… waiting for the CONNECT will delay media establishment and result in media clipping

# Purpose

- H.245 provides “control” to the multimedia session that has been established
  - Terminal capability exchange
  - Master/Slave determinations
  - Logical channel signaling
  - Conference control

# H.245 Control Channel

- H.245 messages are carried via a special "channel" called the H.245 Control Channel
- Opening the H.245 Control Channel is optional (see later discussion on "Fast Connect")
- The H.245 channel is often a separate TCP connection, but it may be "tunneled" inside of the H.225.0 Call Signaling Channel
- When using UDP for call signaling, the H.245 Control Channel *must* be tunneled inside the H.225.0 call signaling channel

# H.245 Message


Additional H.245 PDUs may be encoded following the first one. However, many implementations cannot handle this and, as such, it is ill-advised to place them end-to-end like this. It is strongly recommended to place only one between each TPKT header, but do be prepared for the case that more than one PDU does exist following TPKT

H.245 messages are encoded in ASN.1 PER and follow the TPKT header in the H.245 Control Channel.

Four octets that separate messages on the wire (necessary for TCP). They are defined in section 6 of RFC 1006. There are 0x03, 0x00, HH, LL. HH and LL represent the entire message length, including the TPKT header, in network byte order.

# H.245 Tunneling

- H.245 is generally transmitted on a separate TCP connections by most older endpoints
- Newer endpoints generally support “H.245 Tunneling”, which is the ability to place the H.245 PDUs inside the H.225.0 Call Signaling channel
- When tunneling, TPKT is not used
- Multiple H.245 PDUs may be tunneled in a single H.225.0 message

# Four H.245 Message Types

(and examples of each)

- Request
  - masterSlaveDetermination
  - terminalCapabilitySet
- Response
  - masterSlaveDeterminationAck
  - terminalCapabilitySetAck
- Command
  - sendTerminalCapabilitySet
- Indication
  - userInput

# Capabilities Exchange

- The capability exchange (or “caps exchange”) allows two endpoints to exchange information about what media capabilities they possess, such as G.711, G.723, H.261, and H.263
- Along with the type of media, specific details about the maximum number of audio frames or samples per packet is exchanged, information about support for silence suppression (VAD), etc. are exchanged
- Using this capability information, endpoints can select preferred codes that are suitable to both parties
- The terminalCapabilitySet (TCS) **must** be the first message transmitted on the H.245 Control Channel

# Capabilities are Numbered

- Each capability is numbered in a "capability table"
- All attributes (VAD, frames/packet, etc.) are part of the the capability in the table

Sample Capability Table

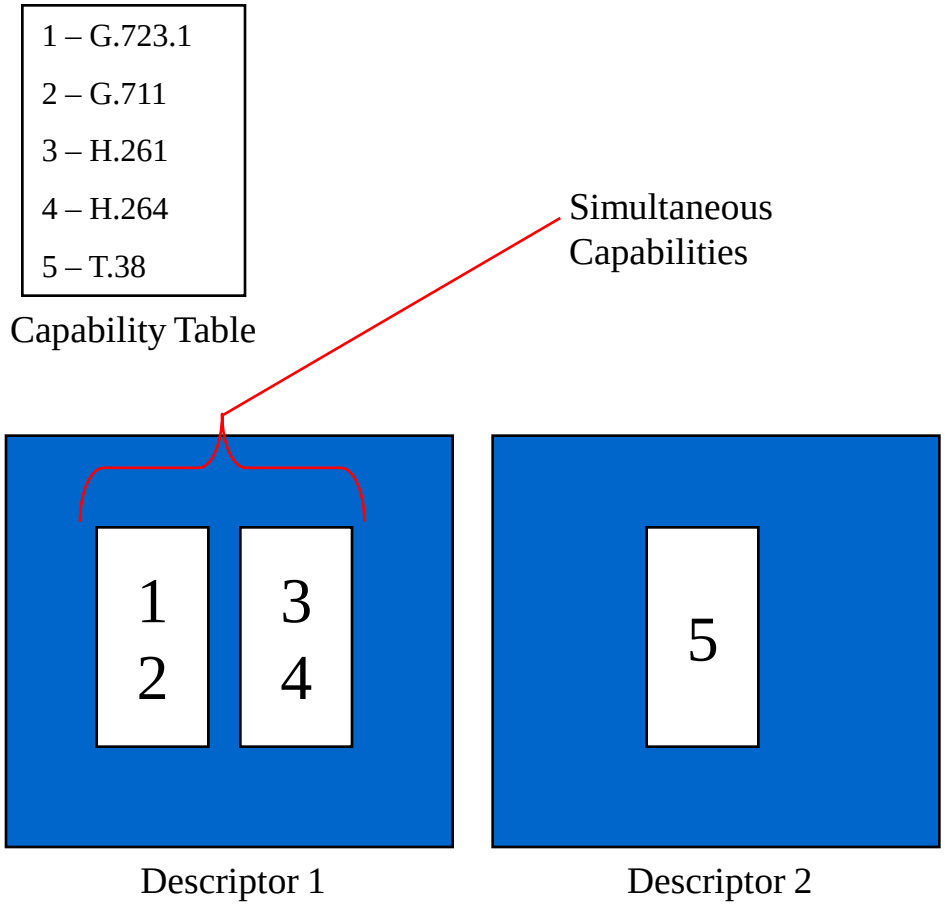
| |
|---|
| 1 – G.723.1 |
| 2 – G.711 |
| 3 – H.261 |
| 4 – H.264 |
| 5 – T.38 |

# Simultaneous Capabilities

- When endpoints advertise capabilities, they also advertise which capabilities may be performed simultaneously
- It may not be possible, for example, for an endpoint to open a T.38 channel at the same time as a V.150.1 channel
- It may not be possible, due to bandwidth limits, to open a high bit-rate video codec at the same time as a high bit-rate audio codec

# Capability Descriptors

- The capability descriptor contains the sets of simultaneous capabilities
- Only one descriptor may be used at a time (i.e., capabilities from descriptor 1 and descriptor 2 may not be used simultaneously)



# Master Slave Determination

- Once capabilities are exchanged, the endpoints negotiate master and slave roles
  - Actually the master/slave messages can be sent along with the TCS message
- The master in a point to point conference really only has the power to indicate when channels are in conflict (e.g., when one the other terminal tries to open a channel that is not compatible)
- The slave device must yield to the requests of the master device and reconfigure channels appropriately

# Logical Channel Signaling

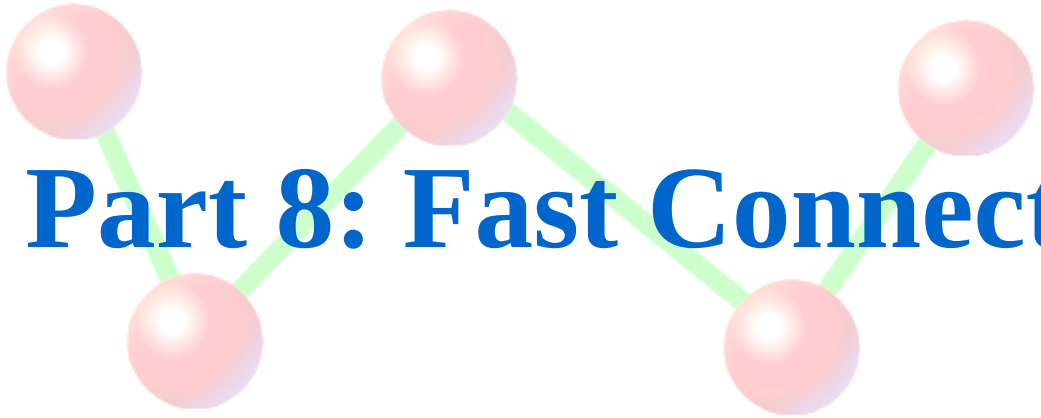
- Channels are opened by exchanging “openLocalChannel” (OLC) messages
- The OLC will contain one of the capabilities that was previously advertised by the other endpoint
- Voice and video channels are “unidirectional”, so each end must transmit an OLC to open a logical channel

# Logical Channel Signaling (cont)

- Within the OLC, a “session ID” is assigned
- Session 1 is the default audio session, 2 is the default video session, and 3 is the default data session
- Additional session IDs may be used, but are assigned by the master in the call
- There is a relationship between H.245 sessions IDs and RTP: OLCs with the same session ID are considered to be part of the same RTP/RTCP session

# Closing the H.245 Control Channel

- H.323 specifies that, in order to close the H.245 Control Channel, the endpoint must:
  - Close all open logical channels
  - Wait for all acknowledgement messages
  - Send an “endSession” command
  - Wait for an “endSession” from the other side
- In reality, most endpoint vendors don’t bother– they just use the H.225.0 Release Complete command to terminate the call and close the H.245 Control Channel, as that is much more efficient



# Part 8: Fast Connect

# What is Fast Connect

- Fast Connect (also improperly referred to as “fast start”, after the name of the associated field) is a means of establishing an H.323 call with as few as two messages
- With the use of Fast Connect, there is no need to open an H.245 channel, as long as all needed media can be negotiated via Fast Connect

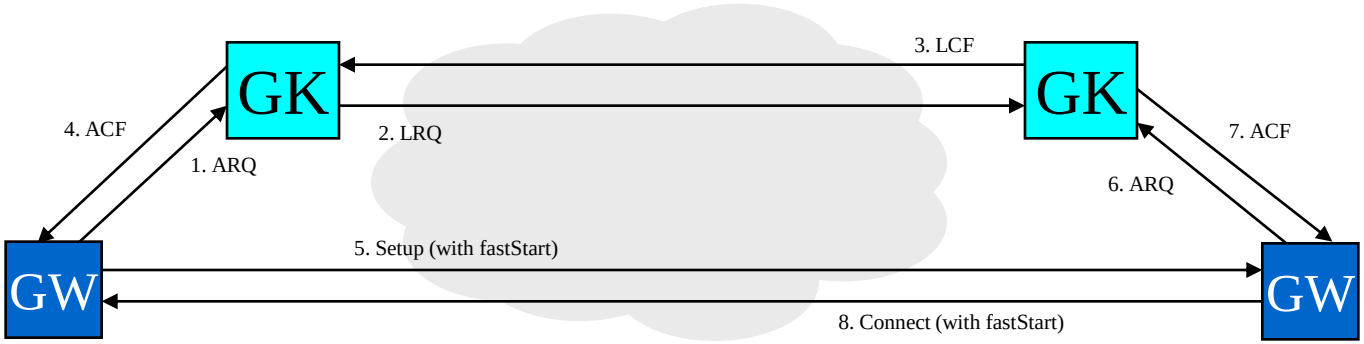
# Initiating Fast Connect

- When transmitting a SETUP message, the endpoint will populate the fastStart element with OpenLogicalChannel messages from H.245 (Note: these are just the data structures, not the entire H.245 “message”)
- Each OLC represents a proposed channel in either the “forward” (transmitting from the caller to the called party) or the “reverse” (transmitting from the called to the calling party) direction
- Each OLC with the same session ID number is considered alternate “proposals” (i.e., if two proposals are made for session ID 1, only one of the two may be selected)

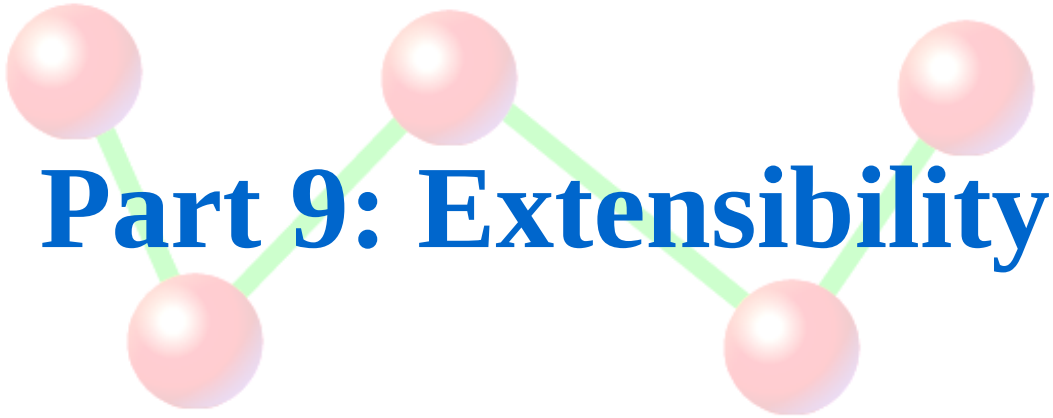
# Responding to Fast Connect

- To “accept” Fast Connect, an endpoint may select any fastStart element in the SETUP message, populate the necessary data fields (as specified in H.323), and return a fastStart element in any message to the caller
- An endpoint “rejects” Fast Connect by either explicitly indicating so (there is a flag for this), initiating any H.245 communications, or providing an H.245 address for the purposes of initiating H.245 communications
- Until Fast Connect is accepted or rejected, the calling endpoint may not initiate H.245 procedures (there is a exception to this rule in H.323v4 designed to avoid race conditions that would otherwise exist)

# A Fast Connect Call




**Of course, there are generally intermediate H.225.0 messages– fastStart should be returned as quickly as possible**



# Part 9: Extensibility

# Nonstandard Data

- H.323 may be extended by vendors, service providers, or national bodies by using the “non-standard” data fields found throughout H.323 and H.245
- The nonstandard data fields are essentially OCTET STRINGS that be populated with any kind of data
- Most often, vendors will create ASN.1 data structures that are encoded within the non-standard data elements
- The nonstandard data can be identified by object identifier, vendor identifier
  - Vendor identifiers are comprised of a T.35 country, extension (nationally assigned), and manufacturer code (nationally assigned)

# Generic Extensibility Framework

- The Generic Extensibility Framework (GEF) was introduced in H.323v4 to address the need to extend H.323 with features that are not necessarily of horizontal interest
- One can think of GEF as a means of extending H.225.0 in such a way as to not grow the base H.225.0 or H.323 documents, but still provide the same desired capability as if such extensions had been added to the base documents
- GEF capabilities (or “feature sets”) may be advertised, signaled as “desired” or “required”, etc.
- If a caller indicates that it requires feature X, but the called entity cannot provide feature X, the call should not complete

# GEF Documents

- GEF-based extensions to H.323 are defined in a series of documents number H.460.x
- H.460.6, as an example, extends the concept of Fast Connect in such a way that media channels may be re-negotiated, closed, and then re-opened on-the-fly without ever starting H.245 procedures– very useful for routing to an IVR, then re-routing to a final destination

# Part 10: TCS=0

# Third Party Pause and Re-Routing

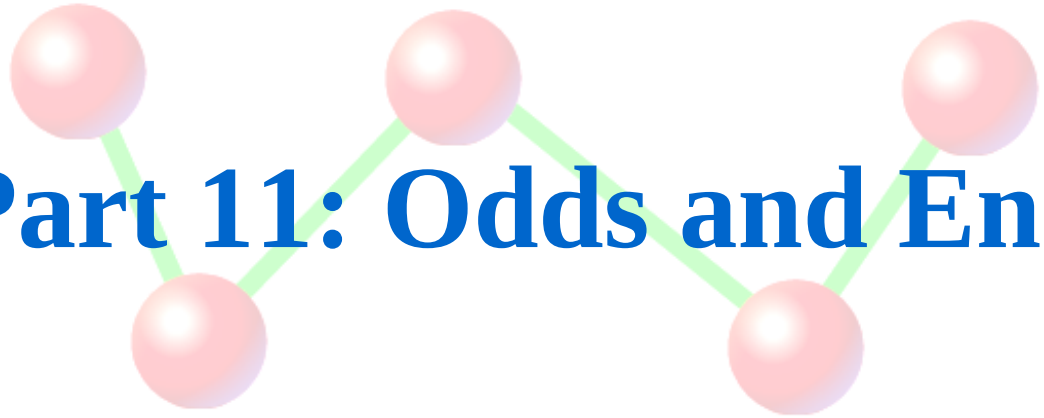
- H.323v2 defined a mechanism whereby a third party may “pause” an endpoint and then re-route the call
- This is done be sending an “empty capability set” (TCS=0 or ECS)
- The reception of this results in the receiver closing its transmit channels and awaiting to be awoken
- While it is “paused” the re-routing entity may re-route the call

# Leaving the Paused State

- Once re-routed, the re-routing entity will transmit a non-empty capability set, which indicates the capabilities of the new remote endpoint
- Upon leaving the paused state, the endpoint shall “reset” its H.245 state machine and re-negotiate master/slave, open channels, etc.
- When leaving the paused state, a TCS message is not sent, as it is considered the responsibility of the re-routing entity

# Poor Man's Call Hold

- Just as a Facility message with reason "callForwarded" is used to "transfer" a call, some use TCS=0 as a means of putting an endpoint on hold
- It works, but there is a more complete "Hold" service specified in H.450.4



# Part 11: Odds and Ends

# Alternate Gatekeepers

- For the purposes of redundancy, H.323 has the notion of “alternate Gatekeepers”
- When an endpoint registers with its Gatekeeper, it will be provided with a list of alternate Gatekeepers
- If the primary Gatekeeper fails, the endpoint switches over to an alternate Gatekeeper and continues communication
- More than one alternate Gatekeeper may be provided to the endpoint

# Capacity Reporting

- So as to prevent calls from being directed to endpoints (especially Gateways) that are at or near capacity, endpoints have the wherewithal to report their call capacity
- The capacity information shared includes the maximum number of calls and the current number of calls
- This is a complementary feature to the RAI message in RAS

# Alias Addresses

- H.323 offers a number of alias address types
  - dialedDigits (formerly called e164)
  - h323-ID
  - url-ID
  - transportID
  - email-ID
  - partyNumber
  - mobileUIM
- Of these, dialedDigits is the most widely used for placing calls (as users today generally still use telephone numbers)
- h323-IDs are of local significance and are usually just used between an endpoint and its Gatekeeper
- url-ID (including the H.323 URL and “tel” URL) and e-mail addresses are becoming more popular

# Informative H.323 Sites




- Packetizer
  http://www.packetizer.com/
- H.323 Forum
  http://www.h323forum.org/
- H.323 Plus
  http://www.h323plus.org/


Packetizer®