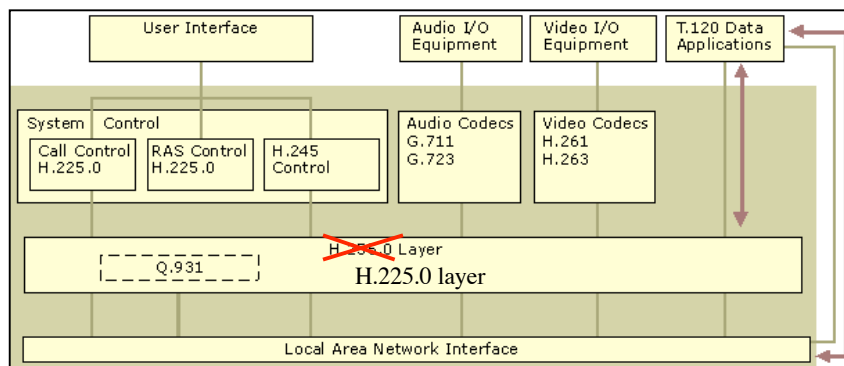


Z24: Signalling Protocols

Mark Handley

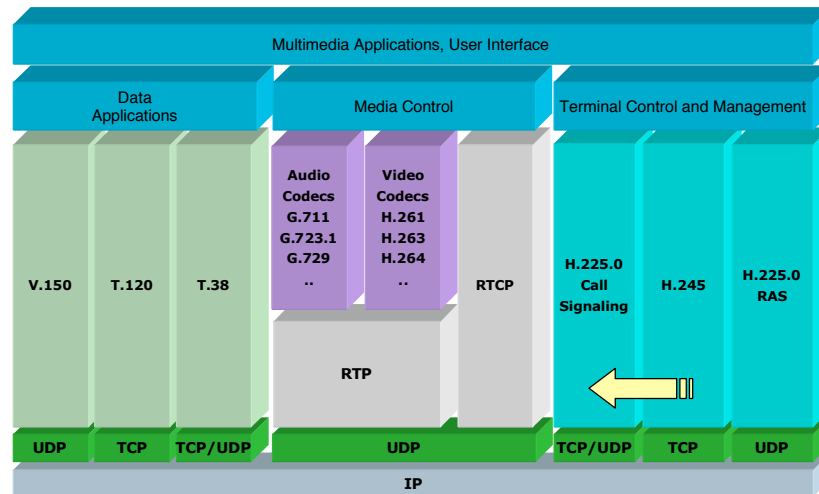
H.323

ITU protocol suite for audio/video conferencing over “networks that do not provide guaranteed quality of service”.



Source: microsoft.com

H.323 stack



Source: packetizer.com

H.323 Protocol Components

- A/V codecs (H.261, H.263, G711, G.723).
- H.225.0 transport
 - Uses RTP/RTCP for audio/video packets
 - Includes RAS (Registration, Admission and Status) signalling channel for communication with the *gatekeeper*.
- Q.931
 - “ISDN user-network interface layer 3 specification for basic call control”
- H.245 call control
 - Negotiation of codec, bit rate, frame rate, etc.
- T.120 data communications.



H.323 System Components

- Terminal
 - videophone, MS netmeeting software, etc
- Gateway
- Gatekeeper
- MCU (multipoint control unit).



Gateways

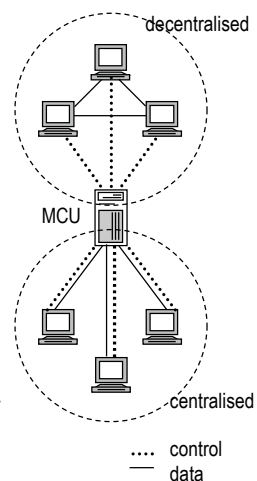
- Optional element in an H.323 conference.
 - Not usually needed for pure H.323 to H.323 calls.
- Principle role is *translation function* between H.323 conferencing endpoints and other terminal types. Eg:
 - Establishing links with analog PSTN terminals.
 - Establishing links with remote H.320-compliant terminals over ISDN-based switched-circuit networks.
 - Establishing links with remote H.324-compliant terminals over PSTN networks.

Gatekeepers

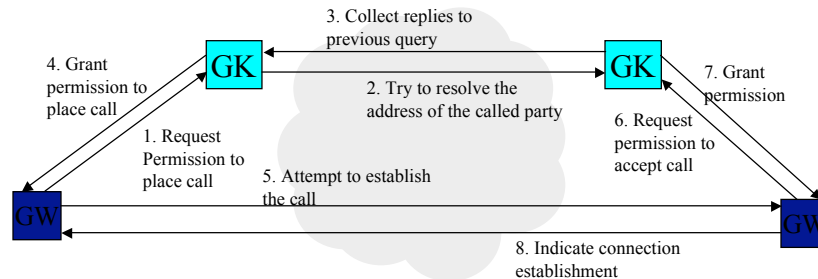
- *Optional* component used for admission control and address resolution. Acts as the central point for all calls within its zone and provides call control services to registered endpoints.
 - May allow calls to be placed directly between endpoints
 - May route the call signaling through itself to perform functions such as follow-me/find-me, forward on busy, etc. Service providers can also use this to bill for calls placed through their network.
- Can be used to limit the total conferencing bandwidth to some fraction of the total available.

Multipoint Control Units (MCU)

- Responsible for managing multipoint conferences.
 - Three or more endpoints.
- MCU contains:
 - *Multipoint Controller (MC)* that manages the call signaling, and handles H.245 negotiations between all terminals to determine common capabilities for A/V processing.
 - *Multipoint Processors (MPs)* to handle audio and video mixing, switching, or other media processing. MP is an optional component of the MCU.



High level call flow



H.323 Call Progress

H.323 has several different ways a call can progress

- ☐ Direct mode vs Routed Mode
- ☐ Regular call vs Fast start call (in H.323v2)

TCS = Terminal Capability Set

MSD = Master/Slave Determination

DRQ = Disconnect Request

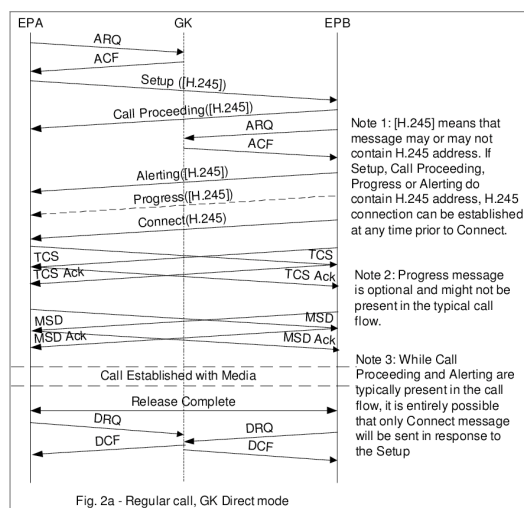


Fig. 2a - Regular call, GK Direct mode

[Source: L. H.323 forum]

H.323 Usage

- Microsoft NetMeeting
- Lots of commercial videoconferencing equipment
 - Eg: Polycom
- Some IP phones (including some of Cisco's)



IETF Multimedia Protocol Suite

- **RTP/RTCP** for data flow, A/V sync, and reception feedback.
- **SDP** (Session Description Protocol) for describing a multimedia session
 - Also used for negotiation of session parameters.
- **SAP** (Session Announcement Protocol) for broadcast-style announcement of multicast sessions.
- **SIP** (Session Initiation Protocol) for setting up and reconfiguring multimedia calls.
 - Telephony-like signalling.
 - Instant Messaging.
- **RTSP** (Real Time Streaming Protocol) for remote control of VCR-style functionality.



Session Description Protocol (SDP)

- SDP is a standard way to describe multimedia sessions.
- These descriptions can then be used in different contexts:
 - Session Announcements using SAP
 - Session Invitations using SIP
 - RTSP stream descriptions
 - H.323 announcements
 - PINT (PSTN/Internet IN feature mapping)
 - Advanced Television Enhancement Forum (!)
- SDP was really only designed for SAP - the other uses stretch it a little beyond its design space.
 - Sometimes this shows.



SDP

- SDP is a text-based description format.
 - It is extensible through attributes (which don't have to be registered) and by several other namespaces that are registered with IANA.
- It was not intended for content negotiation.
 - SIP can use it for this purpose, but it's not as elegant as if SDP had been designed for this.

SDP

It was intended for announcing the existence of sessions. It conveys:

- ☐ Information to allow you to choose whether to join the session.
- ☐ Session timing information
- ☐ Information to inform you of the resources required to participate.
- ☐ Sufficient information to allow you to join the session
 - protocols and codec formats
 - multicast addresses and ports
 - encryption keys
- ☐ Information that RTP needs passed out-of-band.

SDP: Example

```
v=0
o=mjh 8357876234 8357876234 IN IP4 18.26.0.4
s=MCI Internet Multimedia Session
i=A course on Internet Multimedia by Mark Handley...
e=Mark Handley <mjh@isi.edu>
p=+1 617 253 6011
b=CT 512
t=8432343523 8432554353
a=recvonly
m=audio 21453 RTP/AVP 96
c=IN IP4 224.5.6.7/127
a=rtpmap 96 L16/16000
m=video 12348 RTP/AVP 31
c=IN IP4 224.5.6.8/127
a=quality 10
```

Identifies session

contact information

bandwidth information

timing information

session is not interactive

audio stream description

binding of RTP dynamic payload type 96 to L16/16000

SIP: Session Initiation Protocol

Original spec:

RFC 2543

Updated specs:

RFC 3261 (main spec)

RFC 3262 (provisional response reliability)

RFC 3263 (locating SIP servers)

RFC 3264 (offer/answer use of SDP)

RFC 3265 (specific event notification)

SIP: Early History

Early session initiation protocols:

- ivsd (Turletti, INRIA)
- mmcc (Schooler, ISI)

These led eventually to *SIPv1* (Handley and Schooler) which was intended to initiate loosely coupled sessions.

- SIP stopped when the session started
- UDP based

Henning Schulzrinne designed *SCIP* using RTSP as a basis around the same time.

- TCP based
- Continued during the call

Eventually we merged *SIPv1* and *SCIP* into *SIPv2*

- best features of both, UDP and TCP

Note: *H.323* was also being drafted at the same time

SIP: Aims

- Both SIPv1 and SCIP were trying to allow user mobility.
 - Few people used *mmcc* because people move around and change hosts.
- In the merged draft, this became a key issue.
- Goal was to be able to support invitations to public multiparty multicast sessions or to private sessions.
 - SIP may or may not continue to be involved after joining the session.

SIP: User Location

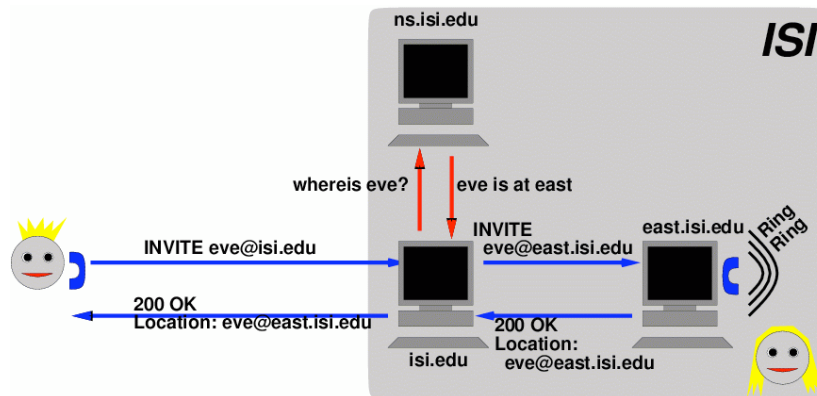
There are two basic ways to do user location:

- **Have a distributed directory.**
 - Lookup user's location in directory.
 - Address a call to that location.
- **Lookup during call routing**
 - Lookup a well-known address for the user.
 - Route the call there. Let them do the lookup of user location.
 - Either relay or redirect the call.
 - Multiple lookups can occur if required.

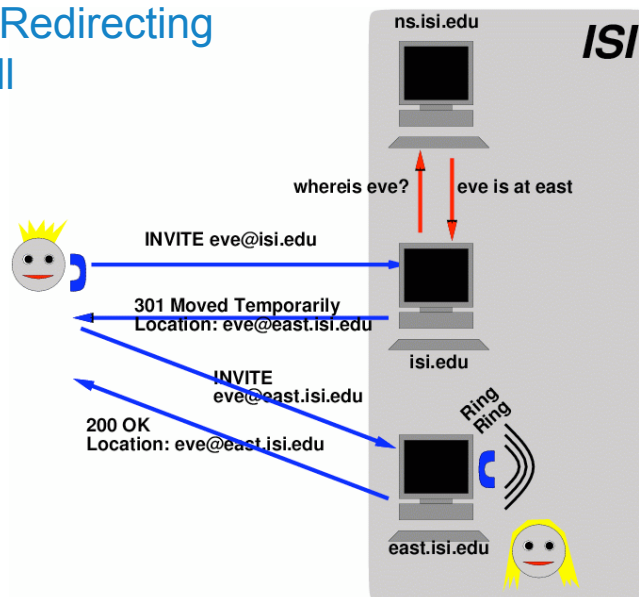
In the Internet, *heterogeneity* is key. Uniform distributed directories such as X500 have failed to be deployed.

- Lookup during call routing allows heterogeneity of user-location mechanisms.
- Improved security.

SIP: Relaying a Call



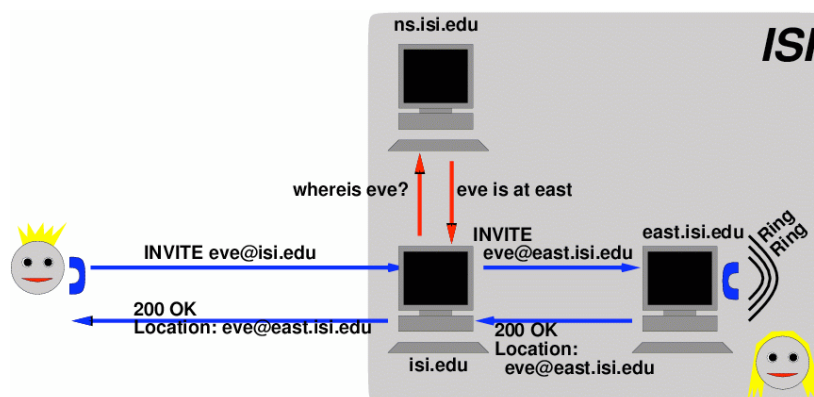
SIP: Redirecting a Call



SIP Proxies

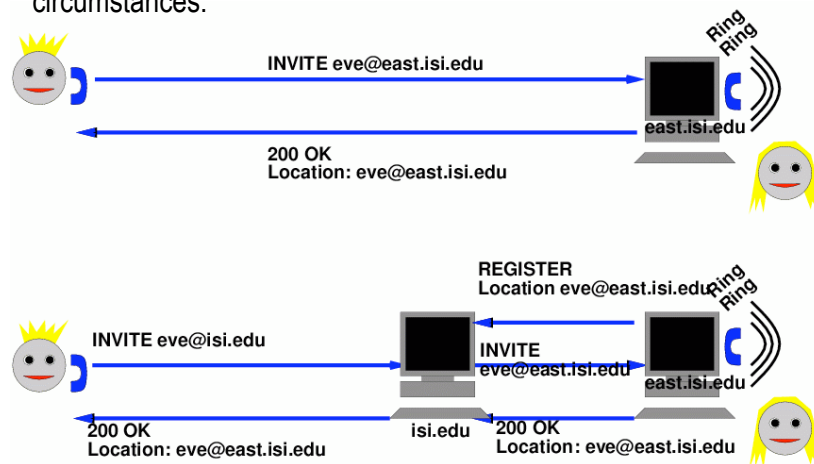
- SIP proxies can use any reasonable search algorithm
 - Send requests in parallel
 - Send requests sequentially
- Normally only a proxy close to the callee can decide on an appropriate search strategy.
- SIP specifies only the rules that proxies must use to combine responses when multiple requests are made in parallel.
- A standard way to specify proxy call processing rules is desirable, but SIP itself doesn't care how the processing is performed.

SIP: User Location Servers

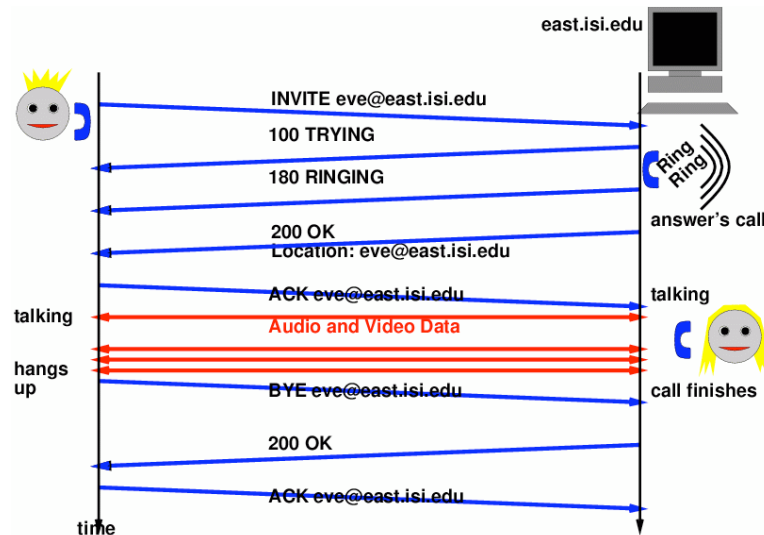


SIP: User Location Servers

- SIP doesn't need a separate user location server in many circumstances:



SIP Normal protocol operation



SIP Syntax

- SIP is a text based protocol, similar in syntax to HTTP and RTSP.
- Messages can be conveyed over UDP or TCP.
 - SIP provides its own reliability over UDP.
 - UDP is preferred - it gives more control over message timing, and requires less state in proxies.
 - TCP is allowed for legacy firewall traversal but in time we hope firewalls themselves will support SIP.
- Typically SIP carries an SDP session description as a payload to describe the session being initiated.

SIP Request (sent to north.east.isi.edu)

```
INVITE sip:mjh@north.east.isi.edu SIP/2.0
Via: SIP/2.0/UDP east.isi.edu           <-- second relay
Via: SIP/2.0/UDP isi.edu                 <-- first relay
Via: SIP/2.0/UDP chopin.cs.caltech.edu   <-- originating host
To: sip:mjh@isi.edu                     <-- original destination
From: eve@cs.caltech.edu                 <-- sender
CSeq: 1                                 <-- command seq. no.
Content-Type: application/sdp
Content-Length: 214

_____ start of payload _____
v=0
o=eve 987329833 983264598 IN IP4 128.32.83.24
s=Quick Call
...
```

SIP Response (sent to east.isi.edu)

SIP/2.0 200 OK

Via: SIP/2.0/UDP isi.edu

Via field for *east* removed already

Via: SIP/2.0/UDP chopin.cs.caltech.edu

To: sip:mjh@isi.edu

Refers to request "to", "from"
not message to and from.

From: sip:eve@cs.caltech.edu

Location: sip:mjh@north.east.isi.edu;tag=76fa98c80aba81

CSeq: 1

Content-Type: application/sdp

Content-Length: 214

v=0

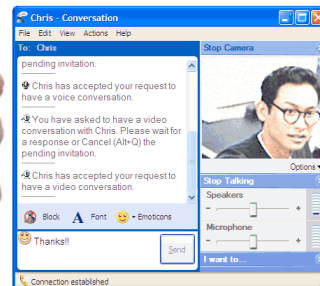
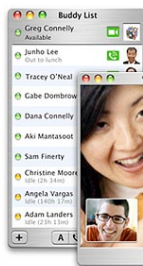
o=eve 987329833 983264598 IN IP4 128.32.83.24

s=Quick Call

...

SIP Usage

- Almost all IP phones
- Microsoft Windows Messenger
- Apple iChatAV
- AT&T, MCI VoIP service
- Sprint PCS cellphone (walkie-talkie service)
- 3G cellular: IP Multimedia Call Control
- many more...





RTSP: Real-Time Stream (Control) Protocol

- RTSP provides a way to set up and control multimedia streams from a media server.
- Essentially RTSP is the remote control for a network-VCR.
- RTSP is in the same "protocol family" as SIP and HTTP:
 - ☐ text based
 - ☐ MIME-format messages
 - ☐ HTTP-like syntax
 - ☐ shared response codes.



RTSP functionality

- Setup a connection and exchange stream transport information
- Describe the session
- Play the session
 - ☐ from specified start times
 - ☐ forwards, backwards
 - ☐ at different speeds and data rates
- Record a session
- Pause playback or recording

RTSP example

■ Client to HTTP Server:

```
GET /twister.sdp HTTP/1.1
Host: www.example.com
Accept: application/sdp
```

■ HTTP Server to Client:

```
HTTP/1.0 200 OK
Content-Type: application/sdp
v=0
o=- 2890844526 2890842807 IN IP4 192.16.24.202
s=RTSP Session
m=audio 0 RTP/AVP 0
a=control:rtsp://audio.example.com/twister/audio.en
m=video 0 RTP/AVP 31
a=control:rtsp://video.example.com/twister/video
```

RTSP example

■ Client to Audio Server:

```
SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast;client_port=3056-3057
```

■ Audio Server to Client:

```
RTSP/1.0 200 OK
CSeq: 1
Session: 12345678
Transport: RTP/AVP/UDP;unicast;client_port=3056-3057;
          server_port=5000-5001
```

RTSP example

■ Client to Video Server:

```
SETUP rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast;client_port=3058-3059
```

■ Video Server to Client:

```
RTSP/1.0 200 OK
CSeq: 1
Session: 23456789
Transport: RTP/AVP/UDP;unicast;client_port=3058-3059;
          server_port=5002-5003
```

RTSP example

■ Client to Video Server:

```
PLAY rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 2
Session: 23456789
Range: smpte=0:10:00-
```

■ Video Server to Client:

```
RTSP/1.0 200 OK
CSeq: 2
Session: 23456789
Range: smpte=0:10:00-0:20:00
RTP-Info:
  url=rtsp://video.example.com/twister/video;
  seq=12312232;rtptime=78712811
```

RTSP example

- *Client to Audio Server:*

```
TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/  
CSeq: 3  
Session: 12345678
```

- *Audio Server to Client:*

```
RTSP/1.0 200 OK  
CSeq: 3
```

- *Client to Video Server:*

```
TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0  
CSeq: 3  
Session: 23456789
```

- *Video Server to Client:*

```
RTSP/1.0 200 OK  
CSeq: 3
```

RTSP Usage

- RealPlayer (and Helix open-source version)
- Microsoft Windows Media 9
 - ☐ Also use Microsoft's proprietary *mms*.
- Apple Quicktime Player
- 3G cellular video streaming.



References

ITU-T Recommendation H.323 “*Packet-based multimedia communications systems*” <http://www.itu.int>

RFC2327, “*SDP: Session Description Protocol*” [M].
Handley, V. Jacobson

RFC3261, “*SIP: Session Initiation Protocol*”, [J]. Rosenberg,
H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson,
R. Sparks, M. Handley, E. Schooler

RFC2326, “*Real Time Streaming Protocol (RTSP)*” [H].
Schulzrinne, A. Rao, R. Lanphier