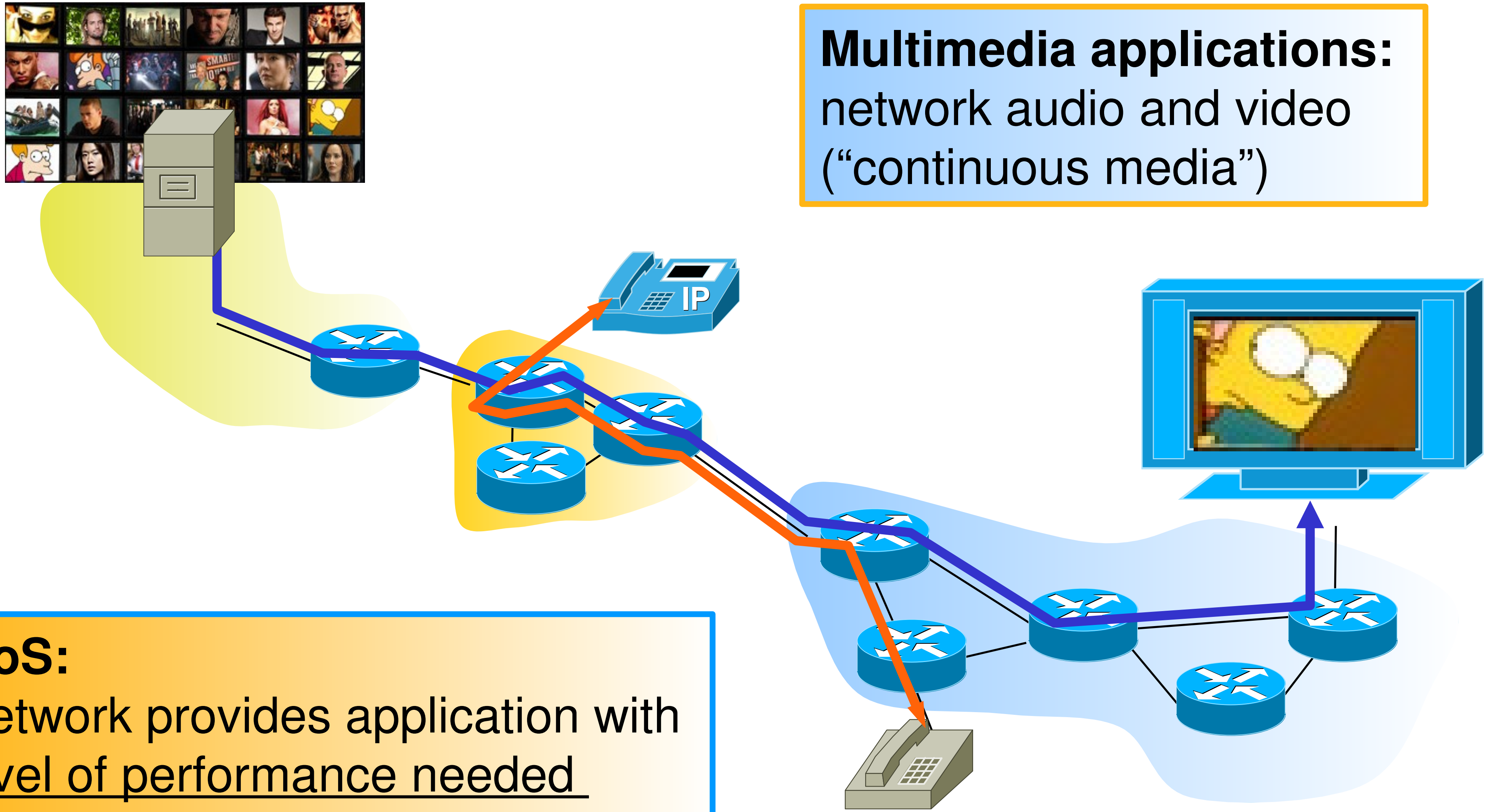
IP Multimedia

Arquitetura de Redes Avançadas

universidade de aveiro
deti.ua.pt

Multimedia, Quality of Service: What is it?

Multimedia applications:
network audio and video
("continuous media")



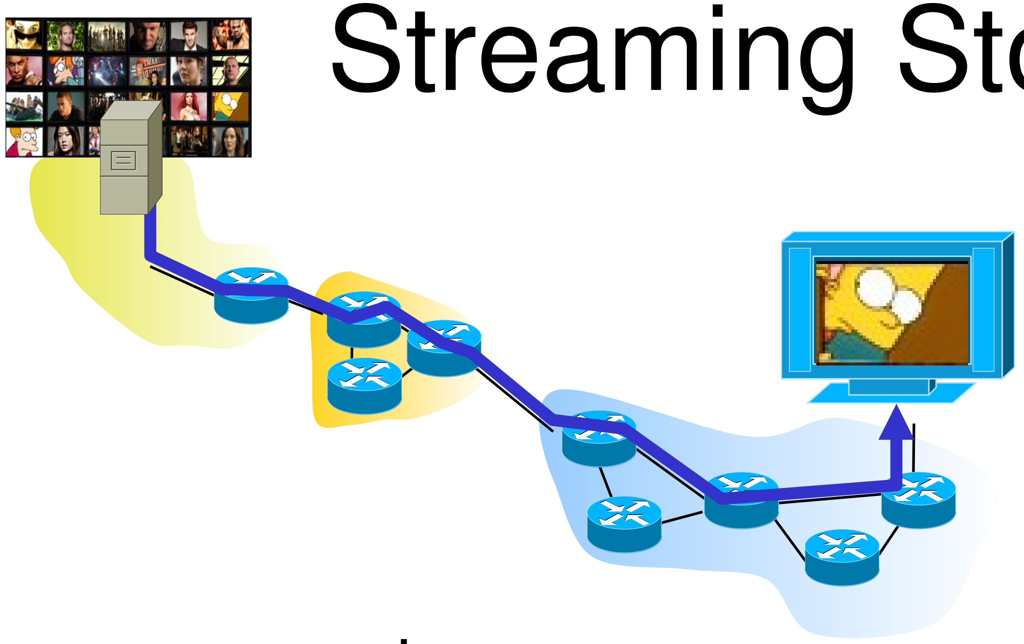
QoS:
Network provides application with
level of performance needed
for application to function.

Multimedia Networking Applications

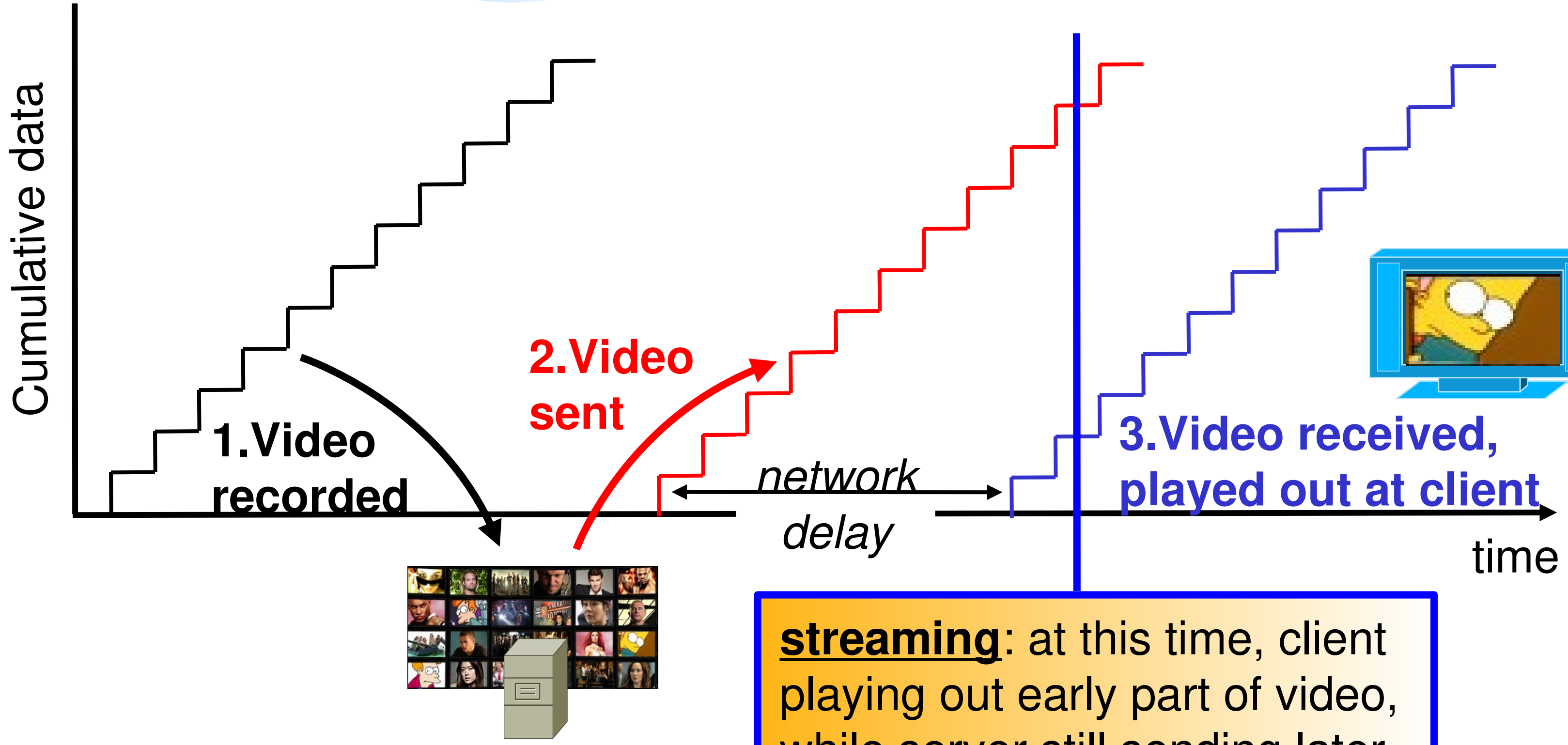
- Fundamental characteristics:
  - Typically delay sensitive
    - end-to-end delay
    - delay jitter
  - But loss tolerant: infrequent losses cause minor glitches
  - Antithesis of data, which are loss intolerant but delay tolerant.
- Classes of multimedia applications:
  - Streaming stored audio and video
  - Streaming live audio and video
  - Real-time interactive audio and video

Jitter is the variability of packet delays within the same packet stream

Streaming Stored Multimedia



- Streaming:
  - Media stored at source is transmitted to client.
  - Client visualization begins before all data has arrived.
    - Timing constraint for transmitted data: in time for playout.



streaming: at this time, client playing out early part of video, while server still sending later part of video

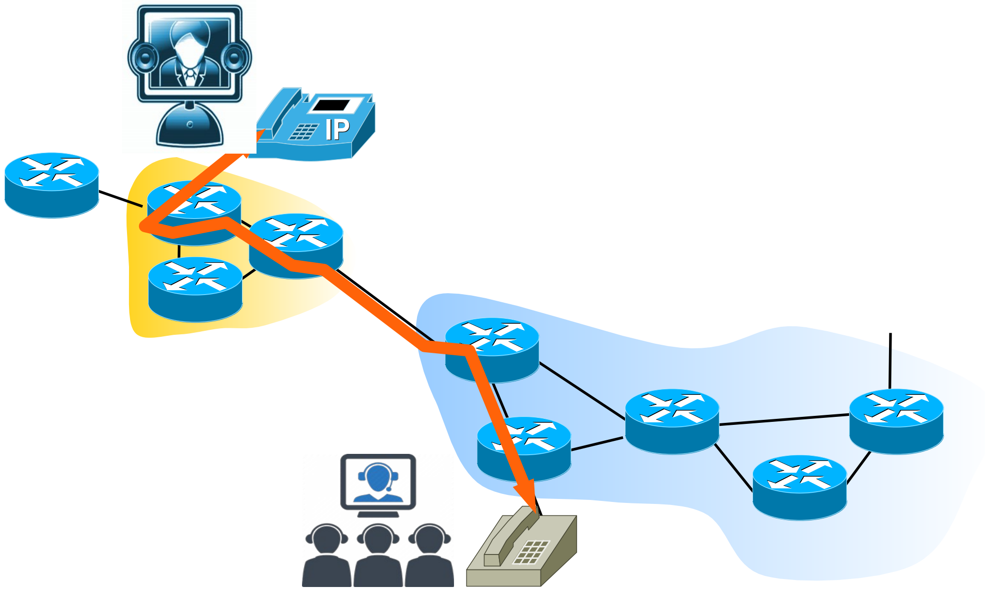
Streaming Stored Multimedia: Interactivity

- VCR-like functionality: client can pause, rewind, fast-foward, push slider bar.
  - 10 sec initial delay OK.
  - 1-2 sec until command effect OK.
  - Timing constraint for still-to-be transmitted data: in time for playout.

Streaming Live Multimedia

- Examples:
  - Internet TV/radio show.
  - Live sporting event.
- Streaming
  - Playback buffer.
  - Playback can lag tens of seconds after transmission.
  - Still have timing constraint.
- Interactivity
  - Fast forward impossible.
  - Rewind, pause possible!

Interactive Real-Time Multimedia



- Applications:
  - IP telephony, video conference, online-game multimedia actions, distributed interactive worlds.
- End-end delay requirements:
  - Audio: < 150 msec good, < 400 msec OK
    - Includes application-level (packetization) and network delays.
    - Higher delays noticeable, impair interactivity.
- Requires session initialization
  - Advertise its IP address, port number, encoding algorithms, required contents, available contents

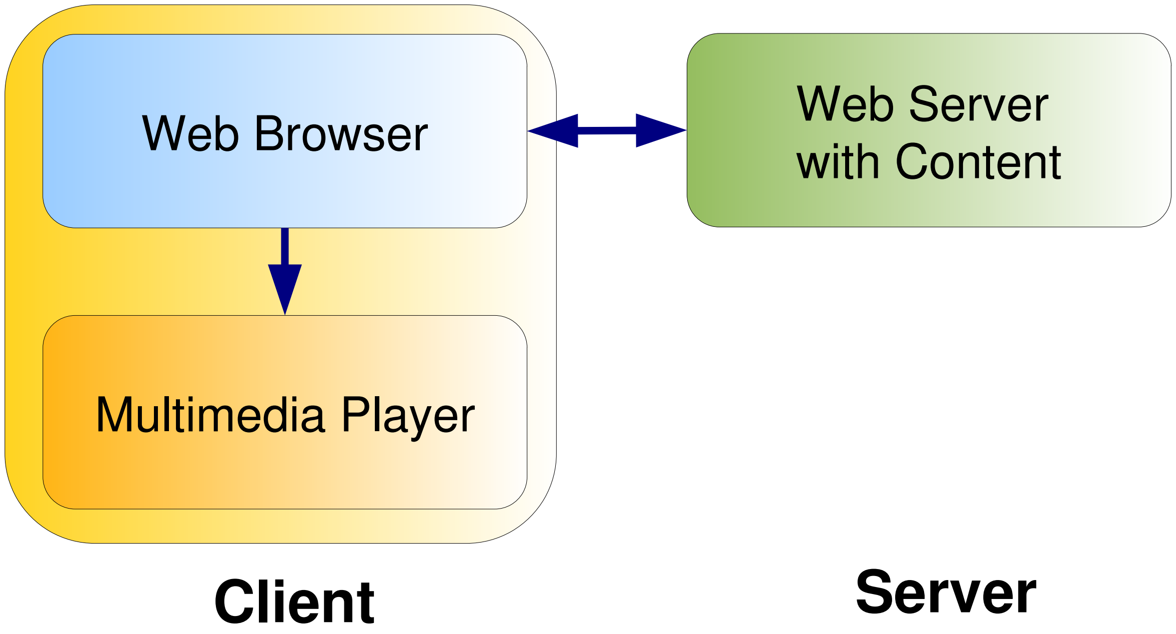
Internet Multimedia Support

- Integrated services philosophy.
  - Requires dedicated links/channels with QoS requirements.
- Differentiated services philosophy.
  - Fewer changes to Internet infrastructure.
- Best effort.
  - No major changes.
  - More bandwidth when needed.
  - Application-level control and distribution.

Streaming Stored Multimedia

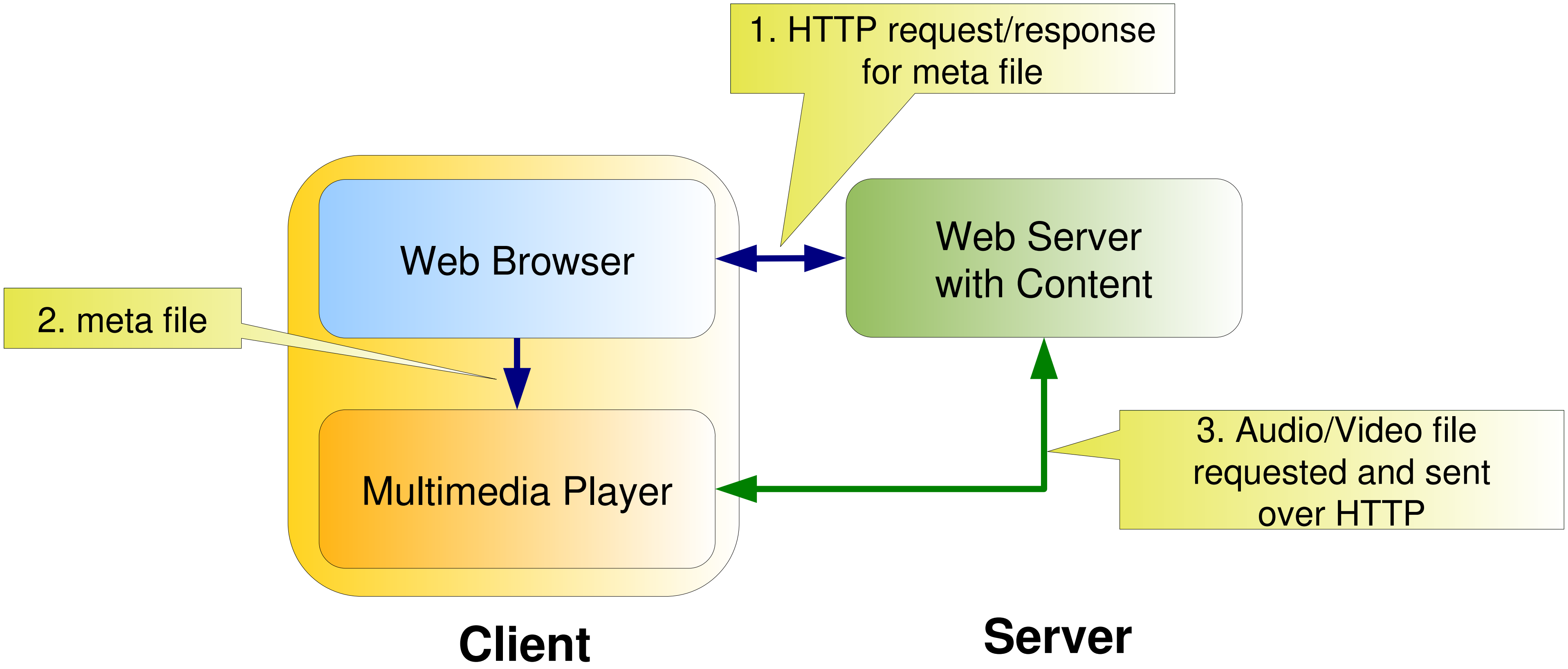
- Application-level streaming techniques for making the best out of best effort service:
  - Client side buffering.
  - Use of UDP versus TCP.
  - Multiple encodings of multimedia.
- Multimedia Player
  - Jitter removal,
  - Decompression,
  - Error concealment,
  - Graphical user interface with controls for interactivity.
- Network
  - Close to client content (multi-content) buffering for faster interactivity
  - Only viable in network operator proprietary services.

Internet Multimedia: Simplest Approach



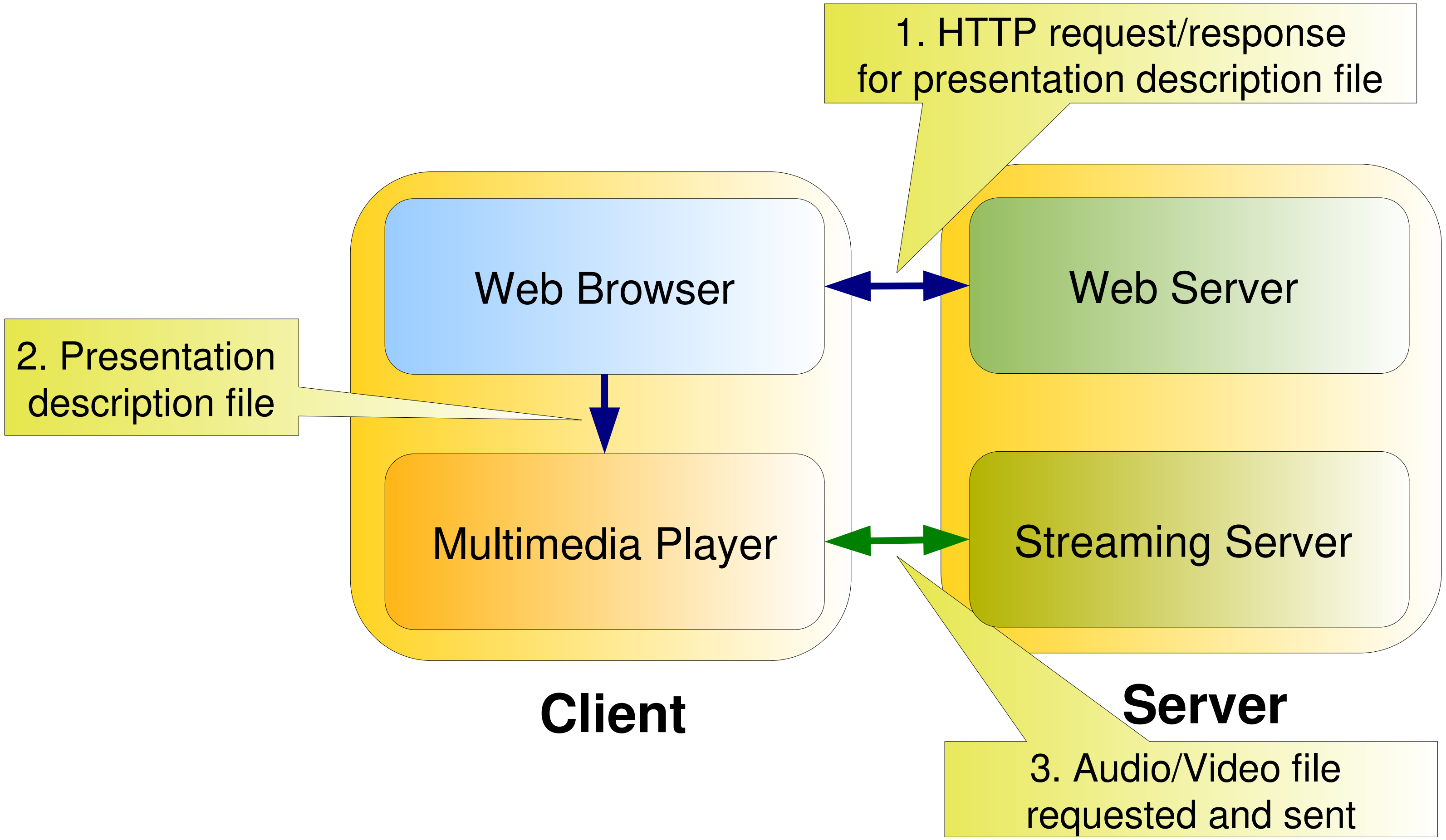
- Audio or video stored in file.
- Files transferred as HTTP object.
  - Received in entirety at client.
  - Then passed to player.
- Audio, video not streamed!
- No “pipelining”, long delays until playout!

Internet Multimedia: Streaming Approach



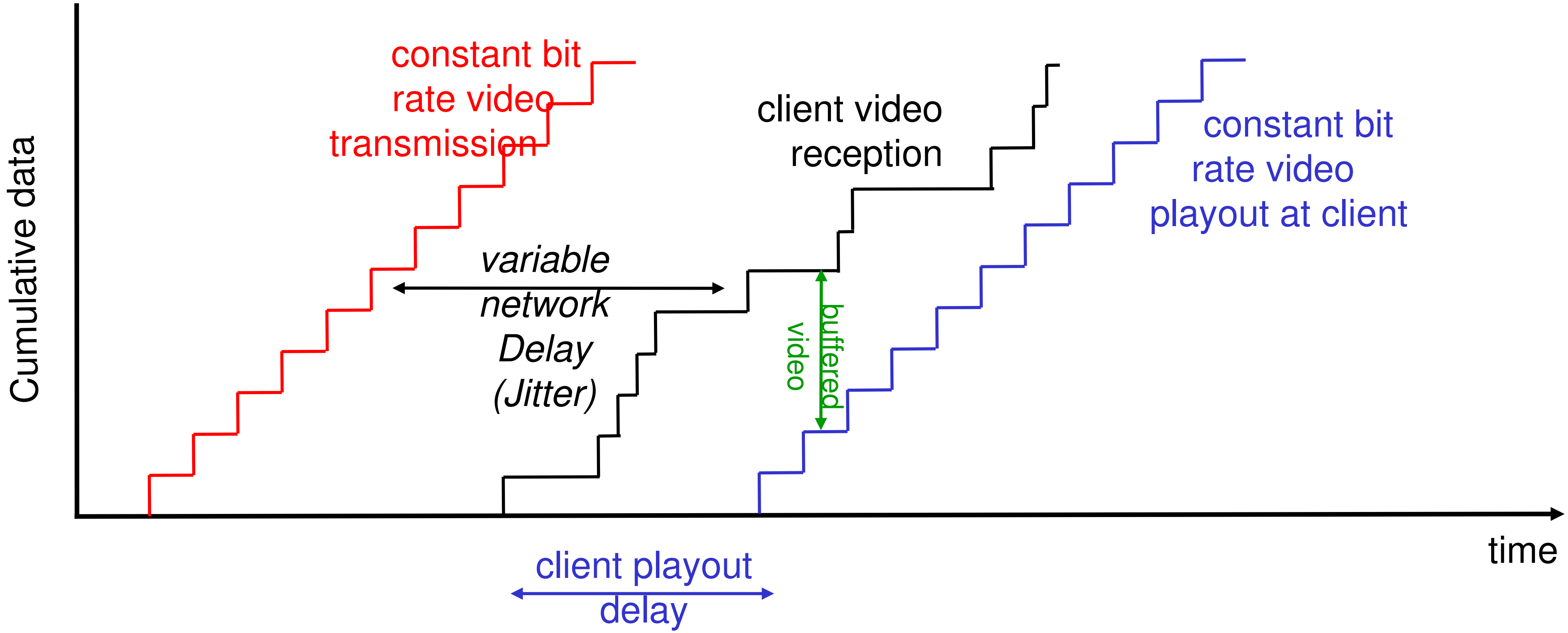
- Browser GETs metafile.
- Browser launches player, passing metafile.
- Player contacts server.
- Server streams audio/video to player.

Streaming from a streaming server



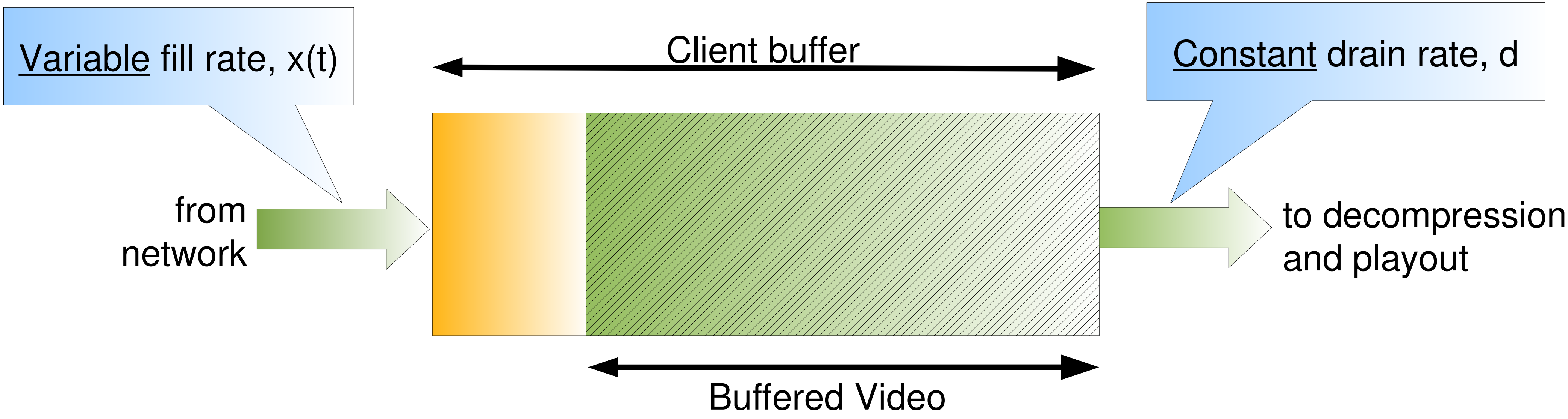
- This architecture allows for non-HTTP protocol between server and media player.
- Can use UDP or TCP transport.

Streaming Multimedia: Client Buffering



- Client-side buffering, playout delay compensate for network-added delay, delay jitter.

Streaming Multimedia: Client Buffering



- Client-side buffering, playout delay compensate for network-added delay, delay jitter.

UDP Streaming vs. TCP Streaming

- UDP
  - Server sends at rate appropriate for client .
    - Often send rate = encoding rate = constant rate.
    - Then, fill rate = constant rate - packet loss.
  - Short playout delay (2-5 seconds) to compensate for network delay jitter.
  - Error recover: time permitting.
- TCP
  - Send at maximum possible rate under TCP.
  - Fill rate fluctuates due to TCP congestion control.
  - Larger playout delay: smooth TCP delivery rate.
  - HTTP/TCP passes more easily through firewalls.

User Control of Streaming Media: RTSP

- RTSP (Real Time Streaming Protocol): RFC 2326
  - Client-server application layer protocol.
  - For user to control display: rewind, fast forward, pause, resume, repositioning, etc…
- Does not define how audio/video is encapsulated for streaming over network.
- Does not restrict how streamed media is transported.
  - Can be transported over UDP or TCP.
- Does not specify how the media player buffers audio/video.
- RTSP messages are also sent out-of-band:
  - RTSP control messages use different port numbers than the media stream: out-of-band
    - Port 554
  - The media stream is considered “in-band”

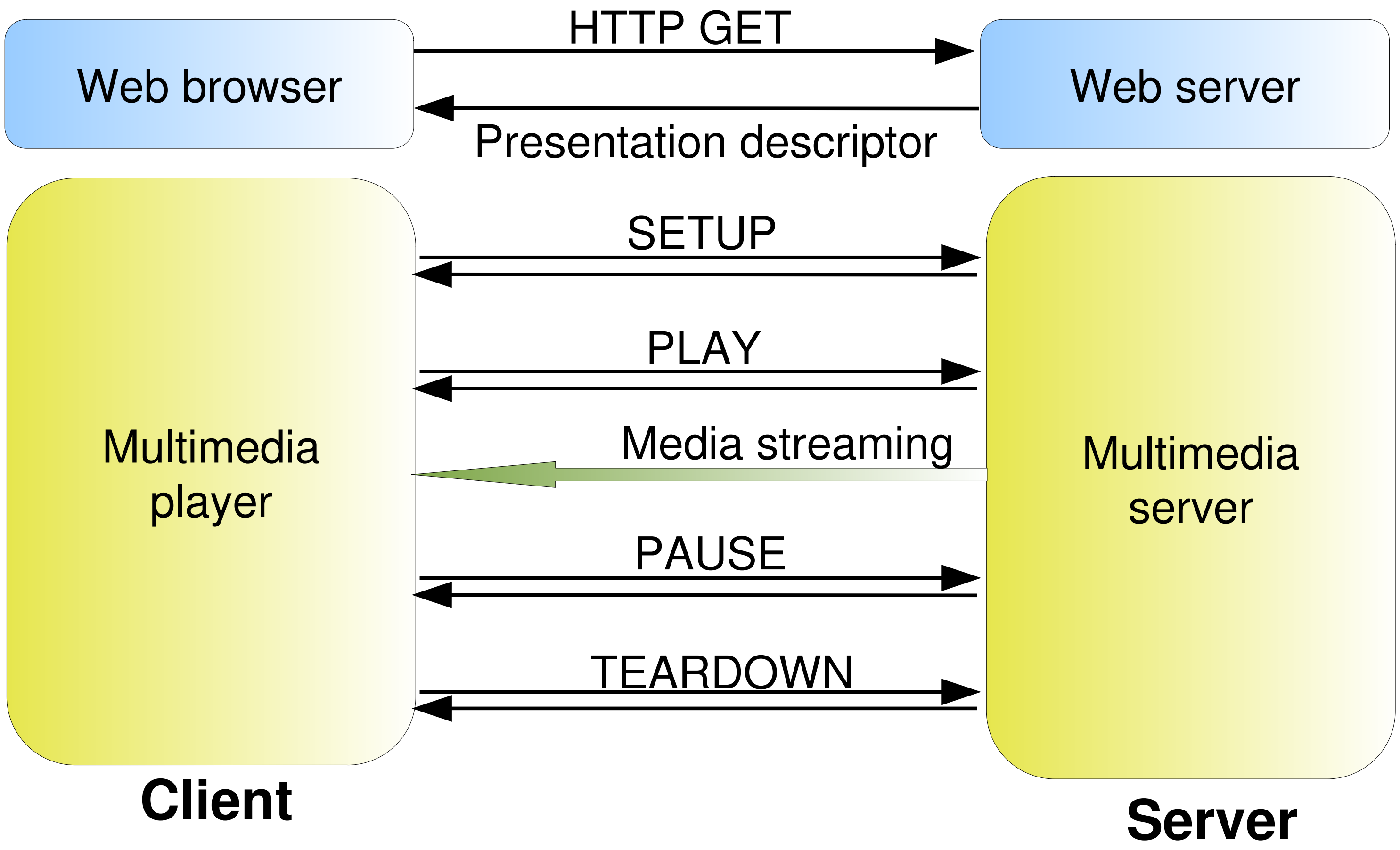
RTSP: out of band control

- FTP uses an “out-of-band” control channel:
  - A file is transferred over one TCP connection
  - Control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection
  - The “out-of-band” and “in-band” channels use different port numbers
- RTSP messages are also sent out-of-band:
  - RTSP control messages use different port numbers than the media stream: out-of-band
  - Port 554
  - The media stream is considered “in-band”

Metafile Example

```
<title>Twister</title>
<session>
    <group language=en lipsync>
        <switch>
            <track type=audio
                e="PCMU/8000/1"
                src = "rtsp://audio.example.com/twister/audio.en/lofi">
            <track type=audio
                e="DVI4/16000/2" pt="90 DVI4/8000/1"
                src="rtsp://audio.example.com/twister/audio.en/hifi">
        </switch>
        <track type="video/jpeg"
                src="rtsp://video.example.com/twister/video">
    </group>
</session>
```

RTSP Operation



RTSP Exchange Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231

S: 200 3 OK

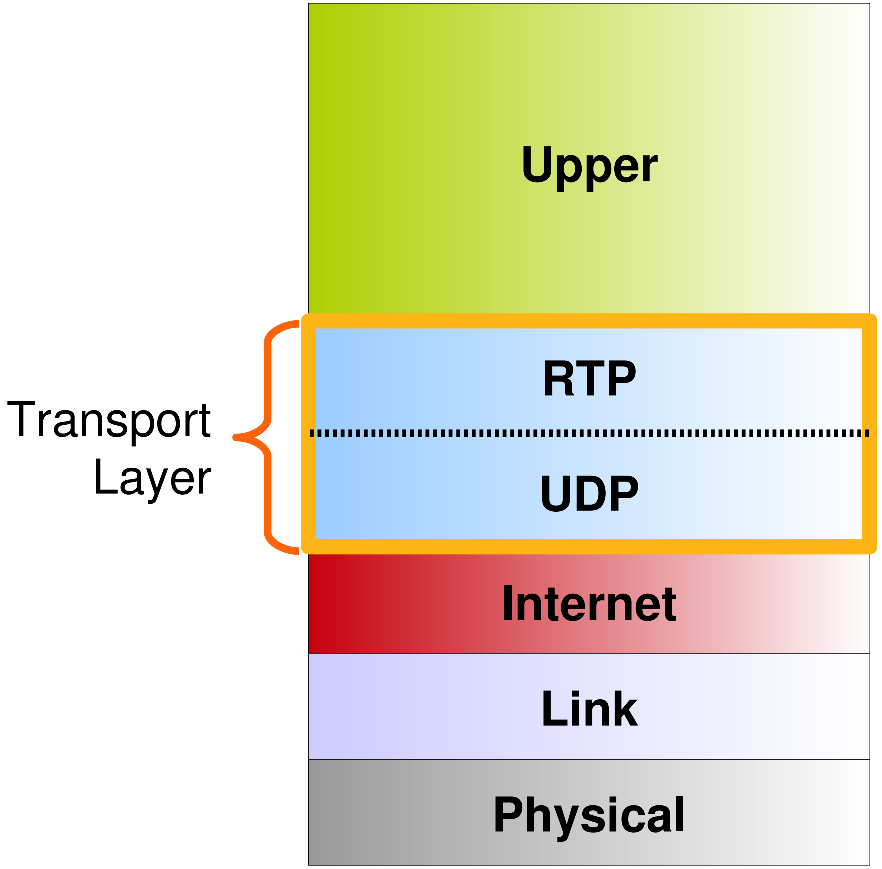
Real-Time Protocol (RTP)

- RTP specifies a packet structure for packets carrying audio and video data
- RFC 1889.
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - timestamping
- RTP runs in the end systems.
- RTP packets are encapsulated in UDP segments
- Interoperability: if two Internet phone applications run RTP, then they may be able to work together

RTP runs on top of UDP

- RTP libraries provide a transport-layer interface

  that extend UDP:
  - Port numbers, IP addresses
  - Payload type identification
  - Packet sequence numbering
  - Time-stamping



RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment
- RTP header indicates type of audio encoding in each packet
  - Sender can change encoding during a conference
- RTP header also contains sequence numbers and timestamps

RTP and QoS

- RTP <u>does not</u> provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

RTP Header

payload type	sequence number	timestamp	synchronization source identifier	misc. fields

- Payload Type (7 bits)
  - Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs the receiver through this payload type field
    - Payload type 0: PCM mu-law, 64 kbps
    - Payload type 3, GSM, 13 kbps
    - Payload type 7, LPC, 2.4 kbps
    - Payload type 26, Motion JPEG
    - Payload type 31. H.261
    - Payload type 33, MPEG2 video
- Sequence Number (16 bits)
  - Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence
- Timestamp field (32 bytes long)
  - Reflects the sampling instant of the first byte in the RTP data packet
- SSRC field (32 bits long)
  - Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC

Real-Time Control Protocol (RTCP)

- Works in conjunction with RTP
- Each participant in RTP session periodically transmits RTCP control packets to all other participants
- Each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application
- Statistics include number of packets sent, number of packets lost, interarrival jitter, etc...
- Feedback can be used to control performance
- Sender may modify its transmissions based on feedback

RTCP Packets

- Receiver report packets:
  - Fraction of packets lost, last sequence number, average interarrival jitter
- Sender report packets:
  - SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.
- Source description packets:
  - Sender e-mail address, sender's name, SSRC of associated RTP stream.
  - Provide mapping between the SSRC and the user/host name.

Synchronization of Streams

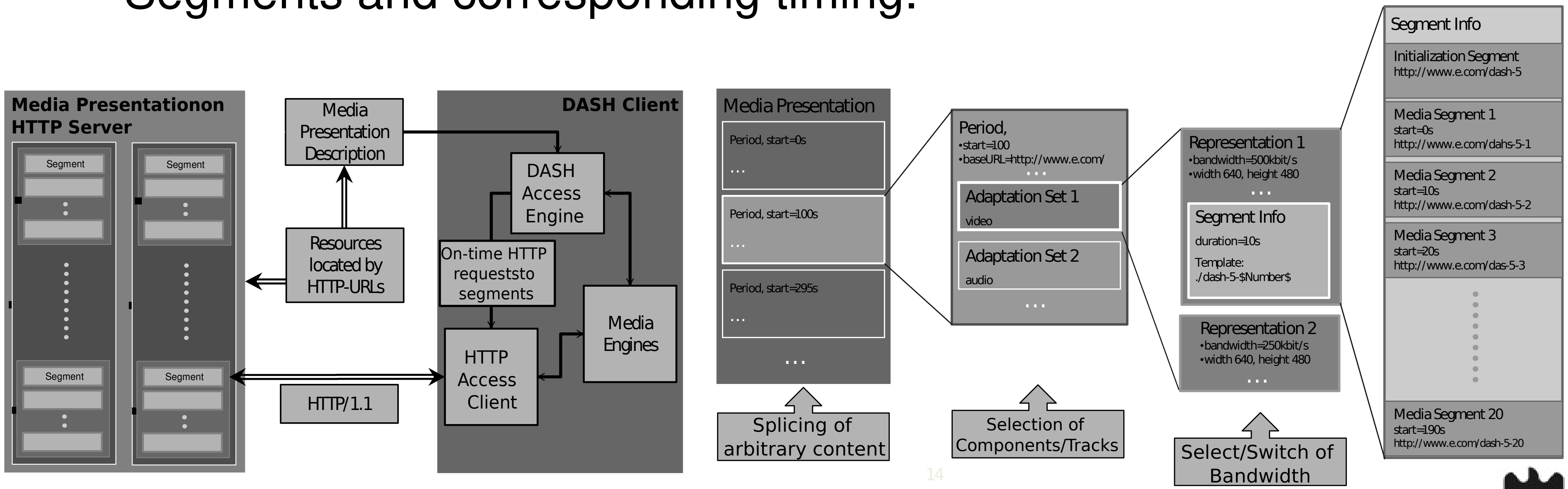
- RTCP can synchronize different media streams within a RTP session
- Consider videoconferencing application for which each sender generates one RTP stream for video and one for audio
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - Not tied to the wall-clock time
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - Timestamp of the RTP packet
  - Wall-clock time for when packet was created
- Receivers can use this association to synchronize the playout of audio and video

HTTP/TCP Streaming

- Multiple versions with distinct/complementary characteristics are generated for the same content
  - With different bitrates, resolutions, frame rates.
- Each version is divided into time segments.
  - e.g., two seconds.
- Each segment is provided on a web server and can be retrieved through standard HTTP GET requests.
- Examples of protocols:
  - MPEG's Dynamic Adaptive Streaming over HTTP (DASH).
    - Standard ISO/IEC 23009-1. YouTube's default.
  - Adobe HTTP Dynamic Streaming (HDS).
  - Apple HTTP Live Streaming (HLS).
  - Microsoft Smooth Streaming (MSS).

Dynamic Adaptive Streaming over HTTP (DASH)

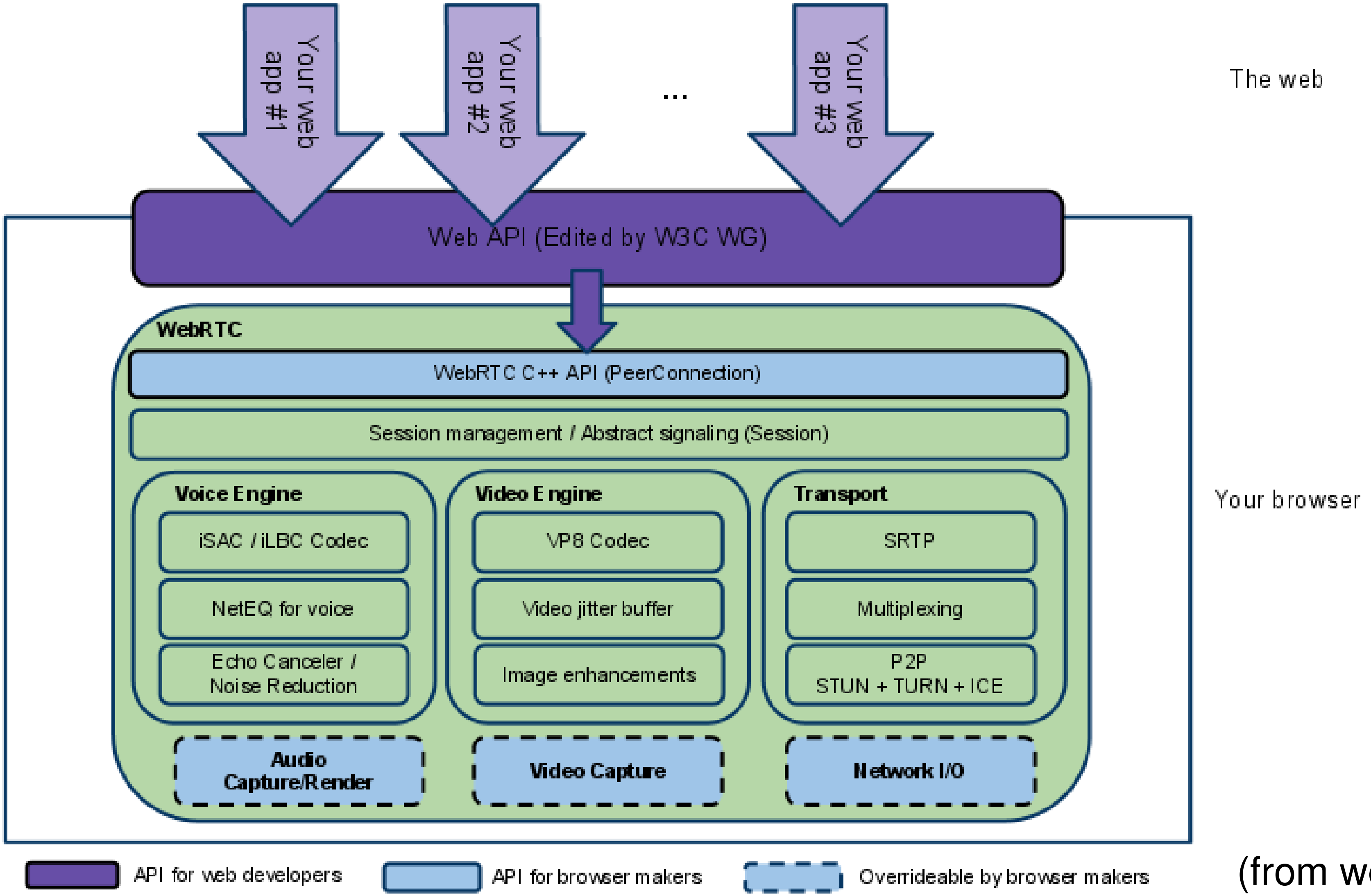
- Developed to be an Open Standard Delivery Format.
  - MPEG DASH ISO/IEC 23009-1.
- Video streaming solution where pieces of video streams/files are requested with HTTP and spliced together by the client.
  - Client entirely controls delivery.
- Media Presentation Description (MPD) describes accessible Segments and corresponding timing.



WebRTC

- Peer-to-peer connections.
  - An instance allows an application to establish peer-to-peer communications with another instance in another browser, or to another endpoint implementing the required protocols.
- RTP Media.
  - Allow a web application to send and receive media stream over a peer-to-peer connection.
- Peer-to-peer Data
  - Allows a web application to send and receive generic application data over a peer-to-peer connection.
- Peer-to-peer DTMF.

WebRTC Architecture



(from webrtc.org)

VoIP
Voice (and Video) over IP

Arquitetura de Redes Avançadas

universidade de aveiro
deti.ua.pt

Voice over IP

- Network loss: IP datagram lost due to network congestion (router buffer overflow).
- Delay loss: IP datagram arrives too late for playout at receiver.
  - Delays: processing, queueing in network; end-system (sender, receiver) delays.
  - Typical maximum tolerable delay: 400 ms.
- Loss tolerance: depending on voice encoding, packet loss rates between 1% and 10% can be tolerated.
- Speaker's audio: alternating talk/speech with silent periods.
  - 64 kbps during talk/speech.
  - Packets generated only during talk/speech.
    - 20 msec chunks at 8 Kbytes/sec: 160 bytes data.
- Requires session establishment.
- VoIP protocols/frameworks:
  - Session Initiation Protocol (SIP)
    - Session Description Protocol (SDP)
  - H.323
- VoIP and PSTN interoperability in large/ISP scalable scenarios require complex control frameworks:
  - Media Gateway Controller Protocol (MGCP);
  - H.248/Megaco.

SIP vs H.323

- SIP comes from IETF: Borrows much of its concepts from HTTP.
- H.323 is another signaling protocol for real-time, interactive.
  - Comes from the ITU (telephony).
- SIP has a Web flavor, whereas H.323 has a telephony flavor.
- SIP is a single component. Works with RTP, but it can be combined with other protocols and services.
- H.323 is a complete, vertically integrated suite of protocols for multimedia conferencing: signaling, registration, admission control, transport and codecs.

Session Initiation Protocol (SIP)

- Defined by RFC 3261.
- Designed for creating, modifying and terminating sessions between two or more participants.
  - Not limited to VoIP calls.
- Is a text-based protocol similar to HTTP.
- Transported over UDP or TCP protocols.
  - Security at the transport and network layer provided with TLS (requires TCP) or IPSec.
- Offers an alternative to the complex H.323 protocols.
- Due to its simpler nature, the protocol is becoming more popular than the H.323 family of protocols.
- SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs):
  - User-agent client (UAC) - A client application that initiates the SIP request.
  - User-agent server (UAS) - A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.
- A SIP endpoint is capable of functioning as both UAC and UAS.

SIP Functionality

- SIP supports five facets of establishing and terminating multimedia communications:
  - User location - determination of the end system to be used for communication;
  - User availability - determination of the willingness of the called party to engage in communications;
  - User capabilities - determination of the media and media parameters to be used;
  - Session setup - "ringing", establishment of session parameters at both called and calling party;
  - Session management - including transfer and termination of sessions, modifying session parameters, and invoking services.

SIP Clients and Servers

- SIP Clients
  - Phones (software based or hardware).
  - Gateways
  - User Agents
  - A User Agent acts as a
    - Client when it initiates a request (UAC),
    - Server when it responds to a request (UAS).
- SIP Servers
  - Proxy server
    - Receives SIP requests from a client and forwards them on the client's behalf.
    - Receives SIP messages and forward them to the next SIP server in the network.
    - Provides functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
  - Redirect server
    - Provides the client with information about the next hop or hops that a message should take and then the client contacts the next-hop server or UAS directly.
  - Registrar server
    - Processes requests from UACs for registration of their current location.
    - Registrar servers are often co-located with a redirect or proxy server.

SIP Messages

- SIP used for Peer-to-Peer Communication though it uses a Client-Server model.
- SIP is a text-based protocol and uses the UTF-8 charset.
- A SIP message is either a **request** from a client to a server, or a **response** from a server to a client.
  - A request message consists of a Request-Line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body;
  - A response message consists of a Status-Line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.
  - All lines (including empty ones) must be terminated by a carriage-return line-feed sequence (CRLF).

SIP Requests

- Requests are also called “Methods”.
- SIP uses SIP Uniform Resource Indicators (URI) to indicate the user or service to which a request is being addressed.
- The general form of a SIP Request-URI is:
  - sip:user:password@host:port;uri-parameters
    - sip:John@doe.com
    - sip:+14085551212@company.com
    - sip:alice@atlanta.com;maddr=239.255.255.1;ttl=15
  - Proxies and other servers route requests based on Request-URI.
- Requests are distinguished by starting with a Request-Line.
  - A Request-Line contains a **Method** name, a **Request-URI**, and **SIP-Version** separated by a single space (SP) character.
    - Request-Line = Method SP Request-URI SP SIP-Version CRLF
  - RFC 3261 defines six methods: INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER.
    - SIP extensions provide additional methods: SUBSCRIBE, NOTIFY, PUBLISH, MESSAGE, ...
  - SIP-Version should be “SIP/2.0”.
  - Example:
    - Request-Line: INVITE sip:2001@192.168.56.101 SIP/2.0
- The remaining of a request message is one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.

SIP Methods and Purposes

- INVITE
  - Requests the establishment of a session.
- ACK
  - Completes a three way session handshake (INVITE request, responses, ACK).
- OPTIONS
  - Requests the capabilities of another User Agent
  - Response lists supported methods, extensions, codecs, etc.
- BYE
  - Terminates an established session
    - User Agents stop sending media packets.
- CANCEL
  - Terminates a pending session (INVITE sent but no final response yet received).
- REGISTER
  - Allows a User Agent to upload current location.

SIP Responses

- SIP responses are distinguished from requests by starting with a Status-Line.
- A Status-Line consists of the **SIP-version** followed by a numeric **Status-Code** and its associated textual **Reason-Phrase**, with each element separated by a single SP character.
  - Status-Line = SIP-Version SP Status-Code SP Reason-Phrase CRLF
  - The Status-Code is a 3-digit integer code that indicates the outcome of an attempt to understand and satisfy a request.
  - The Reason-Phrase is intended to give a short textual description of the Status-Code.
    - The Status-Code is intended for use by automata, whereas the Reason-Phrase is intended for the human user.
    - A client is not required to examine or display the Reason-Phrase.
  - Example:
    - Status-Line: SIP/2.0 180 Ringing
- The remaining of a response message is one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.

SIP Responses Codes and Purposes

- The first digit of the Status-Code defines the class of response.
  - 1xx: Provisional - request received, continuing to process the request;
  - 2xx: Success - the action was successfully received, understood, and accepted;
  - 3xx: Redirection - further action needs to be taken in order to complete the request;
  - 4xx: Client Error - the request contains bad syntax or cannot be fulfilled at this server;
  - 5xx: Server Error - the server failed to fulfill an apparently valid request;
  - 6xx: Global Failure - the request cannot be fulfilled at any server.

- Common Response codes:
  - 100 Trying
    - The request has been received and that some unspecified action is being taken.
  - 180 Ringing
    - Trying to alert the user.
  - 200 OK
  - 301 Moved Permanently and 302 Moved Temporarily
    - User can no longer be found at the address in the Request-URI.
  - 400 Bad Request
    - Request could not be understood.
  - 401 Unauthorized
    - Request requires user authentication.
  - 403 Forbidden
    - Server understood the request, but is refusing to fulfill it.
  - 404 Not Found
    - Server has definitive information that the user does not exist.

SIP Header Fields

- A SIP header field has the form:
  - "header-name: header-value [;header-value;header-value;...;header-value]"
- Required Headers:
  - To
    - Specifies the logical recipient of the request.
    - `To: <sip:Vieira@192.168.56.1:5060>`
  - From
    - Indicates the initiator of the request.
    - `From: "PintoDaCosta" <sip:PintoDaCosta@192.168.56.102>;tag=as078bdcb2`
  - Via
    - Indicates the path taken by the request so far and indicates the path that should be followed in routing responses.
    - `Via: SIP/2.0/UDP 192.168.56.102:5060;branch=z9hG4bK1a8b6d0b;rport`
  - Call-ID
    - Uniquely identifies a particular invitation or all registrations of a particular client.
    - `Call-ID: 353befc372eaf28a57da5cf45ffc2a00@192.168.56.102:5060`
  - Cseq
    - Contains a single decimal sequence number and the request method. Serves to order transactions within a dialog.
    - `CSeq: 102 INVITE`
  - Max-Forwards
    - Limits the number of proxies or gateways that can forward the request.
- Common optional Headers:
  - User-Agent
    - Contains information about the UAC originating the request.
    - `User-Agent: Ekiga/4.0.1`
  - Authorization
    - Contains authentication credentials.
    - `Authorization: Digest username="Vieira", realm="asterisk", nonce="7d88f81c", uri="sip:2001@192.168.56.102", algorithm=MD5, response="b70474b5bbece20a68472e7ad4e37197"`
  - And many others...

SIP Message Body

- Requests may contain message bodies unless otherwise noted.
  - The interpretation of the body depends on the request method.
  - e.g.: INVITATION contains a description of the media session in another protocol. Usually SDP - Session Description Protocol (RFC 2327).
- For response messages, the request method and the response status code determine the type and interpretation of any message body.
- All responses MAY include a body.
- A message body length is provided by the Content-Length header field.

Session Description Protocol (SDP)

- SIP carries (encapsulates) SDP messages.
- When initiating multimedia teleconferences, VoIP calls, streaming video, or other sessions, is required to transmit to participants media details, transport addresses, and other session description metadata.
- SDP (RFC 4566) provides a standard representation for such information, irrespective of how that information is transported.
  - SDP is purely a format for session description.
  - SDP is intended to be general purpose so that it can be used in a wide range of network environments and applications.
  - SDP does not support negotiation of session content or media encodings.

SDP Session Description

- An SDP session description is entirely textual.
- Consists of a number of lines of text of the form *<type>=<value>*
  - *<type> is* one case-significant character.
  - *<value>* is structured text whose format depends on *<type>.*
- Consists of a session-level section followed by zero or more media-level sections.
  - The session-level part starts with a "v=" line and continues to the first media-level section.
  - Each media-level section starts with an "m=" line.

- Types

Session description
v= (protocol version)
o= (originator and session identifier)
s= (session name)
i=* (session information)
u=* (URI of description)
e=* (email address)
p=* (phone number)
c=* (connection information -- not required if included in all media)
b=* (zero or more bandwidth information lines)
One or more time descriptions ("t=" and "r=" lines; see below)
z=* (time zone adjustments)
k=* (encryption key)
a=* (zero or more session attribute lines)
Zero or more media descriptions

Time description
t= (time the session is active)
r=* (zero or more repeat times)

Media description, if present
m= (media name and transport address)
i=* (media title)
c=* (connection information -- optional if included at session level)
b=* (zero or more bandwidth information lines)
k=* (encryption key)
a=* (zero or more media attribute lines)

SDP Payloads

```
▽ Session Initiation Protocol (INVITE)
  ▷ Request-Line: INVITE sip:9001@192.168.56.101:5060 SIP/2.0
  ▷ Message Header
  ▽ Message Body
    ▽ Session Description Protocol
        Session Description Protocol Version (v): 0
      ▷ Owner/Creator, Session Id (o): - 1442419018 2 IN IP4 192.168.56.1
        Session Name (s): Ekiga/4.0.1
      ▷ Connection Information (c): IN IP4 192.168.56.1
      ▷ Time Description, active time (t): 0 0
      ▷ Media Description, name and address (m): audio 5066 RTP/AVP 9 101
        Media Attribute (a): sendrecv
      ▷ Media Attribute (a): rtpmap:9 G722/8000/1
      ▷ Media Attribute (a): rtpmap:101 telephone-event/8000
      ▷ Media Attribute (a): fmtp:101 0-16,32,36
      ▷ Media Attribute (a): maxptime:90
```

v=0

O= - 1442419018 2 IN IP4 192.168.56.1

s=Ekiga/4.0.1

c=IN IP4 192.168.56.1

t=0 0

m=audio 5066 RTP/AVP 9 101

a=sendrecv

a=rtpmap:9 G722/8000/1

a=rtpmap:101 telephone-event/8000

a=fmtp:101 0-16,32,36

a=maxptime:90

v=0

o=root 84591410 84591411 IN IP4 192.168.56.101

s=Asterisk PBX 1.8.10.1~dfsg-1ubuntu1

c=IN IP4 192.168.56.101

t=0 0

m=audio 13128 RTP/AVP 9 101

a=rtpmap:9 G722/8000

a=rtpmap:101 telephone-event/8000

a=fmtp:101 0-16

a=ptime:20

a=sendrecv

```
▽ Session Initiation Protocol (200)
  ▷ Status-Line: SIP/2.0 200 OK
  ▷ Message Header
  ▽ Message Body
    ▽ Session Description Protocol
        Session Description Protocol Version (v): 0
      ▷ Owner/Creator, Session Id (o): root 84591410 84591411 IN IP4 192.168.56.101
        Session Name (s): Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
      ▷ Connection Information (c): IN IP4 192.168.56.101
      ▷ Time Description, active time (t): 0 0
      ▷ Media Description, name and address (m): audio 13128 RTP/AVP 9 101
      ▷ Media Attribute (a): rtpmap:9 G722/8000
      ▷ Media Attribute (a): rtpmap:101 telephone-event/8000
      ▷ Media Attribute (a): fmtp:101 0-16
      ▷ Media Attribute (a): ptime:20
        Media Attribute (a): sendrecv
```

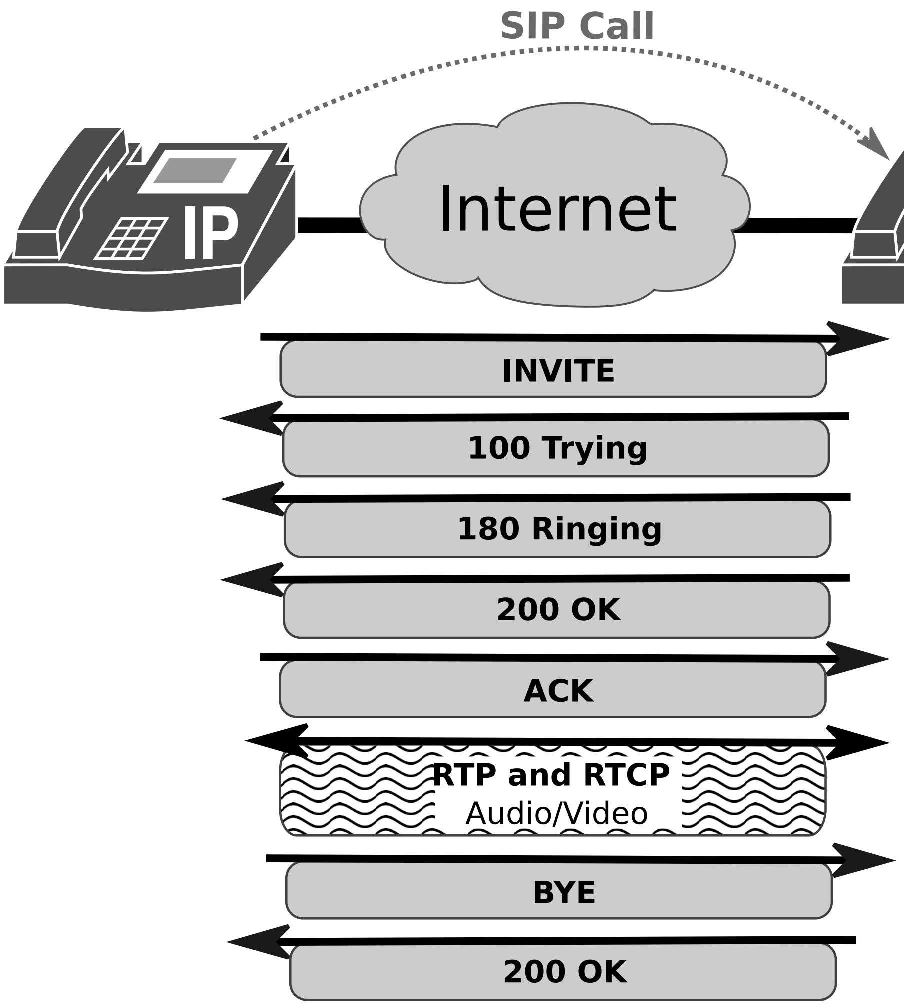
Sample SIP INVITE Request

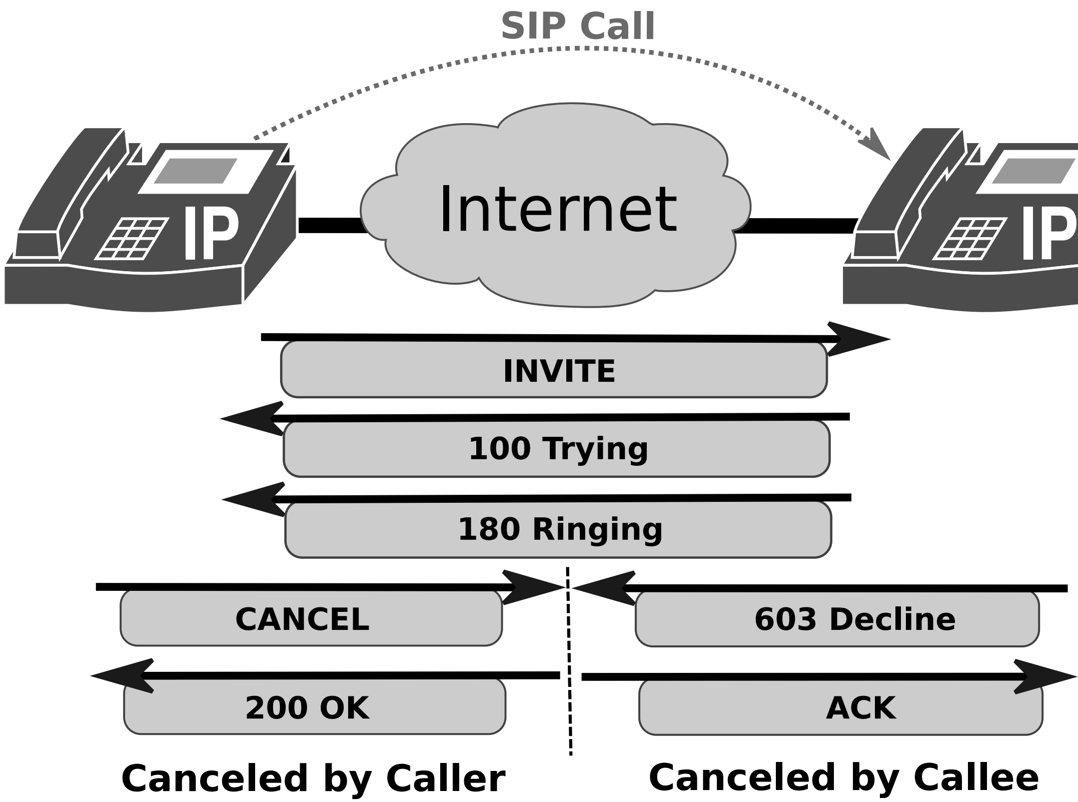
```
▽ Session Initiation Protocol (INVITE)
  ▽ Request-Line: INVITE sip:2001@192.168.56.102:5060 SIP/2.0
      Method: INVITE
    ▷ Request-URI: sip:2001@192.168.56.102:5060
      [Resent Packet: False]
  ▽ Message Header
    ▽ CSeq: 3 INVITE
        Sequence Number: 3
        Method: INVITE
    ▷ Via: SIP/2.0/UDP 192.168.56.1:5060;branch=z9hG4bK84ca3e68-9f5b-e511-99a7-7824afcb1a1a;rport
      User-Agent: Ekiga/4.0.1
    ▷ Authorization: Digest username="Vieira", realm="asterisk", nonce="4c4ee187", uri="sip:2001@192.168.56.102:5060", algorithm=MD5, response="c125cfcc695b41b85
    ▷ From: <sip:Vieira@192.168.56.102>;tag=08649666-9f5b-e511-99a7-7824afcb1a1a
      Call-ID: 38669666-9f5b-e511-99a7-7824afcb1a1a@SalAsus
      Supported: 100rel,replaces
    ▷ To: <sip:2001@192.168.56.102>;tag=as41553a03
    ▷ Contact: <sip:Vieira@192.168.56.1:5060>
      Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,SUBSCRIBE,NOTIFY,REFER,MESSAGE,INFO,PING,PRACK
      Content-Length: 225
      Content-Type: application/sdp
      Max-Forwards: 70
  ▽ Message Body
    ▽ Session Description Protocol
        Session Description Protocol Version (v): 0
      ▷ Owner/Creator, Session Id (o): - 1442490388 2 IN IP4 192.168.56.1
        Session Name (s): Ekiga/4.0.1
      ▷ Connection Information (c): IN IP4 192.168.56.1
      ▷ Time Description, active time (t): 0 0
      ▷ Media Description, name and address (m): audio 5084 RTP/AVP 9 101
        Media Attribute (a): sendrecv
      ▷ Media Attribute (a): rtpmap:9 G722/8000/1
      ▷ Media Attribute (a): rtpmap:101 telephone-event/8000
      ▷ Media Attribute (a): fmtp:101 0-16,32,36
      ▷ Media Attribute (a): maxptime:90
```

Sample SIP 200 OK Response

```
▽ Session Initiation Protocol (200)
  ▽ Status-Line: SIP/2.0 200 OK
      Status-Code: 200
      [Resent Packet: False]
      [Request Frame: 1524]
      [Response Time (ms): 2745]
  ▽ Message Header
    ▷ Via: SIP/2.0/UDP 192.168.56.1:5060;branch=z9hG4bKca459866-9f5b-e511-99a7-7824afcb1a1a;received=192.168.56.1;rport=5060
    ▷ From: <sip:Vieira@192.168.56.102>;tag=08649666-9f5b-e511-99a7-7824afcb1a1a
    ▷ To: <sip:2001@192.168.56.102>;tag=as41553a03
      Call-ID: 38669666-9f5b-e511-99a7-7824afcb1a1a@SalAsus
    ▽ CSeq: 2 INVITE
        Sequence Number: 2
        Method: INVITE
      Server: Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
      Supported: replaces, timer
    ▷ Contact: <sip:2001@192.168.56.102:5060>
      Content-Type: application/sdp
      Content-Length: 558
  ▽ Message Body
    ▽ Session Description Protocol
        Session Description Protocol Version (v): 0
      ▷ Owner/Creator, Session Id (o): root 1089037721 1089037721 IN IP4 192.168.56.102
        Session Name (s): Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
      ▷ Connection Information (c): IN IP4 192.168.56.102
      ▷ Time Description, active time (t): 0 0
      ▷ Media Description, name and address (m): audio 13496 RTP/AVP 3 0 8 115 91 118 9 105 116 101
      ▷ Media Attribute (a): rtpmap:3 GSM/8000
      ▷ Media Attribute (a): rtpmap:0 PCMU/8000
      ▷ Media Attribute (a): rtpmap:8 PCMA/8000
      ▷ Media Attribute (a): rtpmap:115 speex/8000
      ▷ Media Attribute (a): rtpmap:91 iLBC/8000
      ▷ Media Attribute (a): fmtp:91 mode=30
      ▷ Media Attribute (a): rtpmap:118 G726-32/8000
      ▷ Media Attribute (a): rtpmap:9 G722/8000
      ▷ Media Attribute (a): rtpmap:105 G7221/16000
      ▷ Media Attribute (a): fmtp:105 bitrate=32000
      ▷ Media Attribute (a): rtpmap:116 speex/16000
      ▷ Media Attribute (a): rtpmap:101 telephone-event/8000
      ▷ Media Attribute (a): fmtp:101 0-16
      ▷ Media Attribute (a): ptime:20
        Media Attribute (a): sendrecv
      ▷ Media Description, name and address (m): video 0 RTP/AVP 31 34 94 89 92 95 126
```

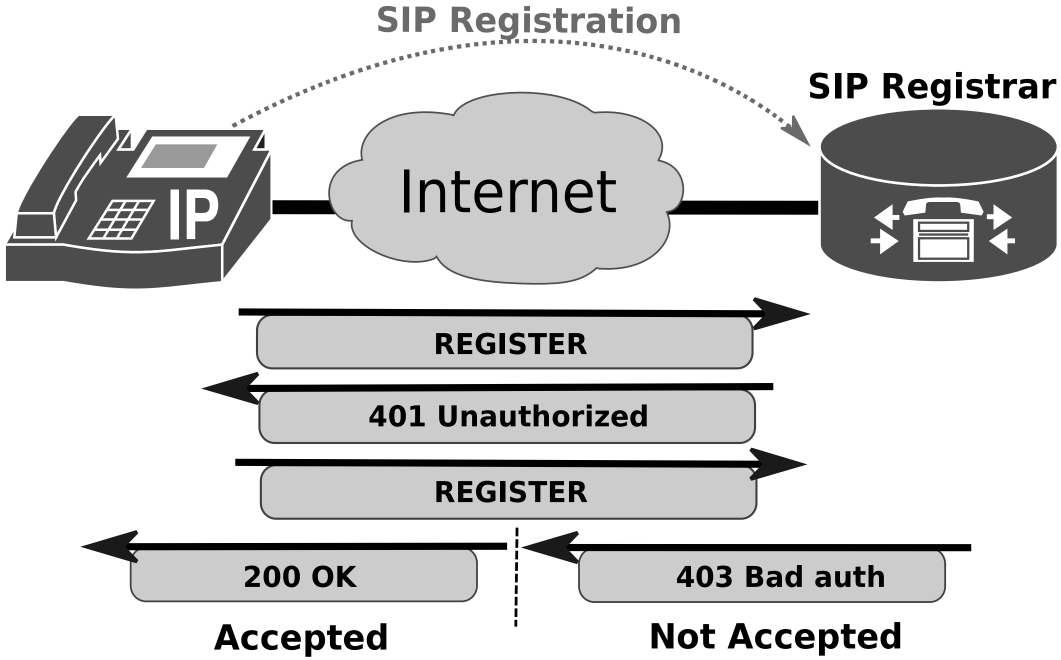
SIP Signaling – Direct Call





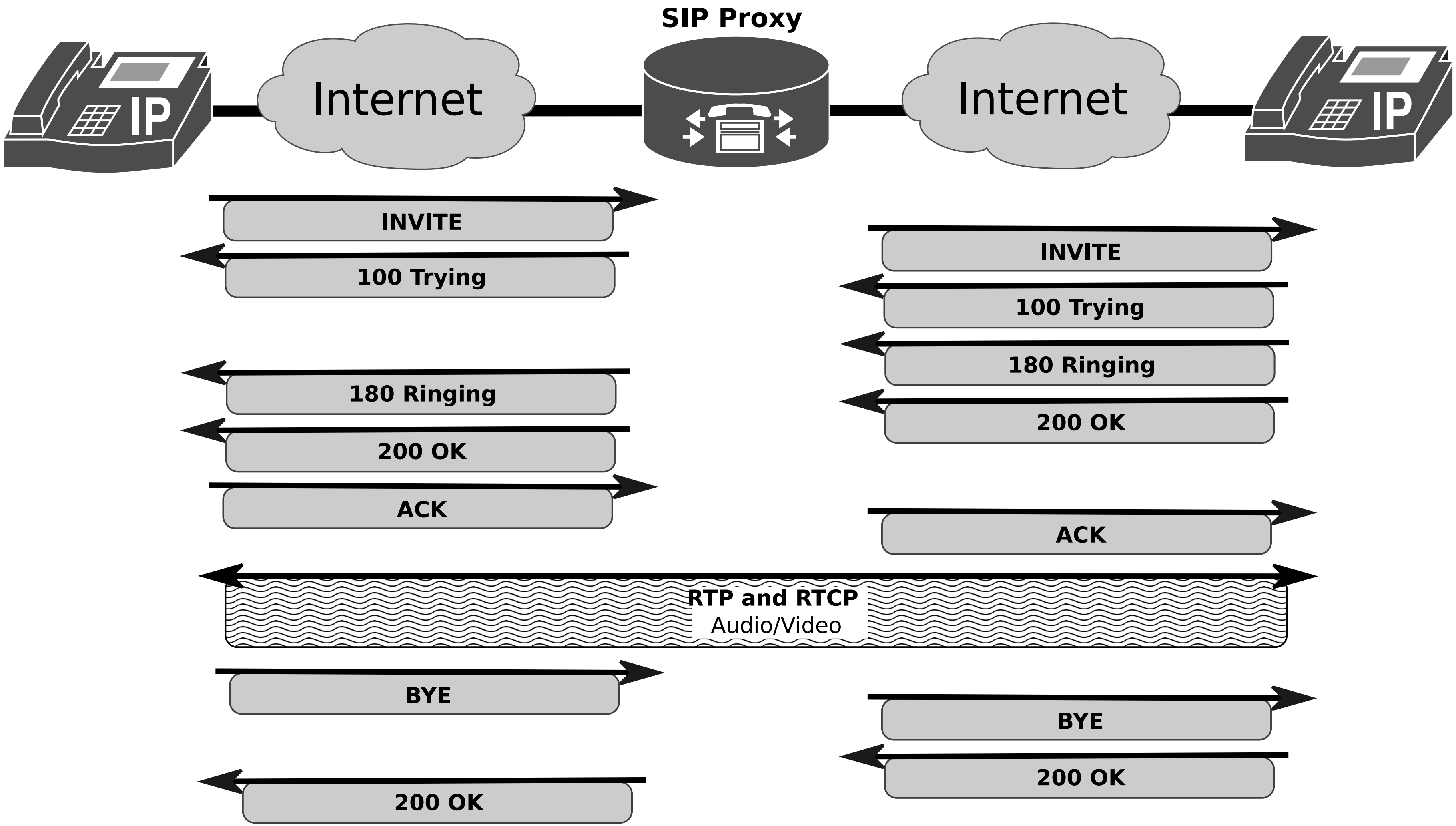
SIP Registrar Server

- SIP Registrar servers store the location of SIP endpoints.
- A user has an account created which allows them to REGISTER contacts with a particular server.
- The account specifies a SIP “Address of Record (AOR)”
- Each SIP endpoint Registers with a Registrar server with a SIP REGISTER request.
  - Using it’s Address of Record and Contact address.
- Address of Record is in From header:
  - `From: <sip:Vieira@192.168.56.102>`
- Contact header tells Registrar server where to send messages:
  - `Contact: <sip:Vieira@192.168.56.1:5060>`
- SIP Proxy servers query SIP Registrar servers for routing information.



- Registration usually requires authentication.
- If REGISTER has no authentication credentials, the SIP Registrar server responds with 401 Unauthorized.
- End-point resends REGISTER with an Authorization header with credentials.
  - `Authorization: Digest username="Vieira", realm="asterisk", nonce="7d88f81c", uri="sip:2001@192.168.56.102", algorithm=MD5, response="b70474b5bbece20a68472e7ad4e37197"`
- Server accepts registration with a 200 OK response.
- Server rejects credentials with a 401 Bad Auth response.

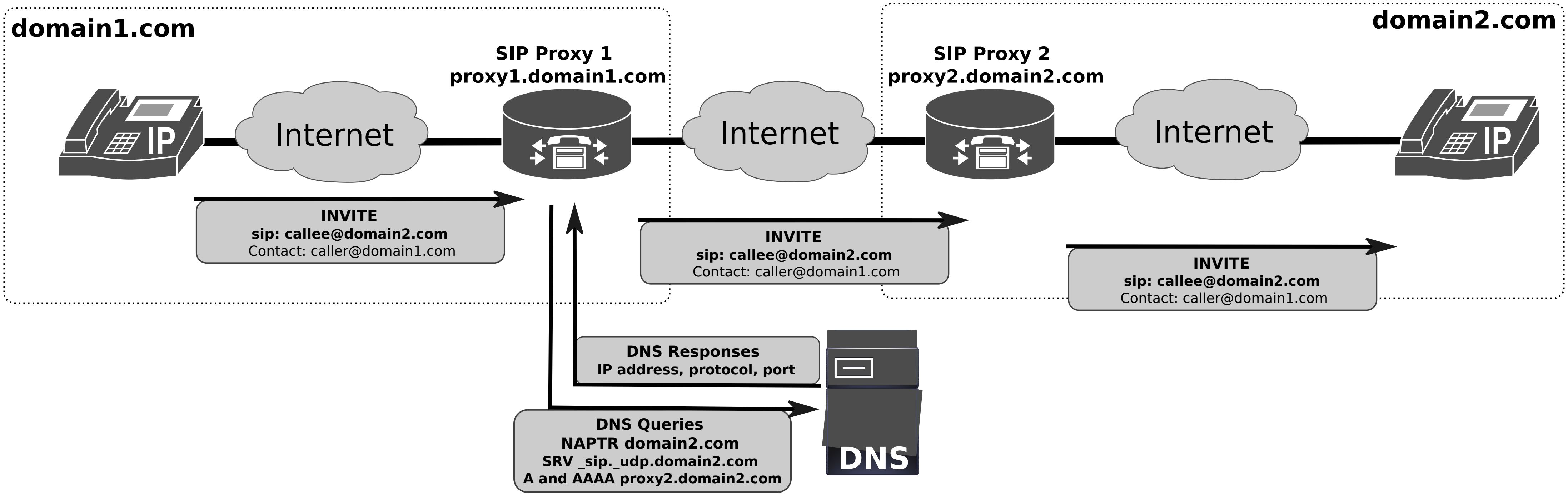
SIP Proxy Server



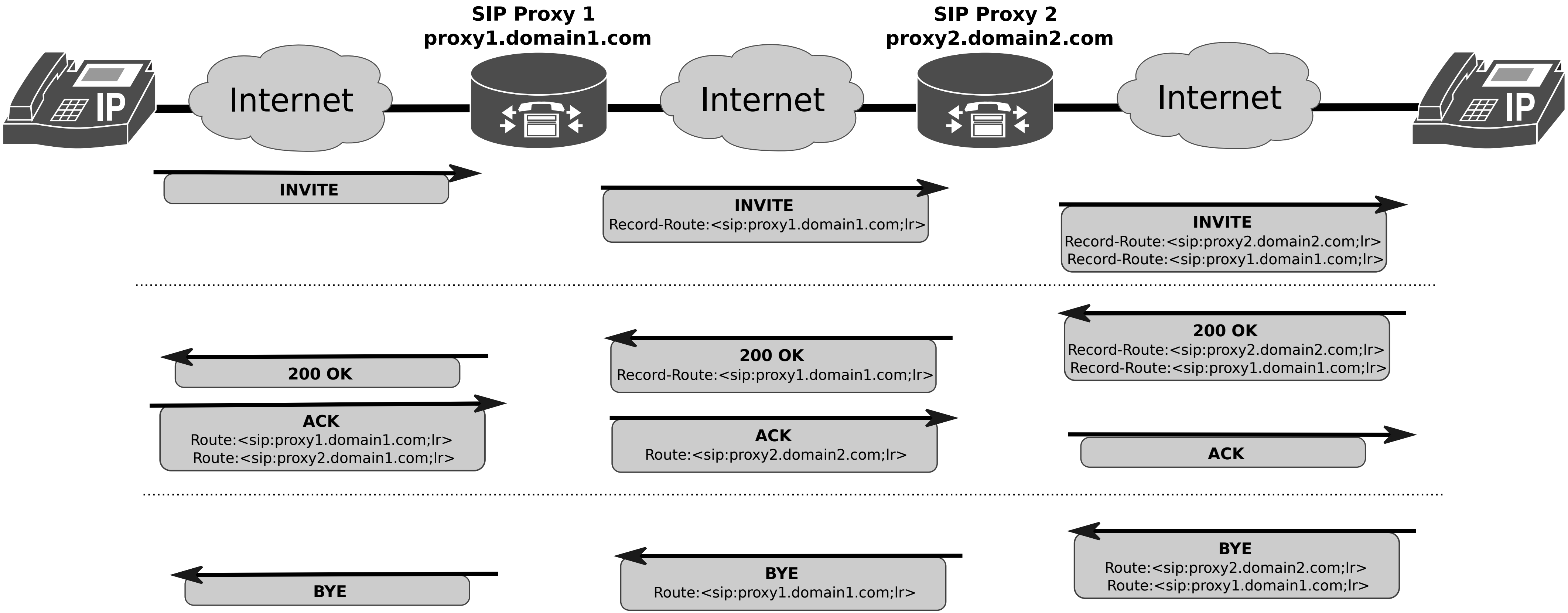
Locating SIP Servers

- RFC 3263 defines a set of DNS procedures to locate SIP Servers.
- SIP elements need to send requests/responses to a resource identified by a SIP URI.
  - The SIP URI may identify the desired target resource or a intermediate hop towards that resource.
  - Requires **Transport protocol**, **IP address** and **Port.**
    - If the URI specifies any of them, then it should be used.
  - Otherwise, must be retrieved from a DNS server.
    - Using **Service (SRV)** and **Name Authority Pointer (NAPTR)** DNS records.
- NAPTR records provide a mapping from a domain name to:
  - A SRV record (that contains the resource responsible server name),
  - And, the specific transport protocol.
- Example:
  - A client/server that wishes to resolve "sip:user@example.com",
  - Performs a NAPTR query for domain "example.com",
    - `IN NAPTR 100  50  "s"  "SIP+D2U"  ""  _sip._udp.example.com.`
  - Has UDP as possible transport protocol, performs a SRV query for "_sip._udp.example.com"
    - `IN SRV  0  1  5060  server1.example.com`
    - `IN SRV  0  2  5060  server2.example.com`
  - Has two possible servers, performs A and AAAA queries for the chosen server.

SIP Proxy Forwarding

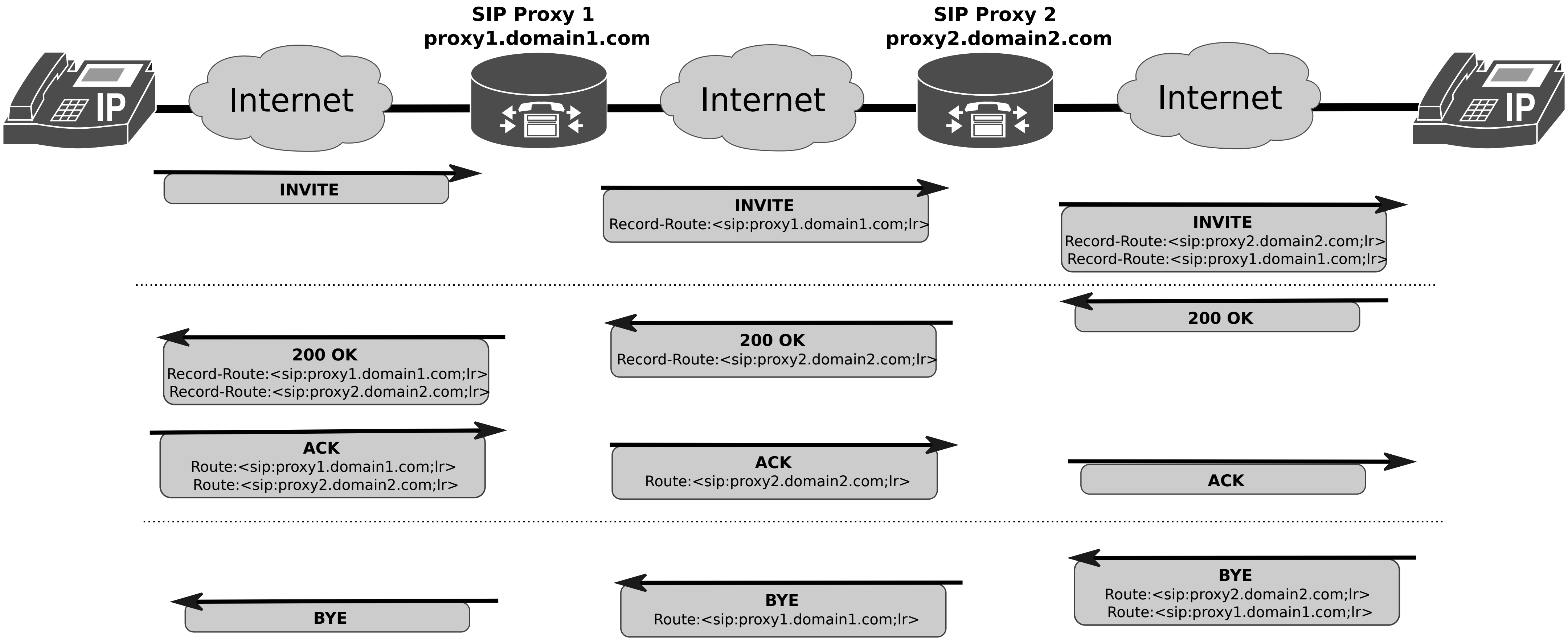


SIP Record-Route and Route Headers



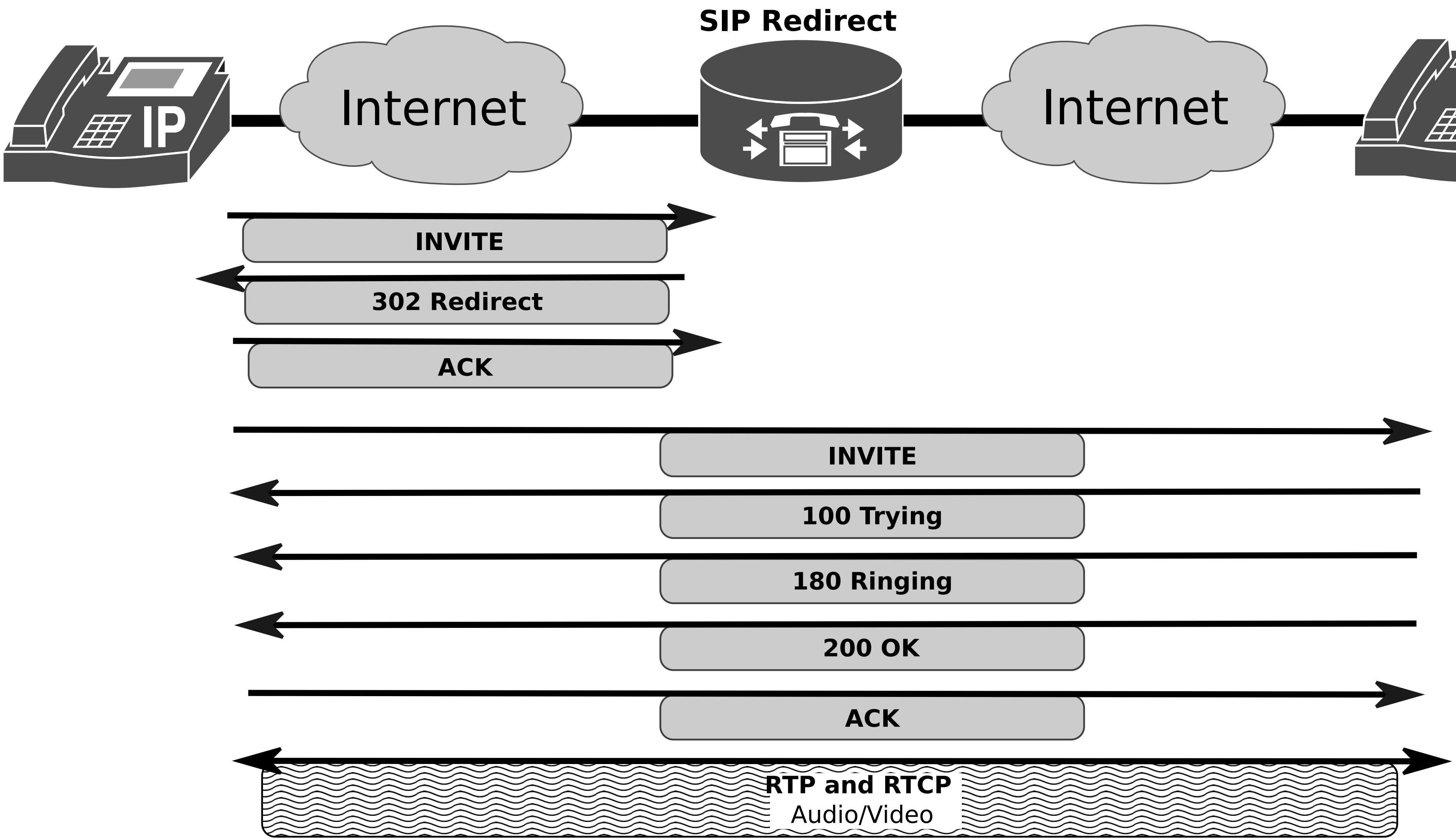
- **Record-Route** headers are used to list intermediary hops.
- **Route** headers are used to define a routing path.
- The lr parameter indicates that the element responsible for this resource implements routing mechanisms.

SIP Record-Route and Route Headers



- **Record-Route** headers are used to list intermediary hops.
- **Route** headers are used to define a routing path.
- The lr parameter indicates that the element responsible for this resource implements routing mechanisms.

SIP Redirect Server



- A Redirect server may redirect to the desired target or a intermediate hop towards that target.

SIP Presence and Instant Messaging

- SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)
  - Provides for presence and buddy lists,
  - Instant Messaging in the enterprise,
  - Telephony enabled user lists.
- Presence
  - SIP-Specific Event Notification (RFC 6665).
    - SUBSCRIBE and NOTIFY methods.
  - Session Initiation Protocol (SIP) Extension for Event State Publication (RFC 3903)
    - PUBLISH mechanism.
- Instant Messaging
  - Page Mode
    - Doesn't require a session. Uses MESSAGE method (RFC 3428).
  - Session Mode
    - Message Session Relay Protocol (RFC 4975, RFC 4976).
    - Text-based protocol for exchanging content between users
    - Requires the establishment of an MSRP session.
      - Set-up using MSRP URI, within SIP and SDP signaling.

SIP for Presence

- The SUBSCRIBE method is used to request current state and state updates/notifications from a remote node for a specific event.
  - Must contain an "Event" header field with information to identify the resource for which event notification is desired.
    - e.g., Voicemail (`Event: message-summary`).
  - Should contain an "Expires" header field indicating the duration of the subscription.
    - Unsubscribing is handled as refreshing a subscription, with the "Expires" header field set to "0".
  - May contain an "Accept" header field indicating the body formats allowed in notifications.
- The NOTIFY requests are sent to inform subscribers of changes in state (events) to which the subscriber has a subscription.
  - Does not terminate its corresponding subscription.
- 200 OK responses are used to acknowledge SUBSCRIBE and NOTIFY requests.
- The PUBLISH method is used to create, modify, and remove an event state.
  - e.g., Presence (away, busy, available, etc...) - `Event: presence`

Sample SUBSCRIBE and NOTIFY

```
▽ Session Initiation Protocol (SUBSCRIBE)
  ▷ Request-Line: SUBSCRIBE sip:PintoDaCosta@192.168.56.102 SIP/2.0
  ▽ Message Header
    ▷ CSeq: 2 SUBSCRIBE
    ▷ Via: SIP/2.0/UDP 10.0.2.15:5060;branch=z9hG4bK5f5c8d9e-af10-1910-9cfa-0800270fe441;rport
      User-Agent: Ekiga/4.0.2
    ▷ Authorization: Digest username="PintoDaCosta", realm="asterisk", nonce="48f80483", uri="s
    ▷ From: <sip:PintoDaCosta@192.168.56.102>;tag=0b5c8d9e-af10-1910-9cf9-0800270fe441
      Call-ID: 0b5c8d9e-af10-1910-9cf8-0800270fe441@Win81
    ▷ To: <sip:PintoDaCosta@192.168.56.102>
      Accept: application/simple-message-summary
    ▷ Contact: <sip:PintoDaCosta@192.168.56.1:56079>
      Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,SUBSCRIBE,NOTIFY,REFER,MESSAGE,INFO,PING,PRACK
      Expires: 3600
      Event: message-summary
      Content-Length: 0
      Max-Forwards: 70
```

```
▽ Session Initiation Protocol (NOTIFY)
  ▷ Request-Line: NOTIFY sip:PintoDaCosta@192.168.56.1:56079 SIP/2.0
  ▽ Message Header
    ▷ Via: SIP/2.0/UDP 192.168.56.102:5060;branch=z9hG4bK6c68d274;rport
      Max-Forwards: 70
    ▷ Route: <sip:PintoDaCosta@192.168.56.1:56079>
    ▷ From: "asterisk" <sip:asterisk@192.168.56.102>;tag=as0f02fcd3
    ▷ To: <sip:PintoDaCosta@192.168.56.1:56079>;tag=0b5c8d9e-af10-1910-9cf9-0800270fe441
    ▷ Contact: <sip:asterisk@192.168.56.102:5060>
      Call-ID: 0b5c8d9e-af10-1910-9cf8-0800270fe441@Win81
    ▷ CSeq: 102 NOTIFY
      User-Agent: Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
      Event: message-summary
      Content-Type: application/simple-message-summary
      Subscription-State: active
      Content-Length: 95
  ▽ Message Body
      Messages-Waiting: yes\r\n
      Message-Account: sip:asterisk@192.168.56.102\r\n
      Voice-Message: 1/0 (0/0)\r\n
```

Sample PUBLISH

```
Session Initiation Protocol (PUBLISH)
 Request-Line: PUBLISH sip:Vieira@192.168.56.102 SIP/2.0
 Message Header
  CSeq: 35 PUBLISH
  Via: SIP/2.0/UDP 193.136.93.144:5060;branch=z9hG4bKf09839d5-1f61-e511-9914-7824afcb1a1a;rport
  User-Agent: Ekiga/4.0.1
  From: <sip:Vieira@192.168.56.102>
  Call-ID: 9ef4fa6e-1f61-e511-9914-7824afcb1a1a@SalAsus
  To: <sip:Vieira@192.168.56.102>
  Expires: 300
  Event: presence
  Content-Length: 551
  Content-Type: application/pidf+xml
  Max-Forwards: 70
 Message Body
  eXtensible Markup Language
   <?xml
    version="1.0"
    encoding="UTF-8"
    ?>
   <presence
    xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    entity="pres:Vieira@192.168.56.102">
    <tuple
     id="TCA427E12">
     <status>
     <contact
     <timestamp>
     </tuple>
    <dm:person
     id="p8">
     <rpid:activities>
      <rpid:busy/>
      </rpid:activities>
     </dm:person>
    </presence>
```

- Content-Type header defines content format.
  - e.g., XML.
- Message Body contains presence description.

SIP for Instant Message (IM)

- The MESSAGE method (an extension to SIP) allows the transfer of Instant Messages (IM).
- MESSAGE requests carry the content in the form of MIME body parts.
  - Content-Type header defines content format.
- MESSAGE requests do not themselves initiate a SIP dialog.
  - May be sent in the context of a dialog initiated by some other SIP request.

```
▽Session Initiation Protocol (MESSAGE)
 ▷Request-Line: MESSAGE sip:2001@192.168.56.102 SIP/2.0
 ▽Message Header
  ▷CSeq: 29 MESSAGE
  ▷Via: SIP/2.0/UDP 192.168.56.1:5060;branch=z9hG4bK6abbfdfc-2361-e511-8e33-7824afcb1a1a;rport
   User-Agent: Ekiga/4.0.1
  ▷From: <sip:Vieira@192.168.56.102>
   Call-ID: d0affdfc-2361-e511-8e33-7824afcb1a1a@SalAsus
  ▷To: <sip:2001@192.168.56.102>
   Expires: 5000
   Content-Length: 5
   Content-Type: text/plain;charset=UTF-8
   Max-Forwards: 70
 ▽Message Body
  ▽Line-based text data: text/plain
    teste
```

DTMF Tones

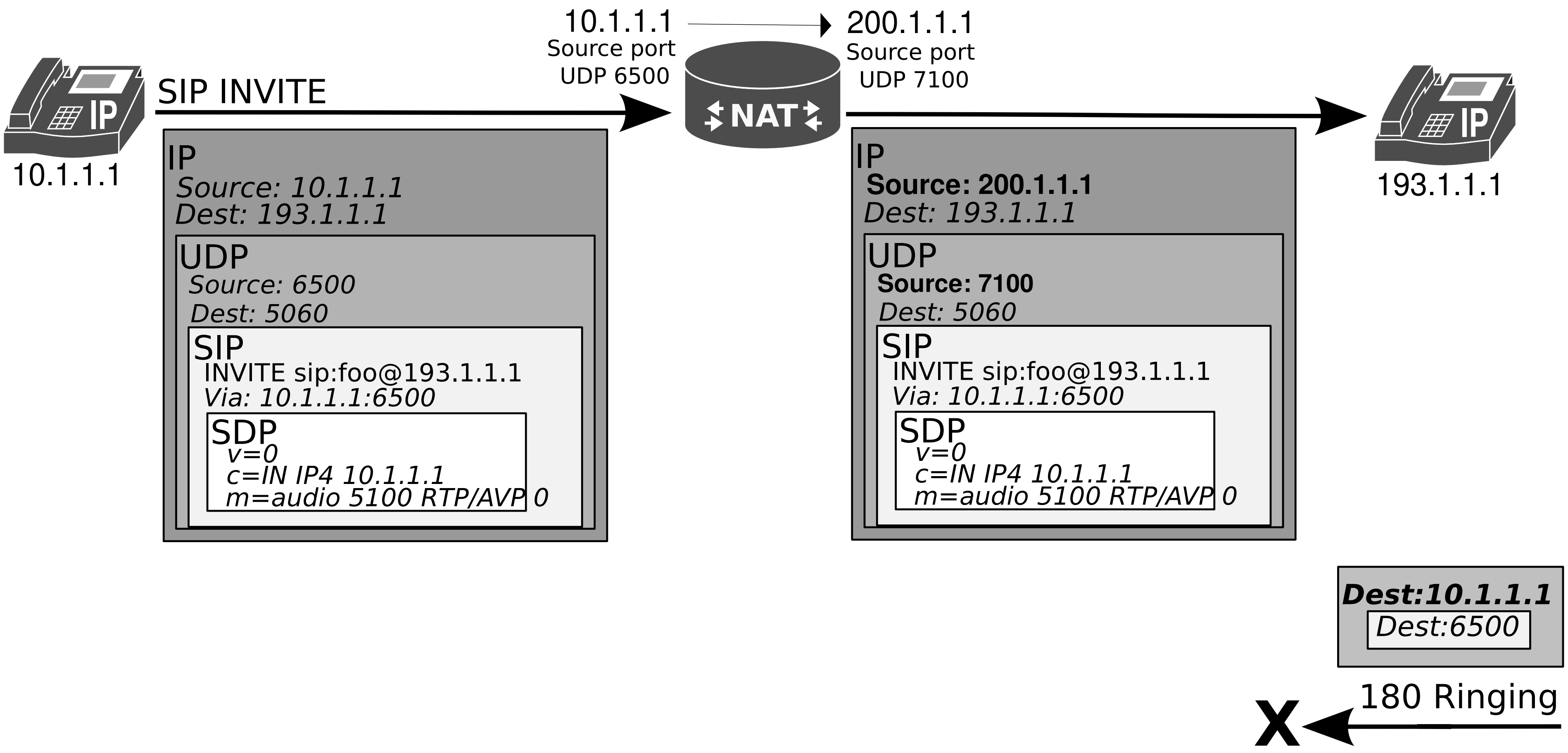
- RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals (RFC 4733 which obsoletes RFC 2833).

```
▷ Internet Protocol Version 4, Src: 192.168.56.1 (192.168.56.1), Dst: 192.168.56.101
▷ User Datagram Protocol, Src Port: 5070 (5070), Dst Port: 17960 (17960)
▷ Real-Time Transport Protocol
▽ RFC 2833 RTP Event
    Event ID: DTMF One 1 (1)
    1... .... = End of Event: True
    .0.. .... = Reserved: False
    ..00 0111 = Volume: 7
    Event Duration: 1440
```

- SIP INFO Method (RFC 6086)

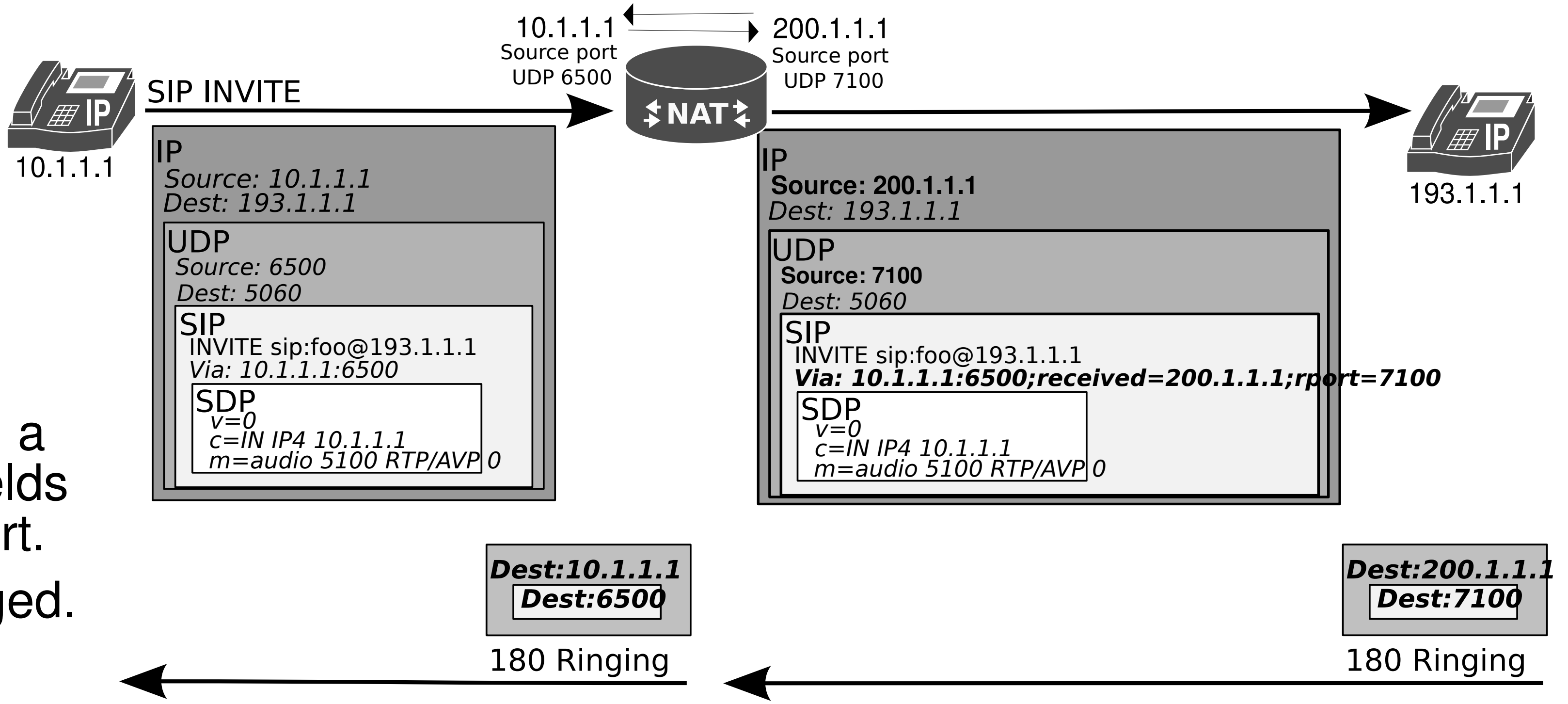
```
▷ Internet Protocol Version 4, Src: 192.168.56.1 (192.168.56.1), Dst: 192.168.56.101
▷ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
▽ Session Initiation Protocol (INFO)
  ▷ Request-Line: INFO sip:9001@192.168.56.101:5060 SIP/2.0
  ▷ Message Header
  ▽ Message Body
      Signal= 2\r\n
      Duration= 180\r\n
```

SIP and NAPT



SIP INVITE
10.1.1.1
10.1.1.1
Source port
UDP 6500
200.1.1.1
Source port
UDP 7100
NAT
193.1.1.1
IP
Source: 10.1.1.1
Dest: 193.1.1.1
UDP
Source: 6500
Dest: 5060
SIP
INVITE sip:foo@193.1.1.1
Via: 10.1.1.1:6500
SDP
v=0
c=IN IP4 10.1.1.1
m=audio 5100 RTP/AVP 0
IP
Source: 200.1.1.1
Dest: 193.1.1.1
UDP
Source: 7100
Dest: 5060
SIP
INVITE sip:foo@193.1.1.1
Via: 10.1.1.1:6500
SDP
v=0
c=IN IP4 10.1.1.1
m=audio 5100 RTP/AVP 0
Dest:10.1.1.1
Dest:6500
180 Ringing
X

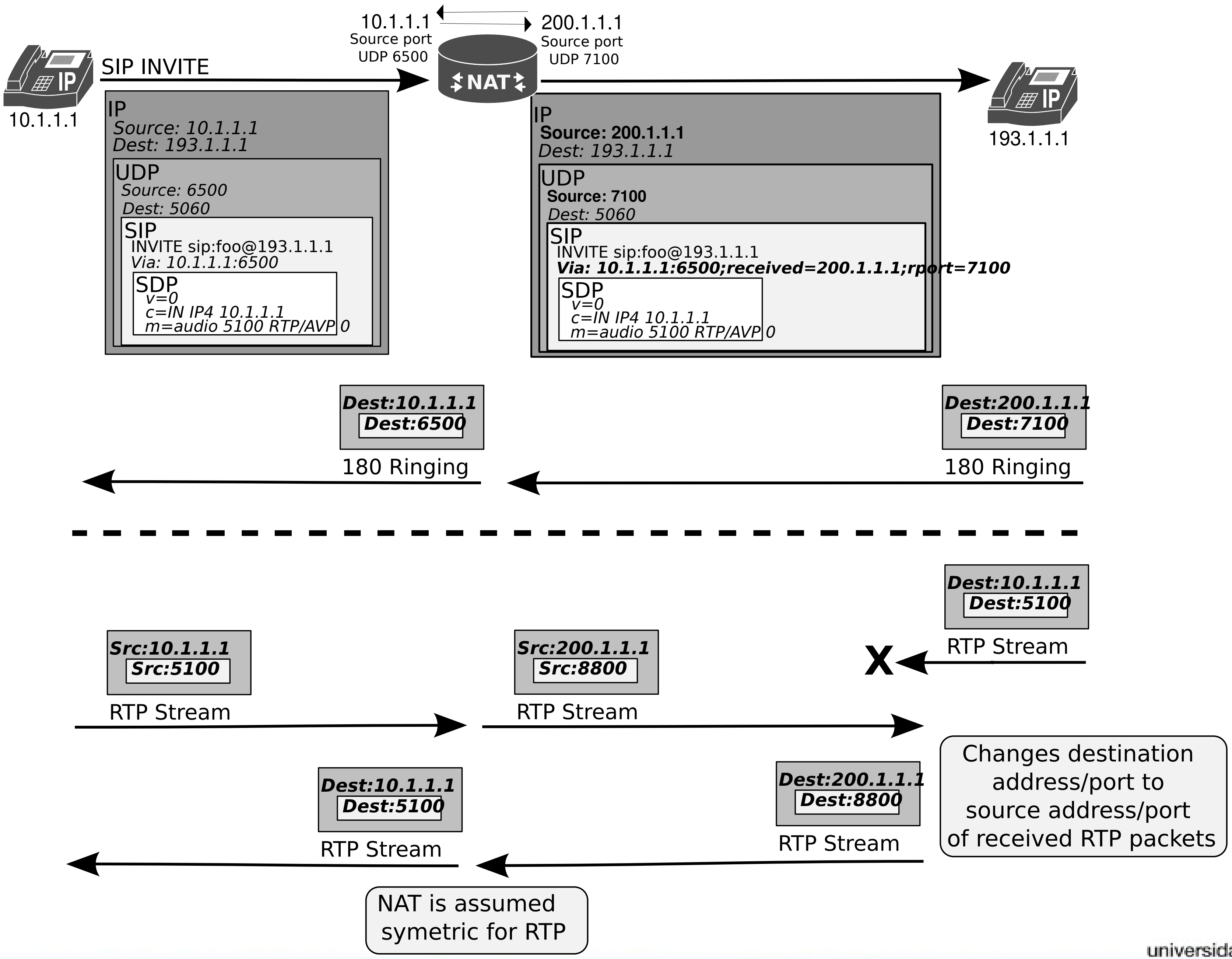
SIP NAPT Traversal



- Symmetric Response Routing (RFC 3581).
- SIP payload is also "translated", by adding a **received** and **rport** fields with public address/port.
- SDP remains unchanged.

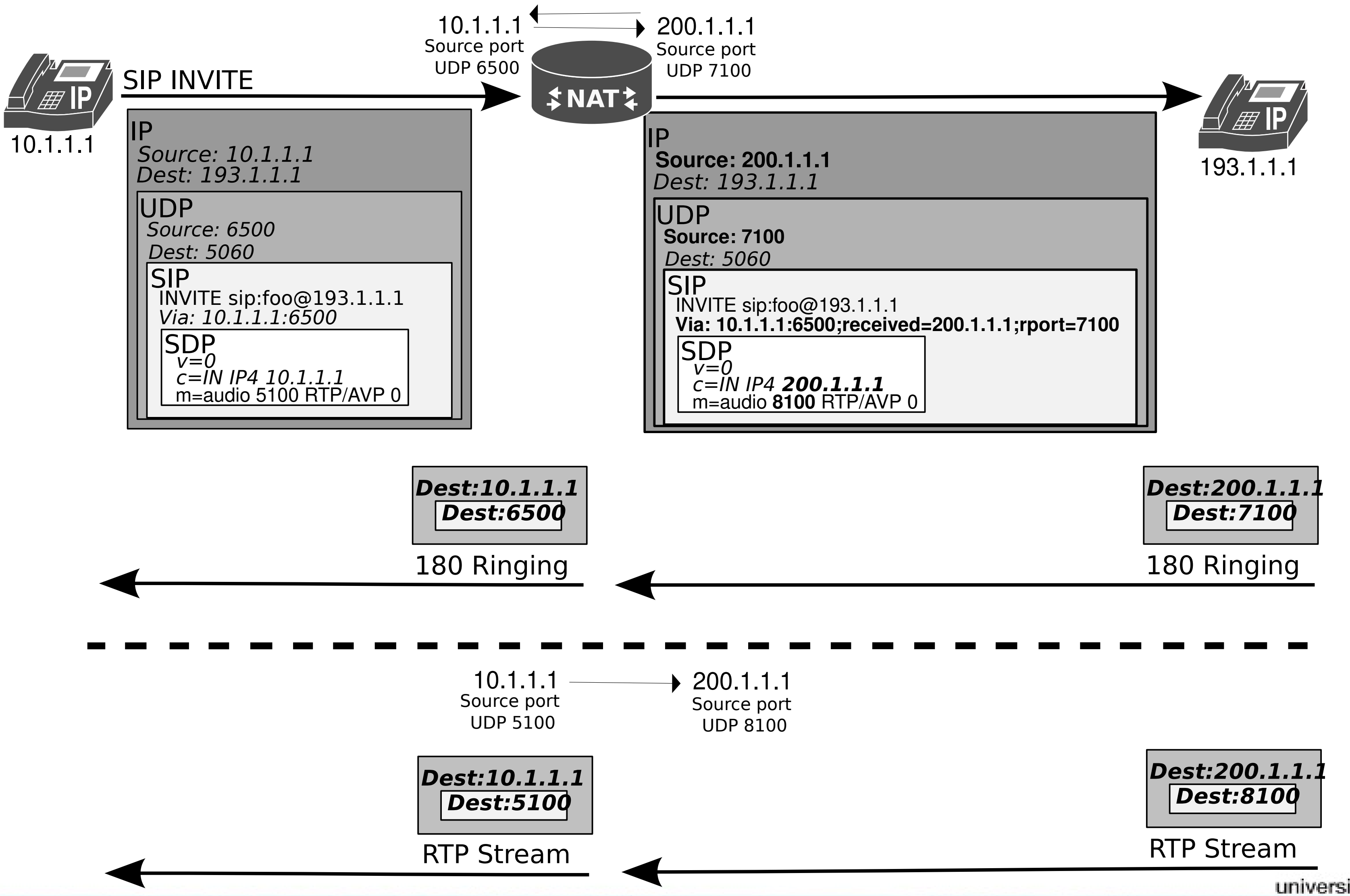
- Media traversal (RTP/RTCP) is still a problem.
  - SDP contents mismatch with public address/port.
  - Possible solutions
    - Let clients (on private network) find out their public address/port and rewrite SDP payload.
      - Manual configuration (when NAT uses static translations).
      - Automatic discovery (when NAT is dynamic) using STUN protocol.
    - Symmetric (RTP/RTCP) NAT (RFC 4961).
    - NAT SIP Application Layer Gateway (ALG).

Symmetric (RTP/RTCP) NAT



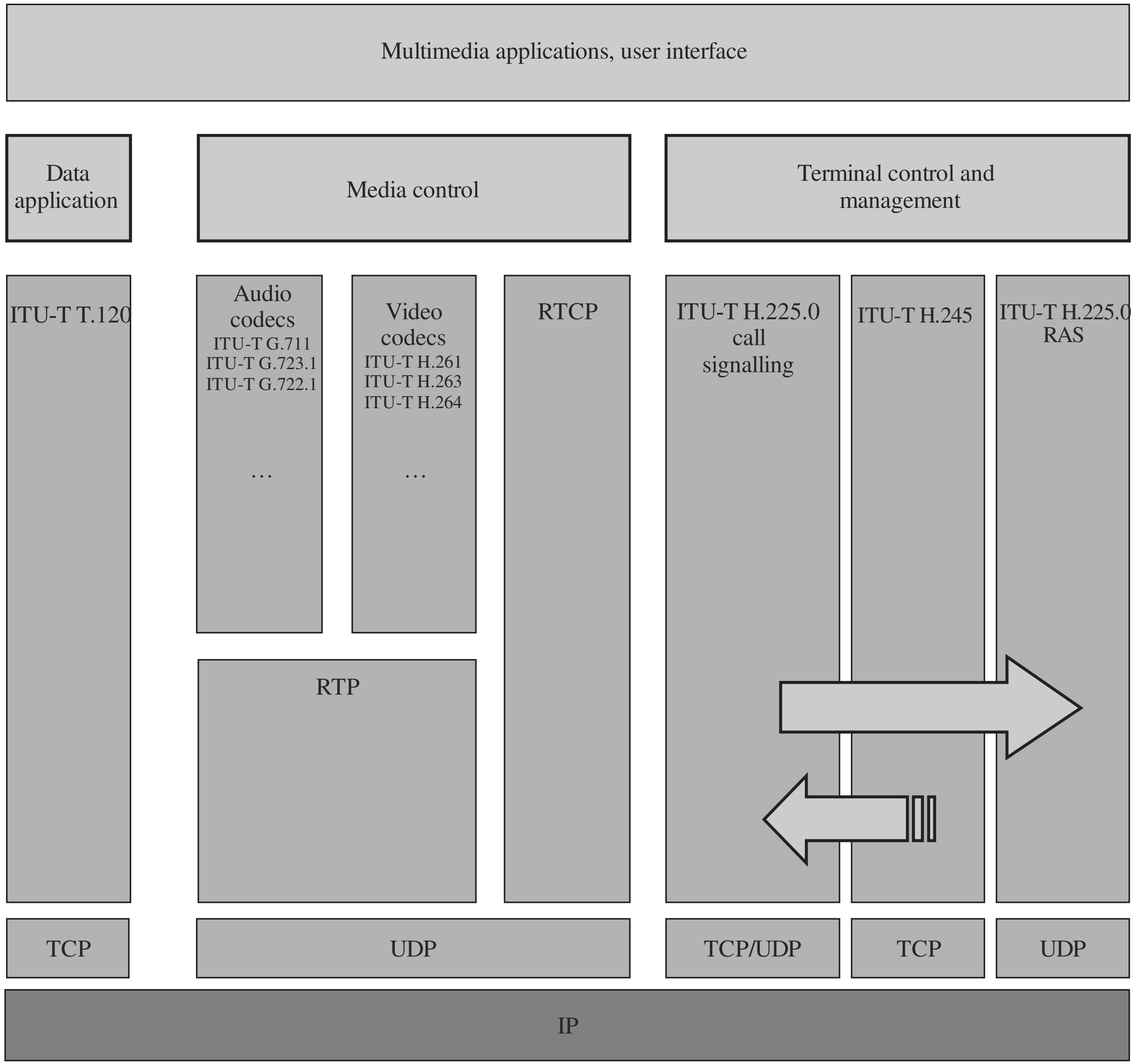
NAT SIP Application Layer Gateway (ALG)

- Required to translate SDP payloads.
- Heavy on NAT gateway.



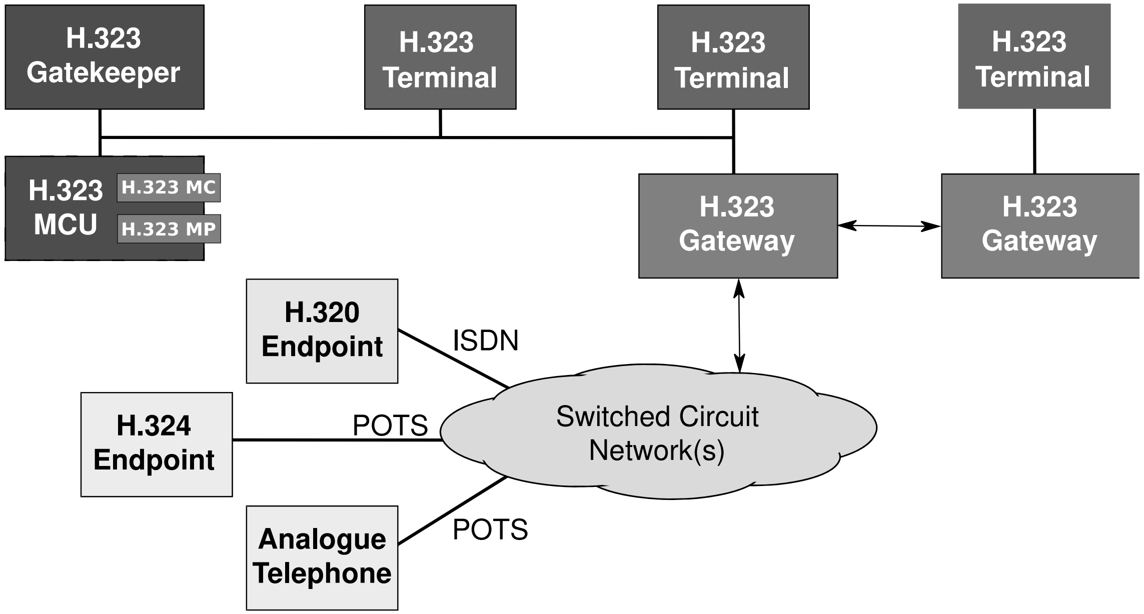
H.323

- H.323 is a set of recommendations from the International Telecommunication Union (ITU).
  - Contains several standards (signaling, control, transport, etc...).
- Consists of family of protocols that are used for call set-up, call termination, registration, authentication and other functions.
- Are transported over TCP or UDP protocols.



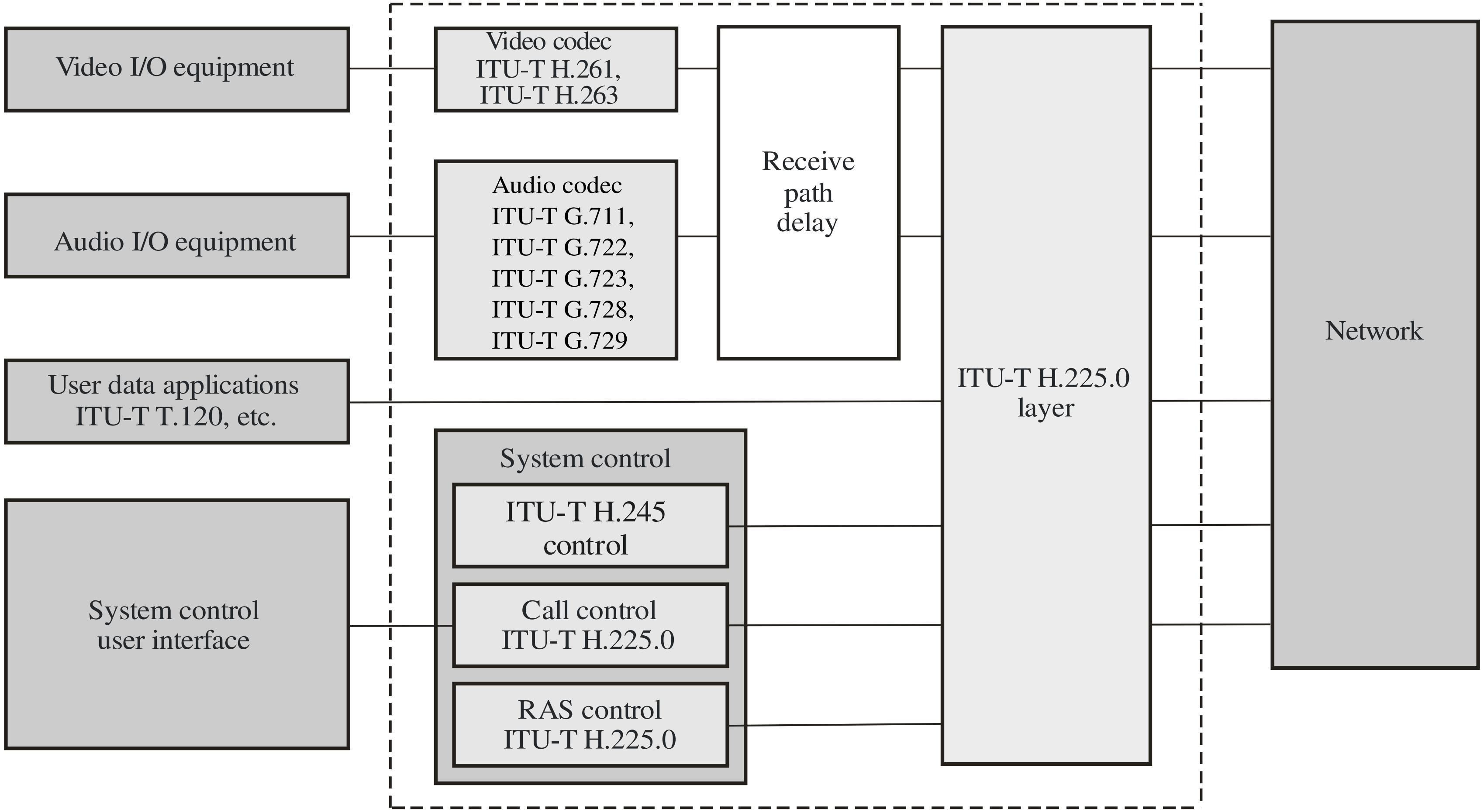
H.323 Elements

- Terminal
  - Endpoint on the network which provides for real-time, two-way communications with another H.323 terminal, Gateway, or Multipoint Control Unit.
  - This communication consists of control, indications, audio, video, and/or data between the two endpoints.
- Gateway (GW)
  - Endpoint on the network which provides for real-time, two-way communications between Terminals on the packet-based network and other Terminals on a switched circuit network or to another H.323 Gateway.
- Gatekeeper (GK)
  - Entity on the network that provides address translation and controls access to the network for H.323 terminals, Gateways and MCUs.
  - The Gatekeeper may also provide other services to the terminals, Gateways and MCUs such as bandwidth management and locating Gateways.



- Multipoint Control Unit (MCU)
  - Endpoint on the network which provides the capability for three or more terminals and Gateways to participate in a multipoint conference.
  - Consists of two parts: a mandatory Multipoint Controller (MC) and optional Multipoint Processors (MP).
  - Multipoint Controller (MC)
    - Entity on the network which provides for the control of three or more terminals participating in a multipoint conference.
  - Multipoint Processor (MP)
    - Entity on the network that provides centralized processing of audio, video and/or data streams in a multipoint conference.

H.323 Terminal Equipment



- H.225
  - Registration, Admission and Status (RAS), which is used between an H.323 endpoint and a Gatekeeper to provide address resolution and admission control services.
  - Call Signaling, which is used between any two H.323 entities in order to establish communication (based on Q.931/ Q.932).
- H.245
  - Control protocol for multimedia communication, which describes the messages and procedures used for capability exchange, opening and closing logical channels for audio, video and data, control and indications.

H.323 Gatekeeper

- Gatekeeper is optional.
  - When present, can provide a set of functionalities:
    - Routing of call signaling (better control, intelligent routing decisions, load balancing of gateways).
    - However, these messages can be sent directly between terminals.
- H.323 networks with IP/PSTN gateways should contain a gatekeeper to make address translation
- Mandatory functions:
  - Address translation, admission and bandwidth control, zone management.
- Optional functions
  - Call control signaling, call authorization and management.

H.323 Operation

- Obtain gatekeeper permission (H.225 RAS Admission Request).
- Press the number (call) (Q.931 Call Signaling).
- Tell the partners what languages it understands/talks (H.245 Capability Negotiation).
- Wait for the communication of its capabilities (H.245 Capability Negotiation).
- Inform what languages will be used during the conversation (H.245 Logical Channel Signaling).
- Start talking (and listening) (Data transfer with RTP/RTCP).
- Upon termination, say Bye (H.245 End Session).
- Disconnect (Q.931 Call Termination).
- Inform the gatekeeper that the call ended (H.225 RAS Disengage Request).

H.225 RAS Messages

- Gatekeeper discovery:
  - **Gatekeeper Request (GRQ), Gatekeeper Confirm (GCF)** and **Gatekeeper Reject (GRJ)**
  - If one gatekeeper answers positively, the endpoint should select which one to use.
- Endpoints registration:
  - **Registration Request (RRQ)** and **Unregistration Request (URQ)**
- Endpoints location:
  - **Location Request (LRQ), Location Confirm (LCF)** and **Location Reject (LRJ)**
  - Through the alias of another endpoint, it can obtain contact information of that endpoint.
- Admission to participate in a session:
  - **Admission Request (ARQ), Admission Confirmation (ACF) and Admission Reject (ARJ)**
- Change of bandwidth by an endpoint or gatekeeper
  - **Bandwidth Request (BRQ), Bandwidth Confirm (BCF)** and **Bandwidth Request (BRJ)**
- State information of an endpoint:
  - **Information Request (IRQ)** and **Information Request Response (IRR)**
- Session leave:
  - **Disengage Request (DRQ), Disengage Confirm (DCF)** and **Disengage Reject (DRJ)**
- Communication of available resources - gateways should inform gatekeepers about its capacities:
  - **Resource Available Indicate (RAI)** and **Resource Available Confirmation (RAC)**

H.225 Call Signaling Q.931 Messages

- Call establishment messages:
  - **Setup, Setup Acknowledge, Alerting**, **Call Proceeding**, **Connect**, **Connect Acknowledge**, and **Progress**.
- Call Clearing messages:
  - **Disconnect**, **Release,** and **Release Complete**.
- Call Information Phase messages:
  - **Resume**, **Resume Acknowledge**, **Resume Reject**, **Suspend**, **Suspend Acknowledge**, **Suspend Reject**, and **User Information.**
- Miscellaneous messages:
  - **Congestion Control**, **Information**, **Notify**, **Status**, and **Status Inquiry**.
- Q.932/H.450 messages:
  - **Facility**, **Hold**, **Hold Acknowledge**, **Hold Reject**, **Retrieve**, **Retrieve Acknowledge**, and **Retrieve Reject**.

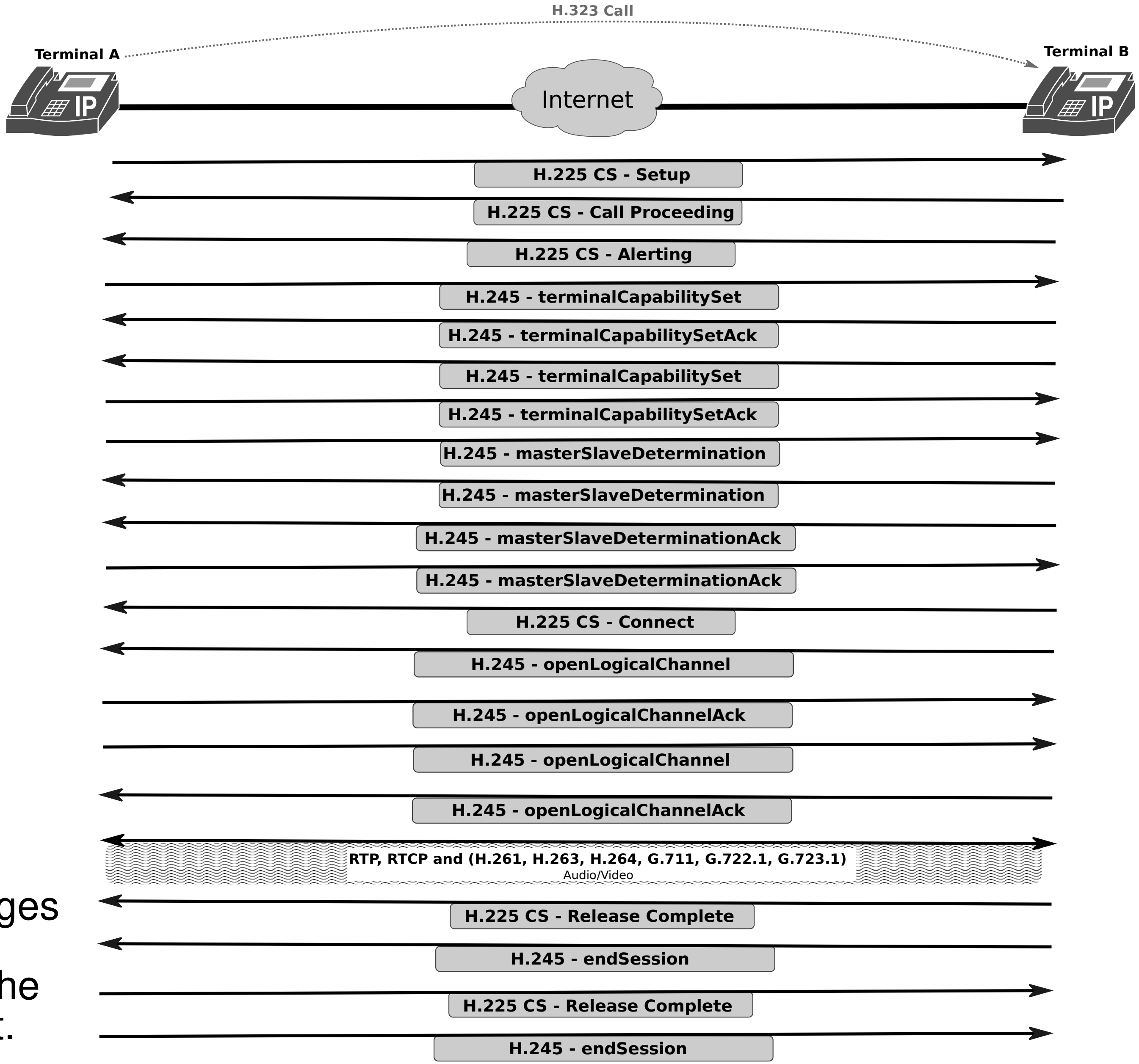
H.225 Call Signaling (most common)

- **Setup** - Establish a session between endpoints.
- **Call Proceeding** (optional) - answer to a setup indicating that it received the establishment process of the running session.
- **Alerting** - message sent by a callee to indicate that the user was already notified (corresponds to the phone ringing).
- **Progress** - optional message sent by a gateway to indicate that the session is in progress.
- **Connect** - message sent by a callee that indicates session acceptation.
- **Release Complete** - message sent by an endpoint to terminate a session.
- **Facility** - message sent by an endpoint to another one to inform where to redirect the session (other information can be sent)
- **Notify** - optional message used by any H.323 entity to send information to another one.
- **Status Inquiry** - message used by an endpoint during a session lifetime to ask another one about its status.
- **Status** - message used to answer to a status inquiry message.

H.245 Control Messages

- Used after the exchange of **Setup** and **Connect** messages to open an H.245 control channel.
- Capacities negotiation (supported formats for sending and reception):
  - **terminalCapabilitySet, terminalCapabilitySetAck, terminalCapabilitySetReject**
- Master/slave determination to solve conflicts that may appear during a session lifetime:
  - **masterSlaveDetermination, masterSlaveDeterminationAck, masterSlaveDeterminationReject**
- Opening of logical channels for several flows:
  - **openLogicalChannel, openLogicalChannelAck, openLogicalChannelConfirm, openLogicalChannelReject**
- Closing of logical channels:
  - **closeLogicalChannel, closeLogicalChannelAck, requestChannelClose, requestLogicalChannelAck, requestLogicalChannelReject**
- When all logical channels are closed, the session can be terminated:
  - **endSession**

H.323 Direct Call



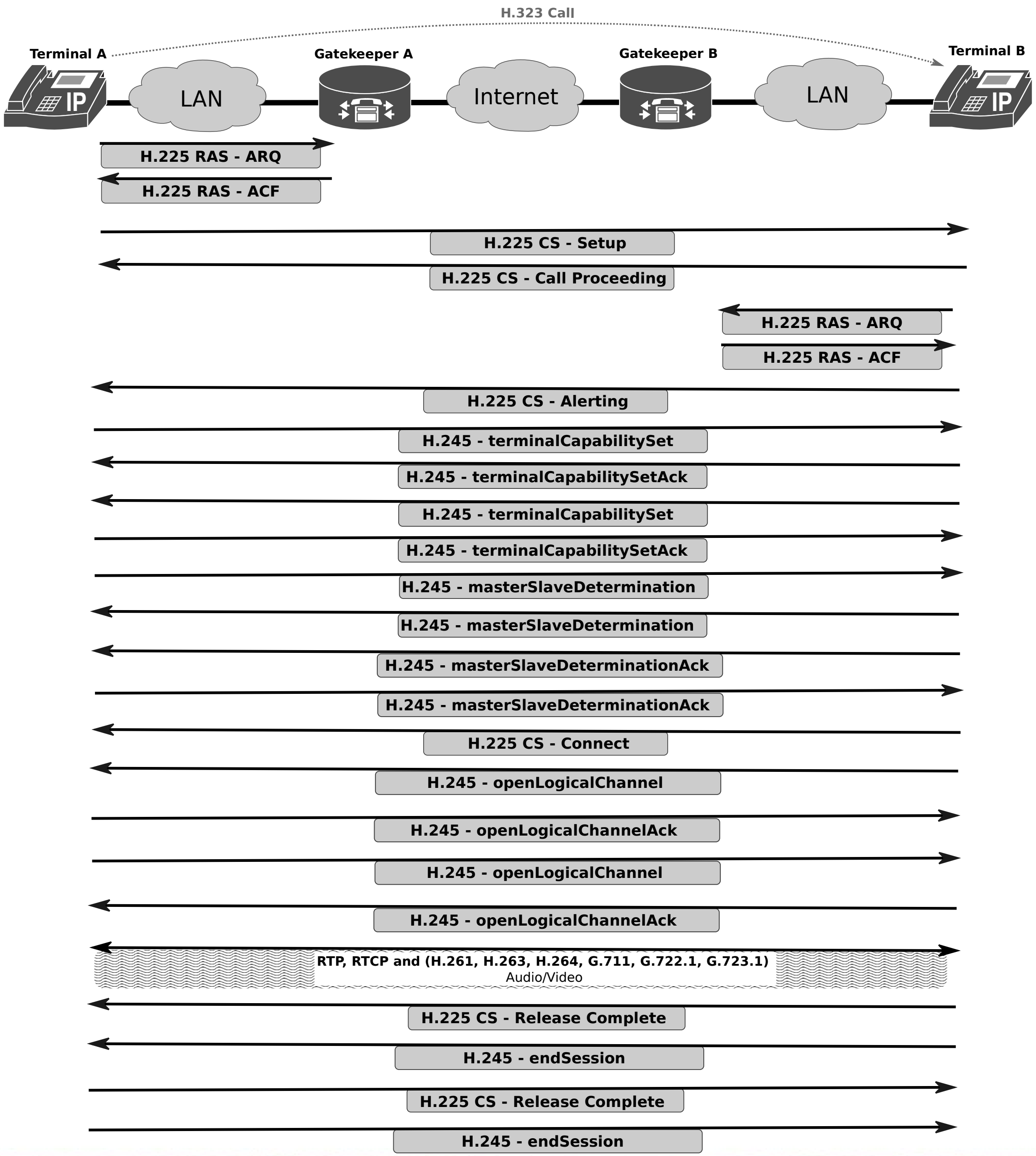
- Multiple messages may be transported in the same IP packet.

H.323 Direct Call

Source	Destination	Protocol	Length	Info
192.168.56.1	192.168.56.101	H.225.0/H	155	masterSlaveDetermination terminalCapabilitySet CS: setup OpenLogicalChannel
192.168.56.101	192.168.56.1	H.225.0	181	CS: callProceeding
192.168.56.101	192.168.56.1	H.225.0/H	460	masterSlaveDeterminationAck terminalCapabilitySetAck terminalCapabilitySet CS: empty
192.168.56.101	192.168.56.1	H.225.0	181	CS: alerting
192.168.56.1	192.168.56.101	H.225.0/H	109	masterSlaveDeterminationAck terminalCapabilitySetAck CS: empty
192.168.56.101	192.168.56.1	H.225.0/H	106	roundTripDelayRequest CS: empty
192.168.56.1	192.168.56.101	H.225.0/H	106	roundTripDelayResponse CS: empty
192.168.56.101	192.168.56.1	H.225.0	360	CS: connect OpenLogicalChannel
192.168.56.101	192.168.56.1	H.225.0/H	131	endSessionCommand CS: releaseComplete
192.168.56.1	192.168.56.101	H.225.0/H	131	endSessionCommand CS: releaseComplete

Source	Destination	Protocol	Length	Info
192.168.56.101	192.168.56.1	H.261	1023	H.261 message
192.168.56.101	192.168.56.1	H.261	1021	H.261 message
192.168.56.101	192.168.56.1	H.261	353	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48411, Time=0, Mark
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48412, Time=320
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48413, Time=640
192.168.56.101	192.168.56.1	H.261	336	H.261 message
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6053, Time=0, Mark
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6054, Time=160
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6055, Time=320
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48414, Time=960
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6056, Time=480
192.168.56.101	192.168.56.1	H.261	386	H.261 message
192.168.56.1	192.168.56.101	H.261	1023	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48415, Time=1280
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6057, Time=640
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6058, Time=800
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48416, Time=1600
192.168.56.101	192.168.56.1	H.261	346	H.261 message
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6059, Time=960
192.168.56.1	192.168.56.101	H.261	1021	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48417, Time=1920
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6060, Time=1120

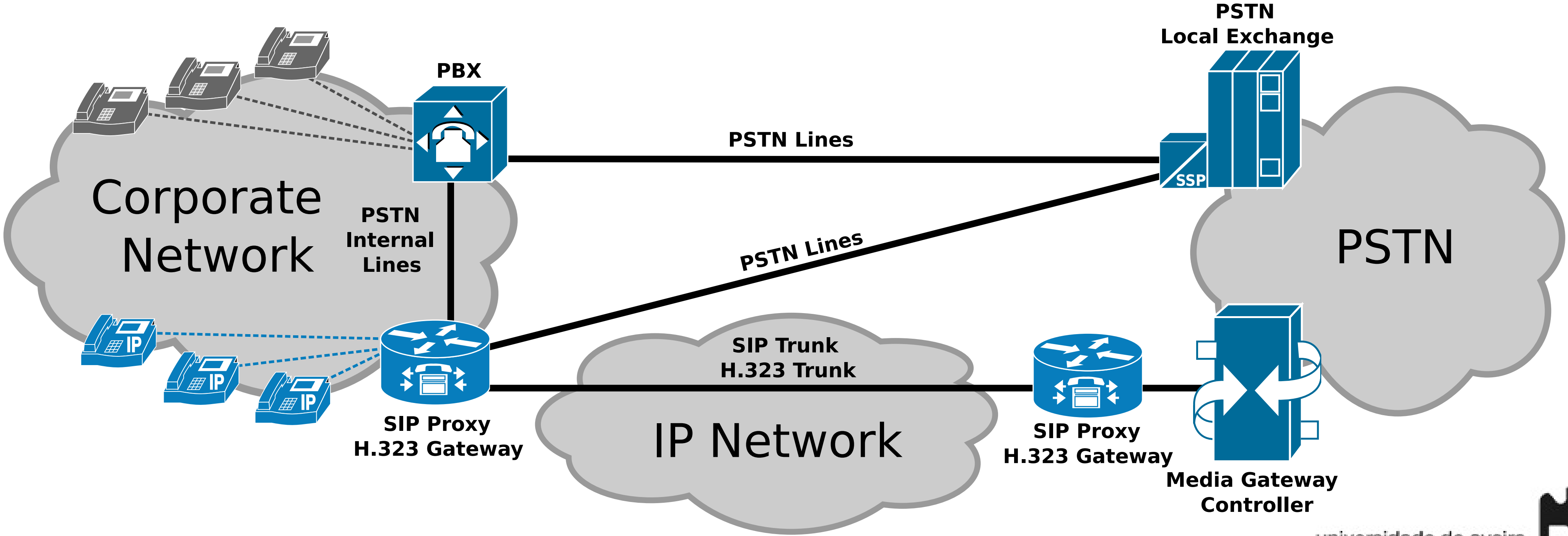
H.323 Call (with Gatekeepers)



- Multiple messages may be transported in the same IP packet.

VoIP and PSTN Connectivity

- SIP proxy or H.323 gateway.
  - With PSTN interface (to ISP or local PBX).
    - Requires multiple PSTN Lines.
    - Not scalable.
  - With SIP or H.323 trunk to remote SIP proxy or H.323 Gateway.
    - Remote proxy/gateway interfaces with PSTN network.
    - Remote proxy/gateway owned by PSTN ISP or by a third-party entity.
    - Usually TCP/IP transport with a TLS security layer.
    - Scalable!



VoIP and PSTN Interoperability in Large Scalable Scenarios

- Requires an application programming interface and a corresponding protocol for controlling VoIP Gateways from external call control elements.
- Signaling must be inter-operable between PSTN and VoIP.
- Protocols:
  - Media Gateway Controller Protocol (MGCP) - RFC 2705
  - MGCP evolution/successor → H.248/Megaco (RFC 3015) → H.248.1/ Gateway Control Protocol (RFC 3525)
    - These are control plane signaling only.
  - SIGTRAN (Signaling Transport) is the standard telephony protocol used to transport Signaling System 7 (SS7) signals over the Internet.
    - Stream Control Transmission Protocol (SCTP) – RFC 3286
      - Is an IP transport designed for transporting signaling information over an IP network.
      - Reliable transport protocol with support for framing of individual message boundaries.

MGCP/H.248 Elements

- Media Gateway Controller (MGC)
  - Controls the parts of the call state that pertain to connection control for media channels in a MG.
- Media Gateway (MG)
  - Converts media provided in one type of network to the format required in another type of network.
  - MG could terminate bearer channels from a switched circuit network (e.g., DS0s) and media streams from a packet network (e.g., RTP streams in an IP network).
- Signaling Gateway (SG)
  - Responsible for transferring signaling messages (e.g., SS7 messages) to different protocols and transports.
    - Signaling Transport (SIGTRAN)
    - e.g., SS7 to SIGTRAN (SCTP/IP).

MGCP/H.248 Scenario