



Special Module on Media Processing and Communication

Multimedia Communication Fundamentals

**Dayalbagh Educational Institute
(DEI)
Dayalbagh Agra**

PHM 961

**Indian Institute of Technology Delhi
(IITD)
New Delhi**

SIV 864



Overview

- Review of last class
- Protocol stack for multimedia services
- Real-time transport protocol (RTP)
- RTP control protocol (RTCP)
- Real-time streaming protocol (RTSP)



Session Initiation Protocol (SIP)

Instead of H.323, also the simpler, Internet-oriented SIP can be used:

- Defined by IETF
- SIP long-term vision

All telephone calls and video conference calls take place over the Internet

People are identified by names or e-mail addresses, rather than by phone numbers. You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using

- SIP is an application layer signaling protocol that defines initiation, modification and termination of interactive multimedia communication sessions between multiple users

Call setup: Agree on media type and encoding, Maps logical address



Overview

SIP and sipd

- Address based on email (alice@home.com)

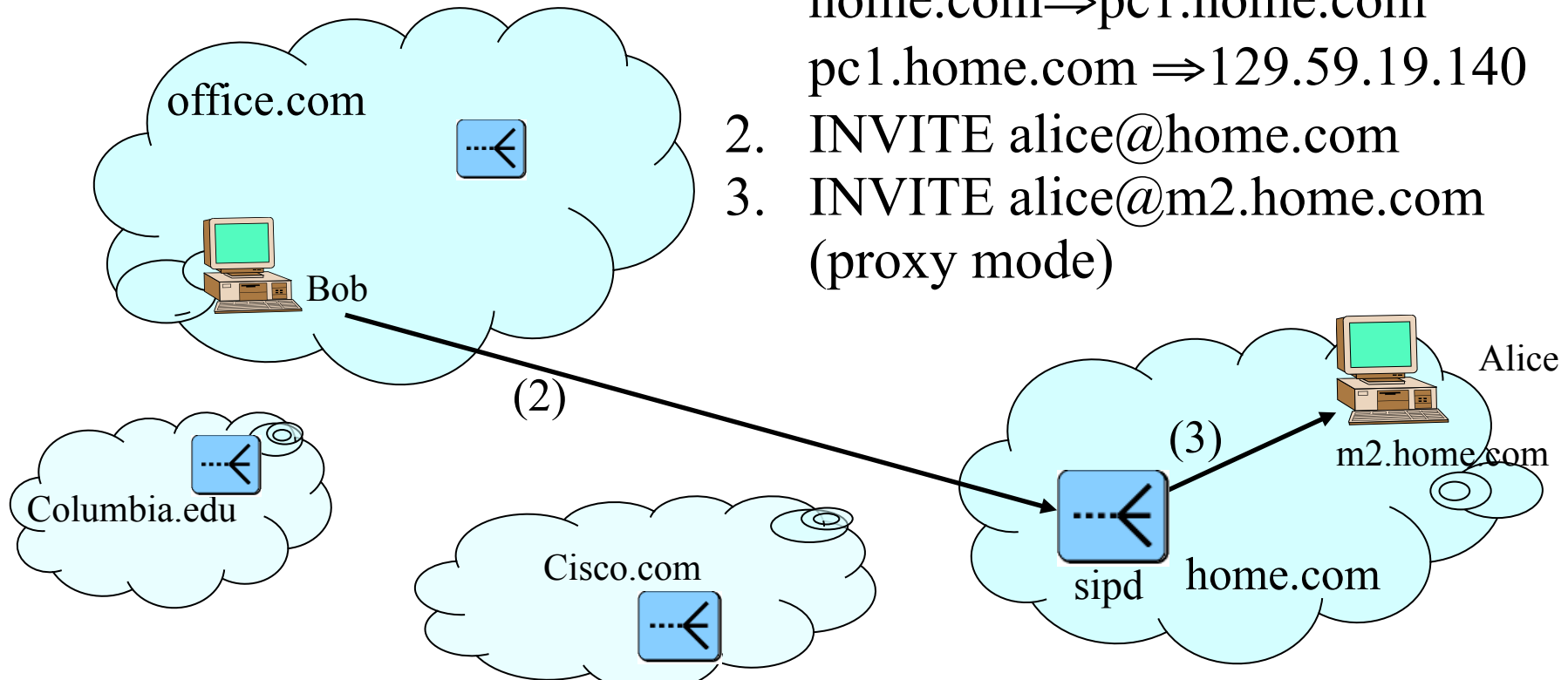
1. DNS

home.com \Rightarrow pc1.home.com

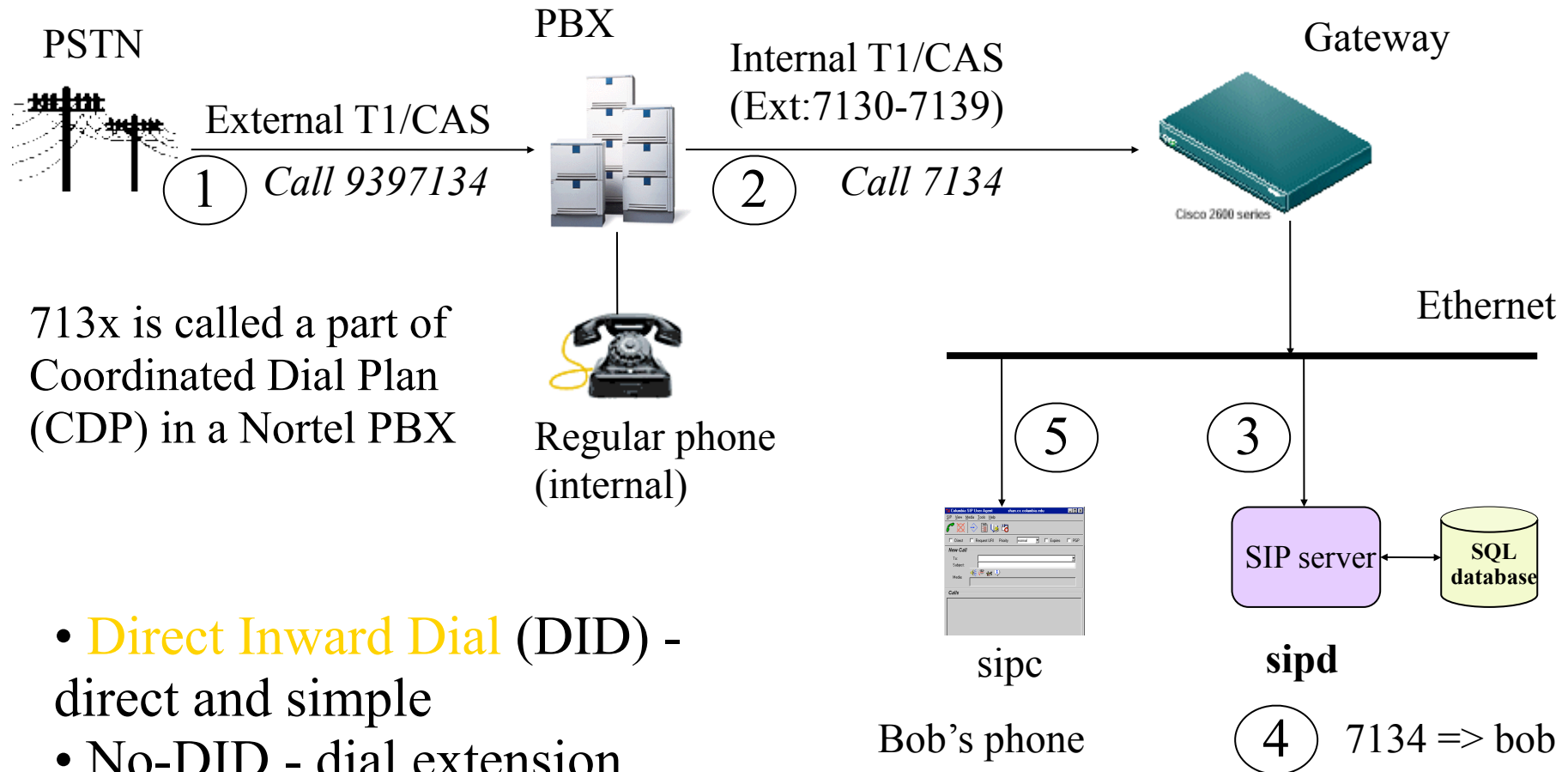
pc1.home.com \Rightarrow 129.59.19.140

2. INVITE alice@home.com

3. INVITE alice@m2.home.com
(proxy mode)

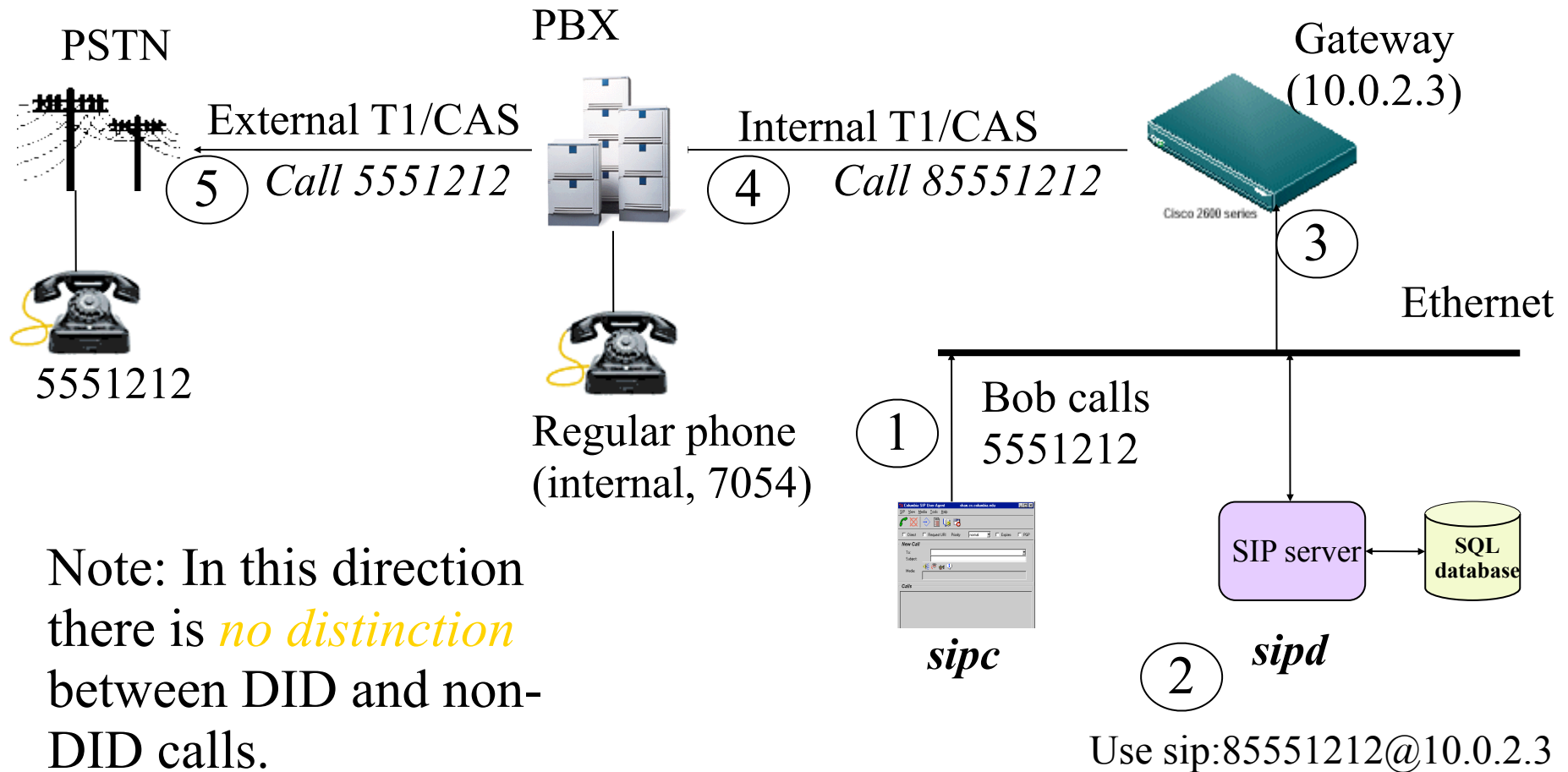


PSTN to IP Call



- **Direct Inward Dial (DID)** - direct and simple
- **No-DID** - dial extension, supports more users

IP to PSTN Call





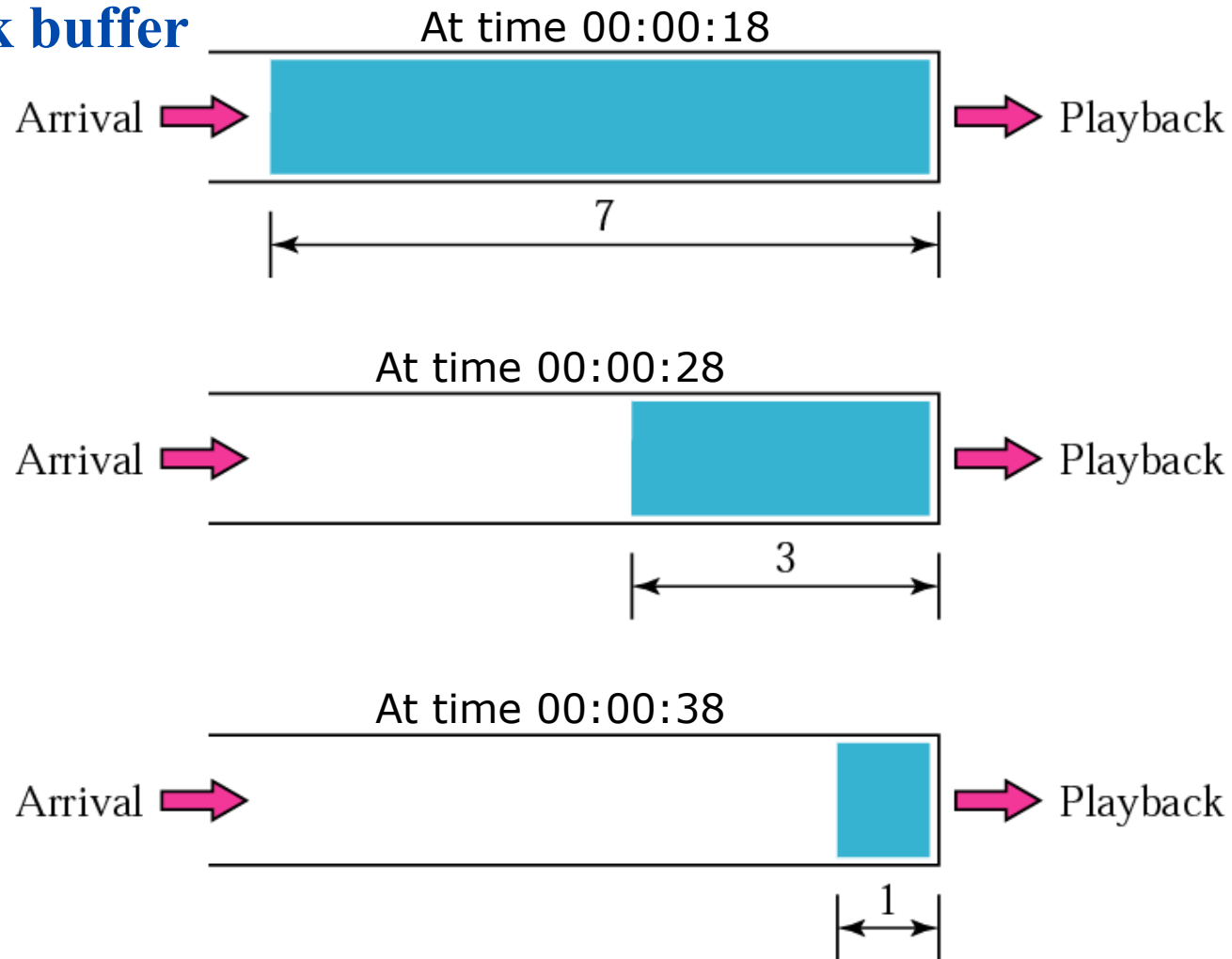
RTP: Streaming performance requirements

- Sequencing
 - to report PDU loss
 - to report PDU reordering
 - to perform out-of-order decoding
- Time stamping and Buffering
 - for play out
 - for jitter and delay calculation
- Payload type identification
 - for media interpretation
- Error concealment –covers up errors from lost PDU by using redundancy in most-adjacent-frame
- Quality of Service (QoS) feedback – from receiver to sender for operation adjustment
- Rate control –sender reduces sending rate adaptively to network congestion



Jitter (contd.)

Playback buffer





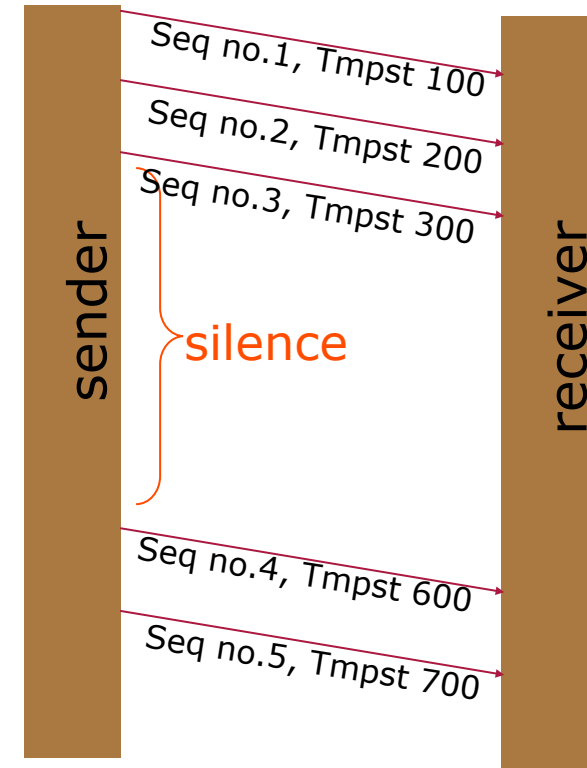
How does Sequence number and Timestamp help ?

Audio silence example:

- Consider audio data
 - What should the sender do during silence?
 - Not send anything
 - Why might this cause problems?
 - Receiver cannot distinguish between loss and silence

Solution:

- After receiving no PDUs for a while, next PDU received at the receiver will reflect a big jump in timestamp, but have the correct next seq. no. Thus, receiver knows what happened.





Streaming performance requirements

- Sequencing
 - to report PDU loss
 - to report PDU reordering
 - to perform out-of-order decoding
- Time stamping and Buffering
 - for play out
 - for jitter and delay calculation
- Payload type identification
 - for media interpretation
- Error concealment –covers up errors from lost PDU by using redundancy in most-adjacent-frame
- Quality of Service (QoS) feedback – from receiver to sender for operation adjustment
- Rate control –sender reduces sending rate adaptively to network congestion



Support from transport layers

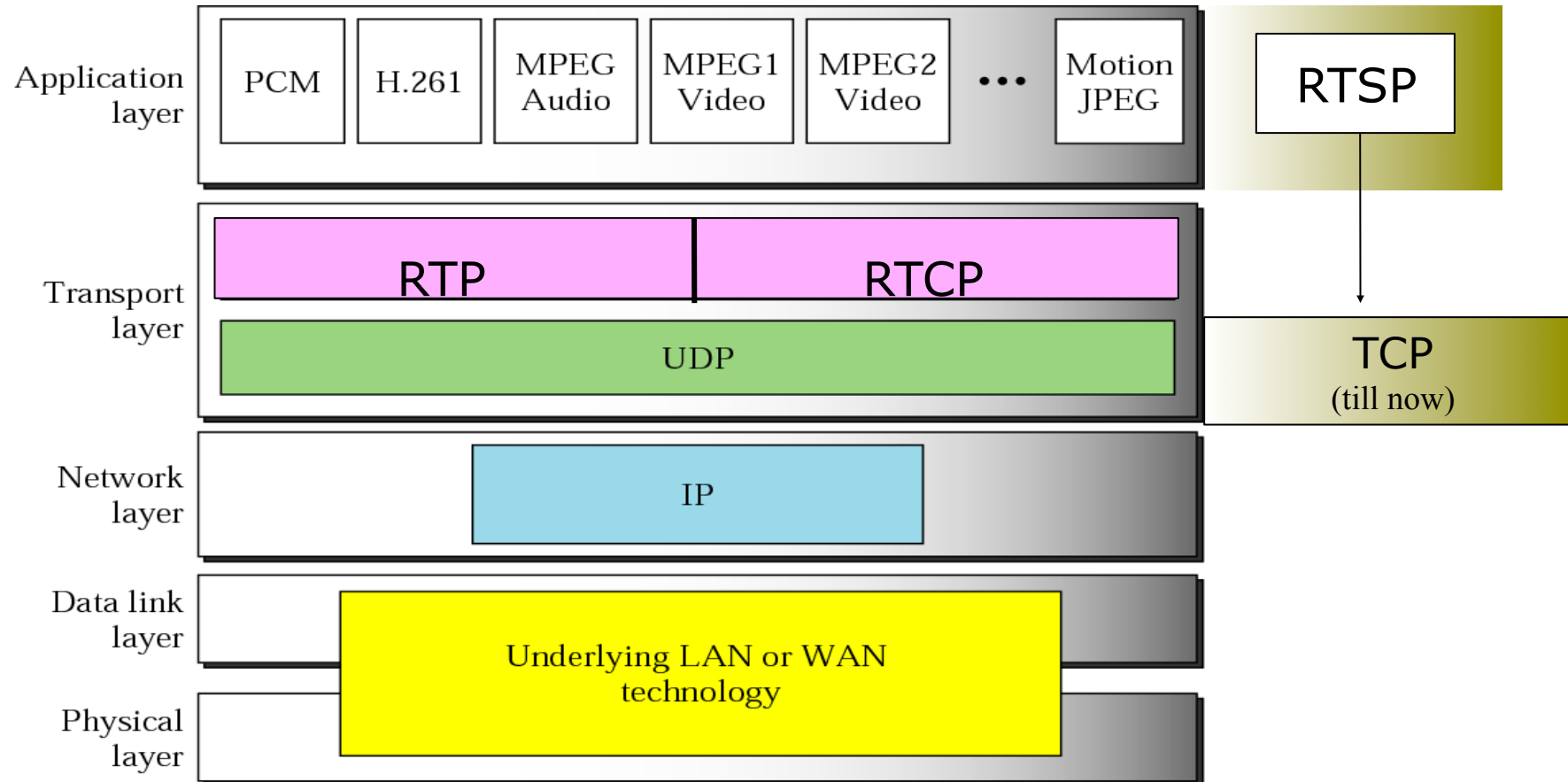
➤ TCP is not used because:

- TCP does retransmissions → unbounded delays
- No provision for time stamping
- TCP does not support multicast
- TCP congestion control (slow-start) unsuitable for real-time transport

RTP + UDP usually used for multimedia services



Protocol stack for multimedia services

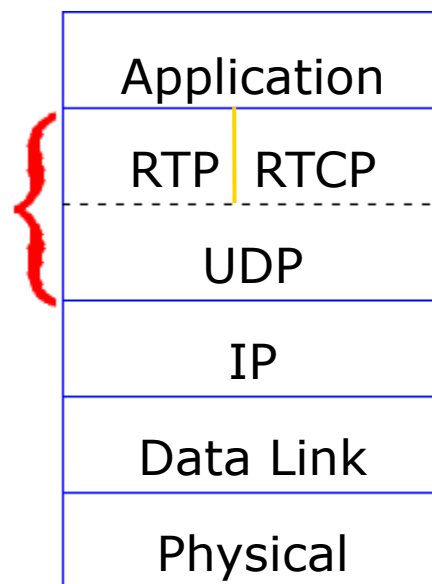




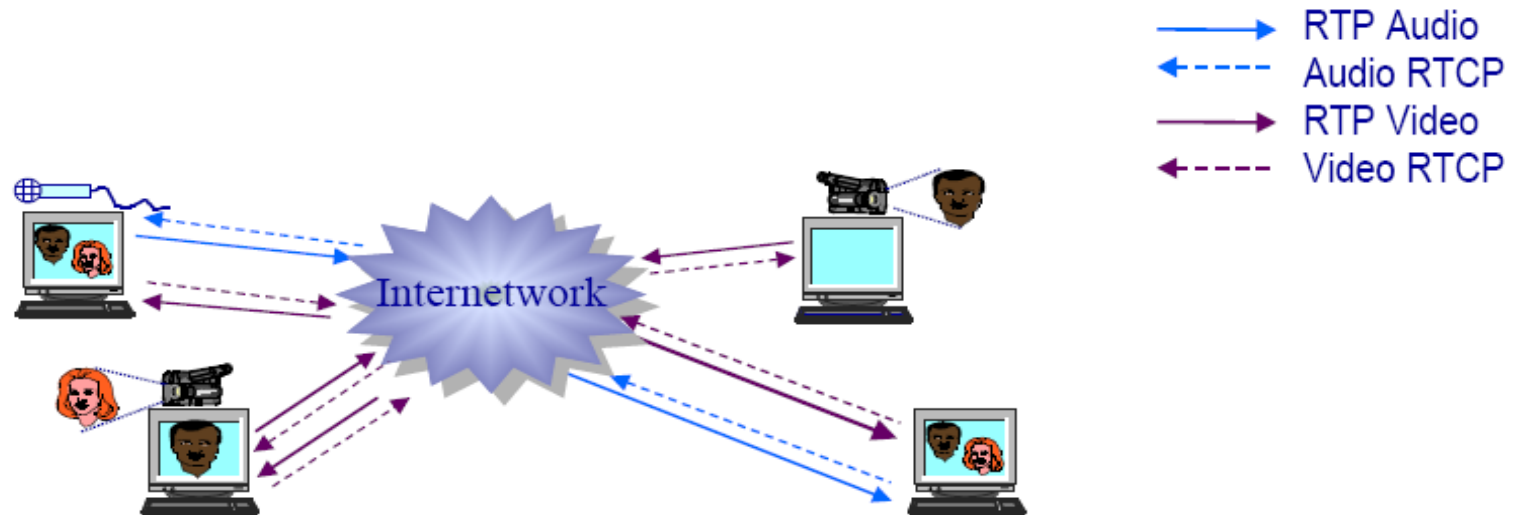
RTP: Introduction

- Provides end-to-end transport functions for real-time applications
- Supports different payload types
- All RTP and RTCP PDUs are sent to same multicast group (by all participants)
- All RTP PDUs sent to an even-numbered UDP port,
- All RTCP PDUs sent to UDP port $2p+1$
- Does NOT provide timely delivery or other QoS guarantees
- Relies on other protocols like RTCP and lower layers
- Does NOT assume the underlying network is reliable and delivers PDUs in sequence
- Uses sequence number

Transport layer



RTP Session

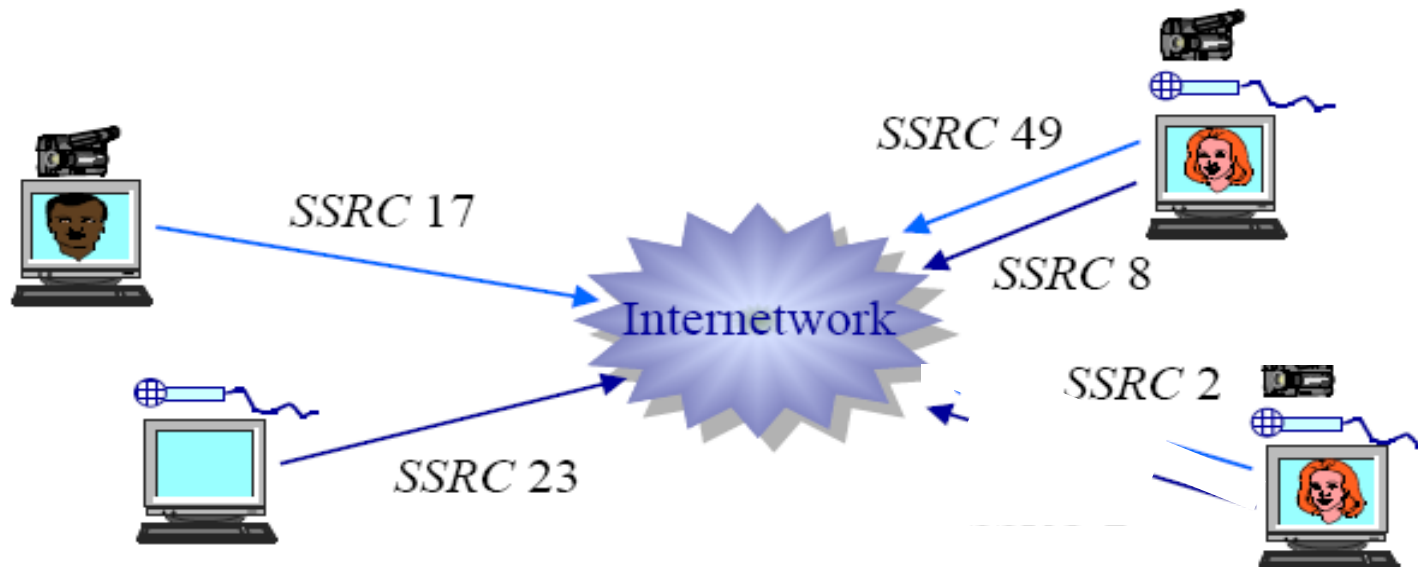


□ RTP *session* is sending and receiving of RTP data by a group of participants

➤ For each participant, a session is a pair of transport addresses used to communicate with the group

□ If multiple media types are communicated by the group, the transmission of each medium constitutes a session.

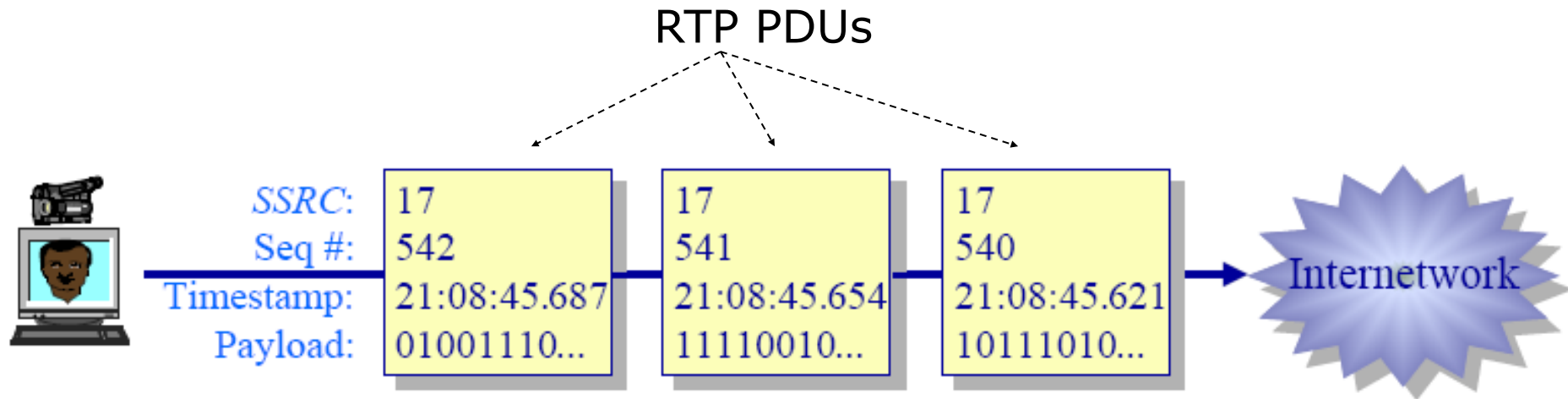
RTP Synchronization Source



- ❑ *synchronization source* - each source of RTP PDUs
- ❑ Identified by a unique, randomly chosen 32-bit ID (*the SSRC*)
- ❑ A host generating multiple streams within a single RTP must use a different SSRC per stream



RTP Basics of Data Transmission





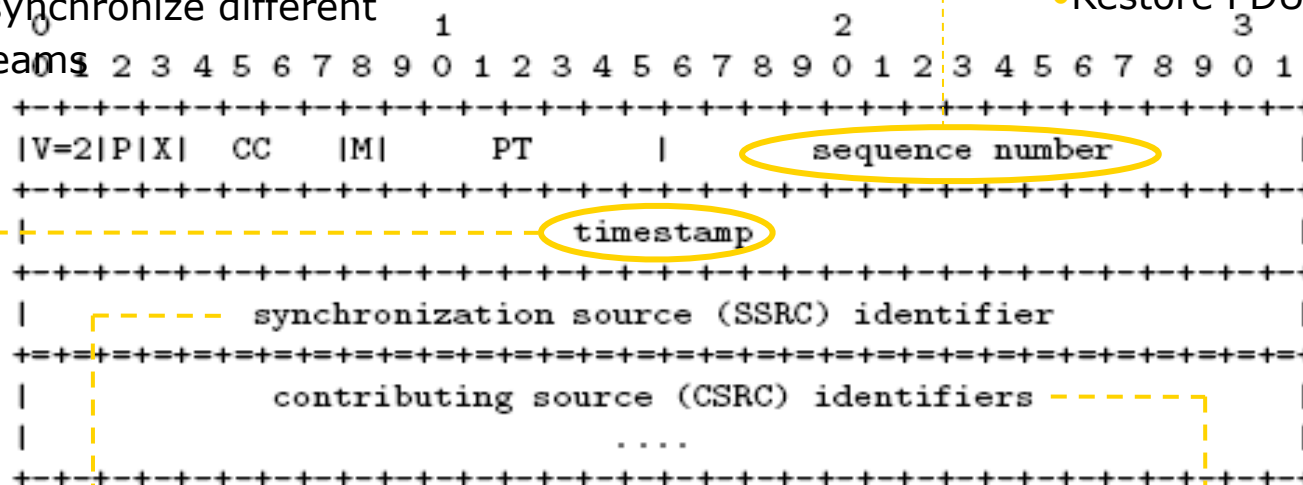
RTP PDU Header

Sampling instant of first data octet Payload type

- multiple PDUs can have same timestamp
- not necessarily monotonic
- used to synchronize different media streams

Incremented by one for each RTP PDU:

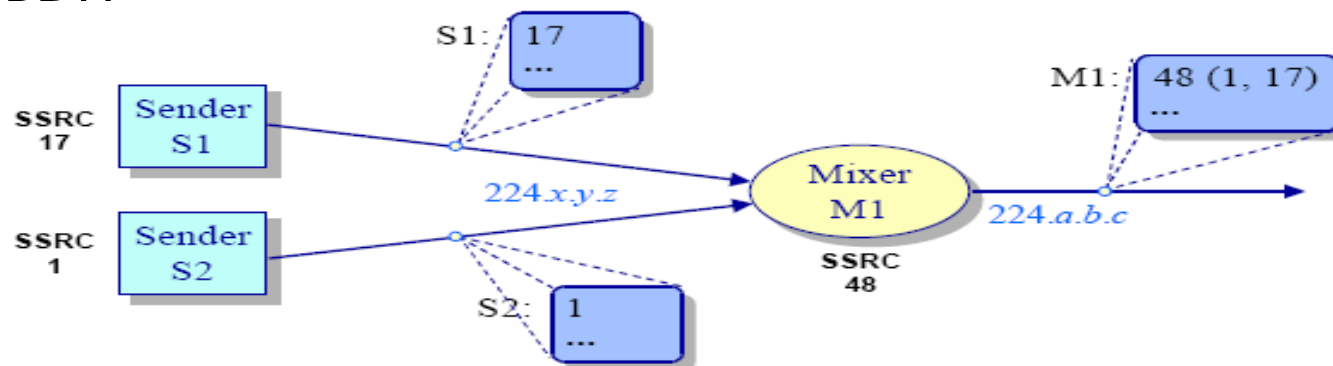
- PDU loss detection
- Restore PDU sequence



Identifies synchronization source

Identifies contributing sources
(used by mixers)

RTP *mixer* - an intermediate system that receives & combines RTP PDUs of one or more RTP sessions into a new RTP

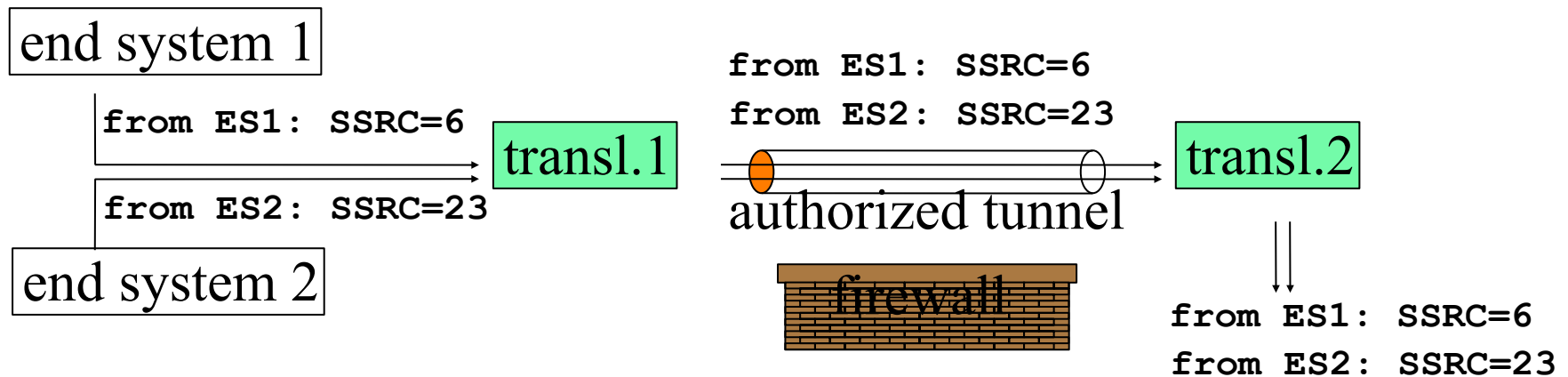


- Stream may be transcoded, special effects may be performed.
- A mixer will typically have to define synchronization relationships between streams. Thus...
 - Sources that are mixed together become **contributing sources (CSRC)**
 - Mixer itself appears as a new source having a new **SSRC**



Translator

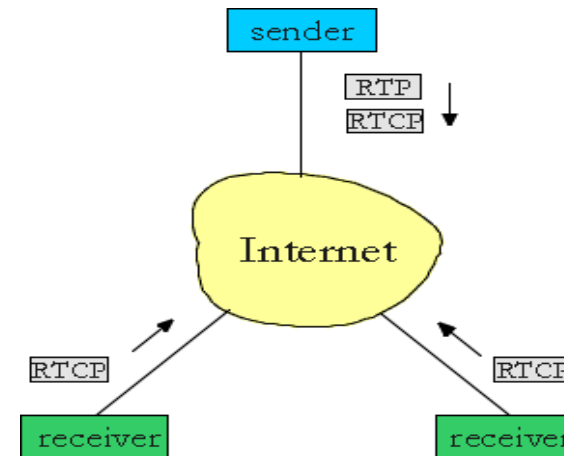
- An intermediate system that...
 - Connects two or more networks
 - ❖ Multicasting through a firewall
 - ❖ Modifies stream encoding, changing the stream's timing
 - ❖ Transparent to participants
 - ❖ SSRC's remain intact





RTP Control Protocol (RTCP)

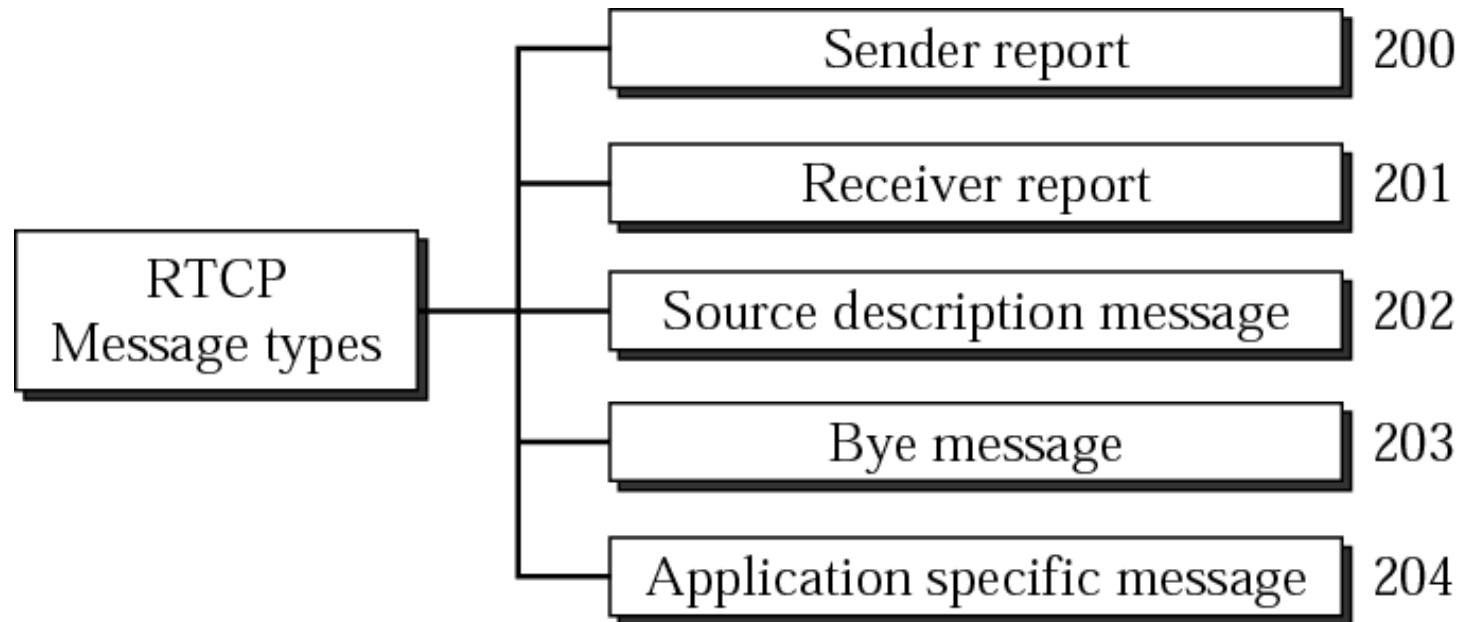
- r RTCP specifies report PDUs exchanged between sources and destinations of multimedia information
- r receiver reception report
- r sender report
- r source description report



- r Reports contain statistics such as the number of RTP-PDUs sent, number of RTP-PDUs lost, inter-arrival jitter
- r Used by application to modify sender transmission rates and for diagnostics purposes



RTCP message types

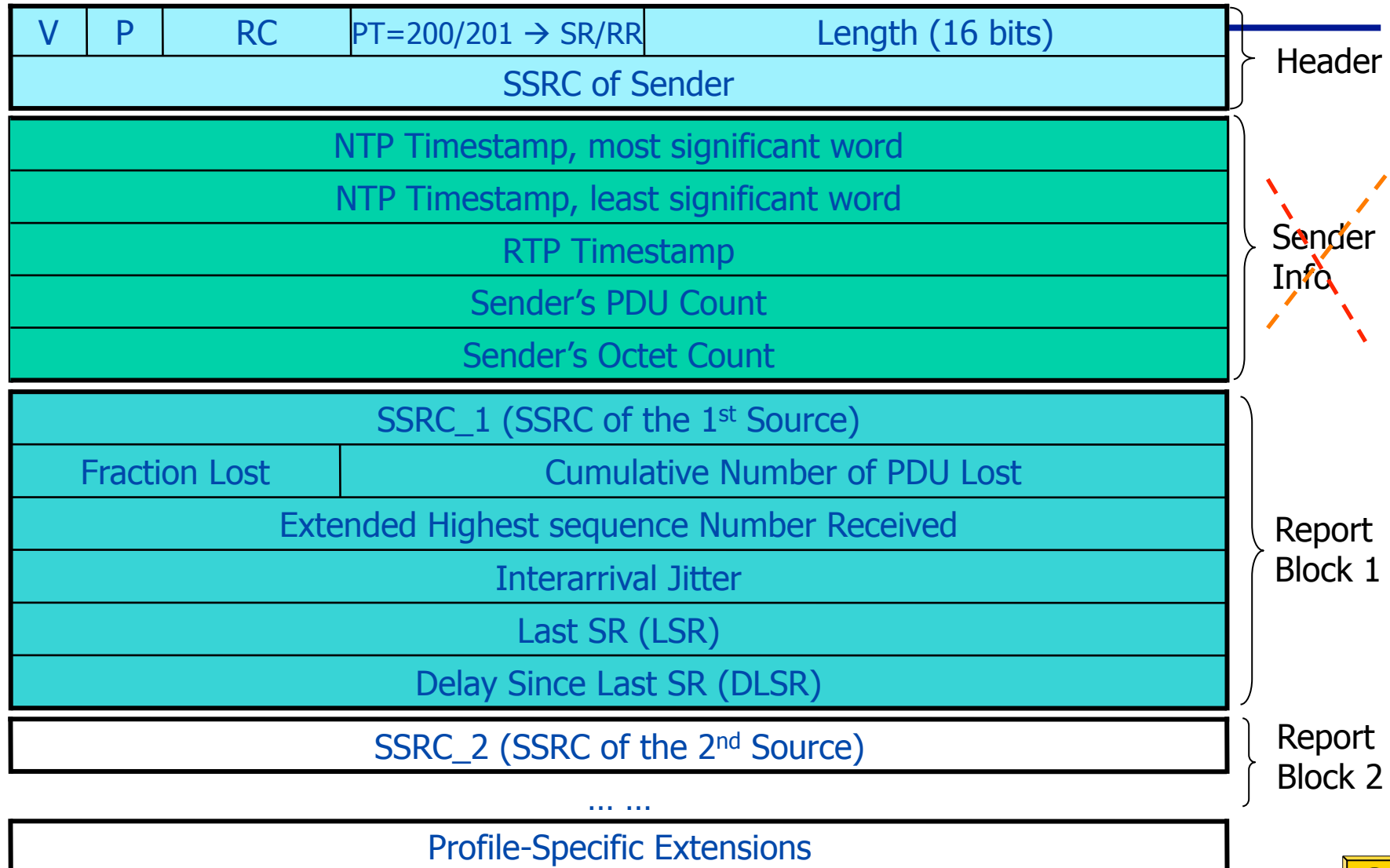


Typically, several RTCP PDUs of different types are transmitted in a single UDP PDU





Sender/Receiver report PDUs





Etherereal capture for RTP-PDU



final_rtp - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
224	15.796849	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
225	15.800481	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
226	15.805614	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
227	15.807143	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
228	15.810710	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
229	15.815526	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
230	15.817036	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
231	15.820637	192.168.0.1	239.255.255.25	SSDP	NOTIFY * HTTP/1.1
232	19.504880	192.168.0.101	192.168.0.103	RTCP	Receiver Report
233	21.115826	192.168.0.103	192.168.0.101	RTCP	Receiver Report
234	24.585085	192.168.0.101	192.168.0.103	RTCP	Receiver Report
235	24.913280	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51425, Time=148008,
236	24.937758	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51426, Time=148248
237	24.962184	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51427, Time=148488
238	24.986585	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51428, Time=148728
239	25.022778	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51429, Time=148968
240	25.038384	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51430, Time=149208
241	25.060818	192.168.0.103	192.168.0.101	G.723	payload type=ITU-T G.723, SSRC=3879416967, Seq=51431, Time=149448

Frame 235 (78 bytes on wire, 78 bytes captured)

- Ethernet II, Src: 192.168.0.103 (00:0e:35:86:7b:80), Dst: 192.168.0.101 (00:30:bd:2c:dc:d6)
- Internet Protocol, src: 192.168.0.103 (192.168.0.103), dst: 192.168.0.101 (192.168.0.101)
- User Datagram Protocol, Src Port: 49608 (49608), Dst Port: 49608 (49608)
- Real-Time Transport Protocol
 - [Stream setup by H245 (frame 49)]
 - 10.. = Version: RFC 1889 Version (2)
 - ..0. = Padding: False
 - ...0 = Extension: False
 - 0000 = Contributing source identifiers count: 0
 - 1... = Marker: True
 - Payload type: ITU-T G.723 (4)
 - Sequence number: 51425
 - Timestamp: 148008
 - Synchronization source identifier: 3879416967
 - G.723
 -0 = RATEFLAG_B0: High rate(6.3kb/s)
 -0. = VADFLAG_B0: speech
 - 1111 01.. = LPC_B5...LPC_B0: 0x3d

Basic header

File: "C:\Documents and Settings\adr P: 1335 D: 1335 M: 0



Ethereal capture for RTCP-PDU



final_rtp - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
859	44.201738	192.168.0.101	192.168.0.103	G.723	Payload type=11001, G.723, SSRC=3860006015, Seq=8704, Time=302004
860	44.227289	192.168.0.103	192.168.0.101	RTCP	Sender Report

Real-time Transport Control Protocol

- [Stream setup by H245 (frame 49)]
 - 10.. = Version: RFC 1889 version (2)
 - ..0. = Padding: False
 - ...0 0001 = Reception report count: 1
 - Packet type: Sender Report (200)
 - Length: 12
 - Sender SSRC: 3879416967
 - Timestamp, MSW: 482
 - Timestamp, LSW: 1212153856
 - RTP timestamp: 302928
 - Sender's packet count: 283
 - Sender's octet count: 6792
- Source 1
 - Identifier: 3860006015
 - SSRC contents
 - Fraction lost: 1 / 256
 - Cumulative number of packets lost: 3
 - Extended highest sequence number received: 8704
 - Sequence number cycles count: 0
 - Highest sequence number received: 8704
 - Interarrival jitter: 7
 - Last SR timestamp: 3842553664
 - Delay since last SR timestamp: 122368
- Real-time Transport Control Protocol
- [Stream setup by H245 (frame 49)]
 - 10.. = Version: RFC 1889 version (2)
 - ..0. = Padding: False
 - ...0 0001 = Source count: 1
 - Packet type: Source description (202)
 - Length: 4
 - Chunk 1, SSRC/CSRC 3879416967
 - Identifier: 3879416967
 - SDES items
 - Type: CNAME (user and domain) (1)
 - Length: 6
 - Text: SADHAK
 - Type: END (0)

P: 1335 D: 1335 M: 0

header of SR report

sender info

receiver report block

SDES items

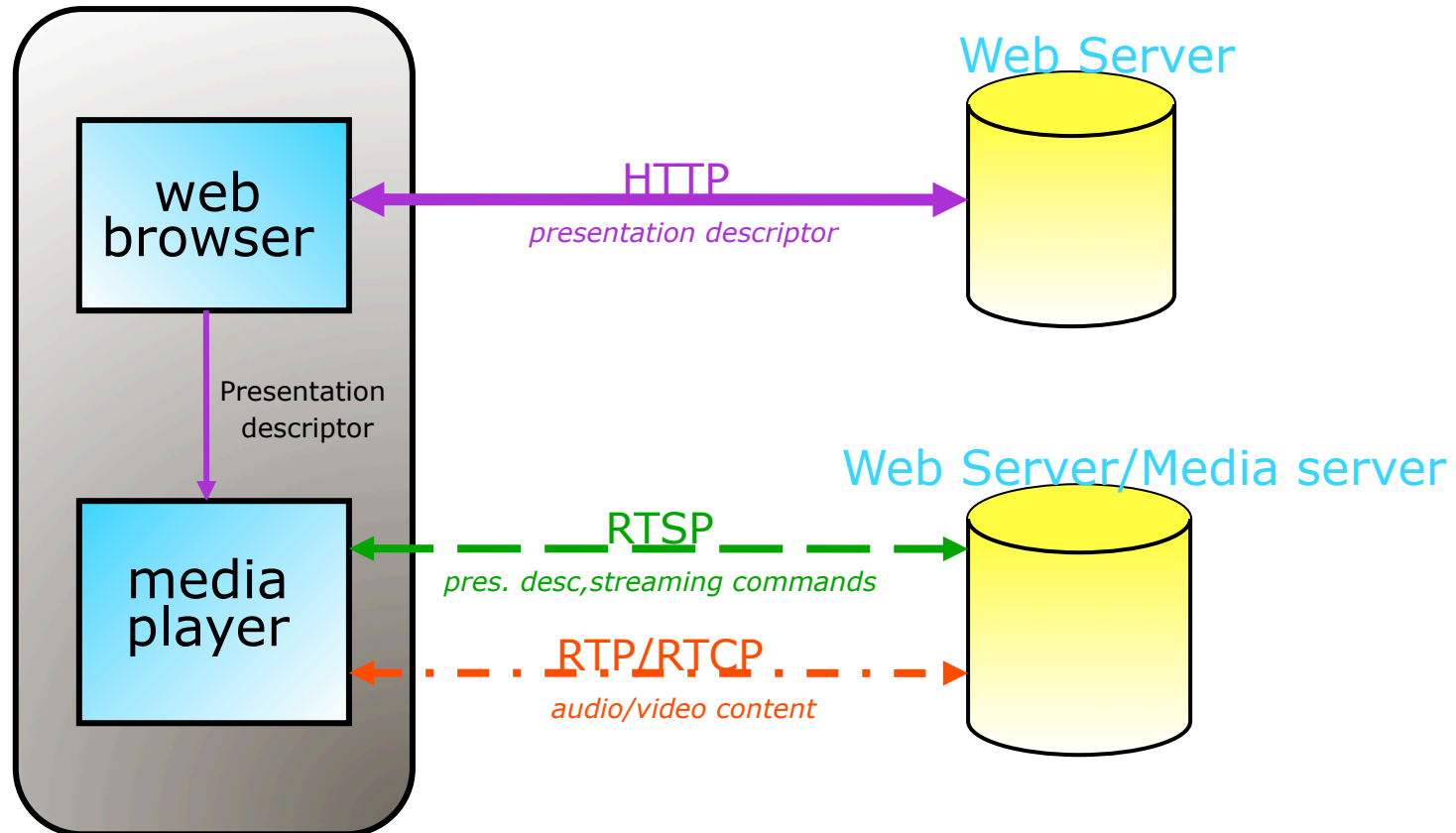


Real-Time Streaming Protocol (RTSP)

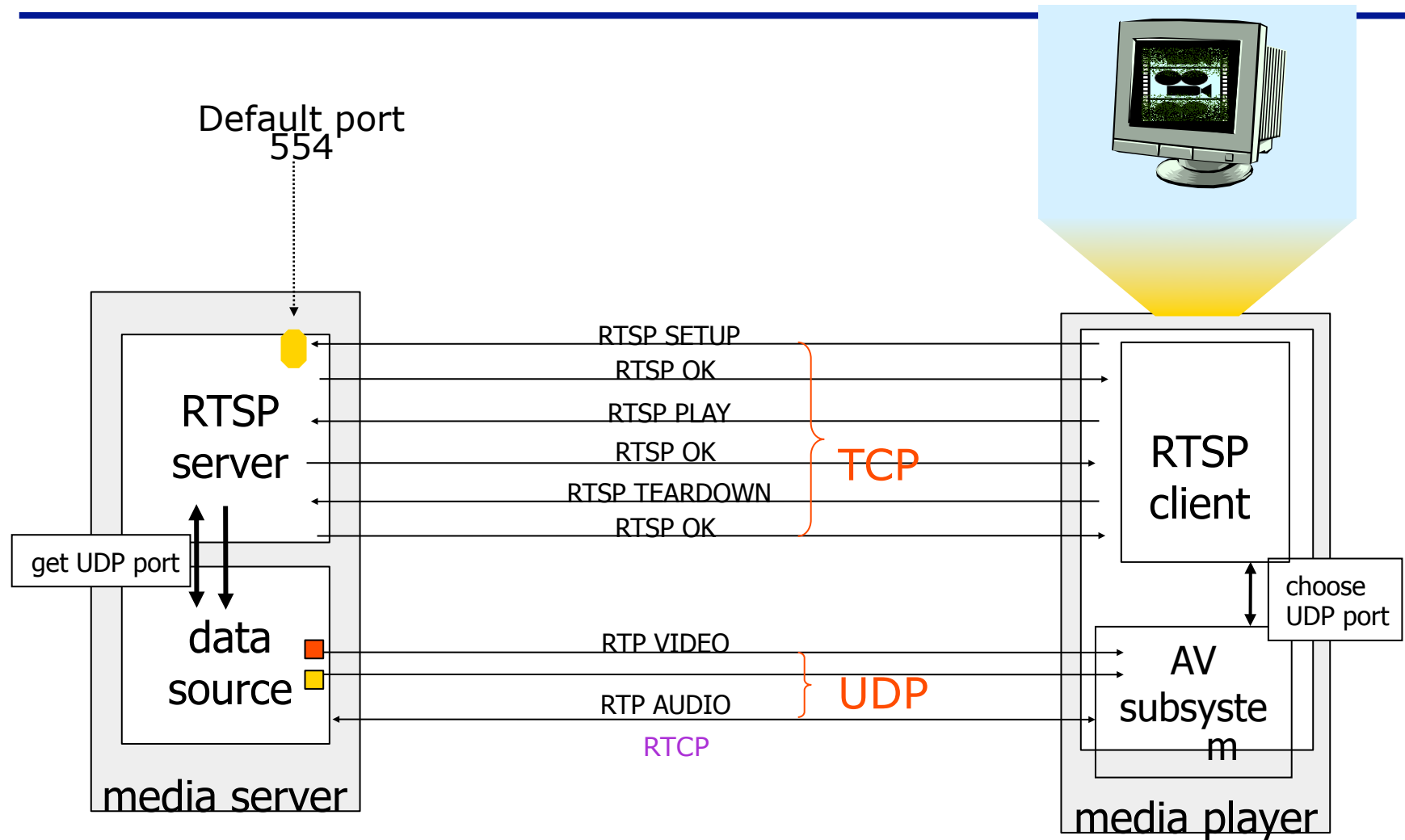
- Application layer protocol (default port 554)
- *Usually* runs on RTP for stream & TCP for control
- Provides the control channel
- Uses out-of-band signaling
- Usable for Live broadcasts / multicast

Also known as “Network remote control” for multi-media servers.

RTSP Overview

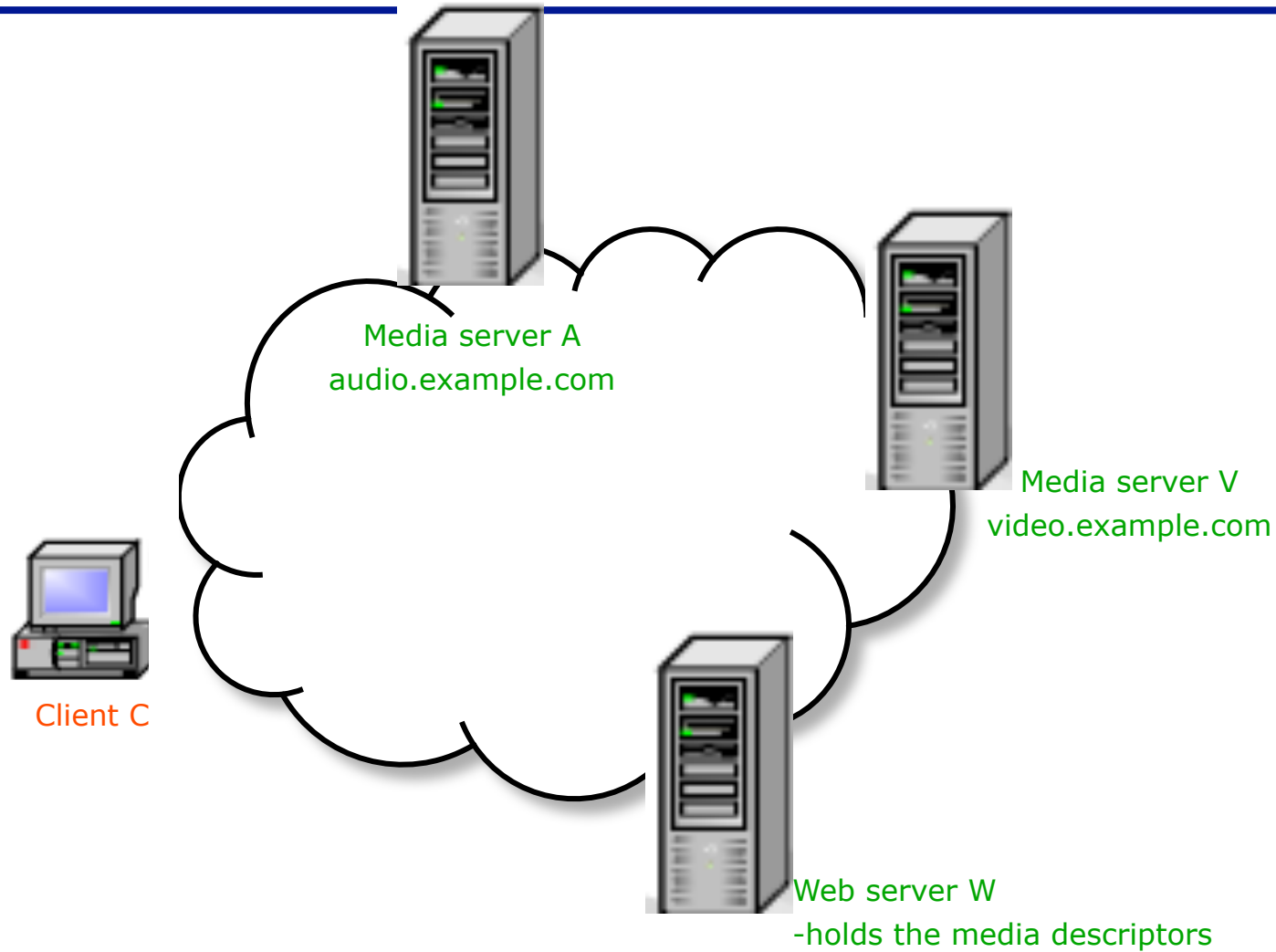


RTSP Session





Example: Media on demand (Unicast)





RTSP Message sequence

C -> W : GET/Twister.sdp HTTP/1.1

Host: www.example.com

Accept: application/sdp

W-> C : HTTP/1.0 200 OK

Content-Type: application/sdp

C-> A : SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0

Cseq:1

Transport : RTP/AVP/UDP;unicast;client_port=3056-3057

A-> C : RTSP/1.0 200 OK

Cseq:1

Session: 12345678

Transport : RTP/AVP/UDP;unicast;client_port=3056-3057

server_port=5000-5001

C->V : SETUP rtsp://video.example.com/twister/video.en RTSP/1.0

Cseq:1

Transport : RTP/AVP/UDP;unicast;client_port=3058-3059

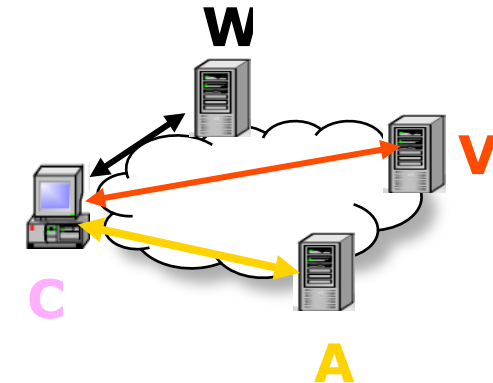
A-> C : RTSP/1.0 200 OK

Cseq:1

Session: 23456789

Transport : RTP/AVP/UDP;unicast;client_port=3058-3059

server_port=5002-5003





RTSP Message sequence (contd.)

C->V: PLAY rtsp://video.example.com/twister/video RTSP/1.0

Cseq: 2

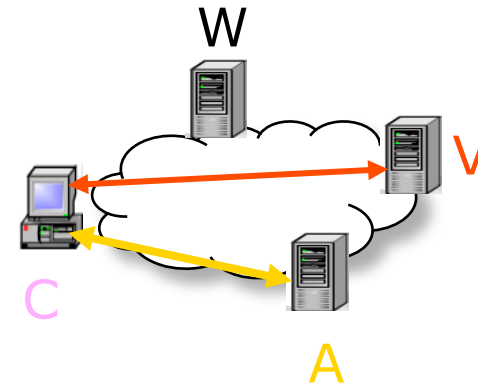
Session: 23456789

V->C: RTSP/1.0 200 OK

Cseq: 2

Session: 23456789

**RTP-Info: url=rtsp://video.example.com/twister/video;
seq=12312232;**



C->A: PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0

Cseq: 2

Session: 12345678

A->C: RTSP/1.0 200 OK

Cseq: 2

Session: 12345678

**RTP-Info: url=rtsp://audio.example.com/twister/audio.en;
seq=876655;**



References

- [1] B. A. Forouzan, "TCP/IP Protocol Suite",
Third edition,
- [2] H. Schulzrinne, S. Casner, R. Frederick and V.
Jacobson, "RTP: a transport protocol for real-time
applications", RFC 3550, July 2003.
- [3] H. Schulzrinne, A. Rao and R. Lanphier, "Real Time
Streaming Protocol (RTSP)", RFC 2326, April 1998.