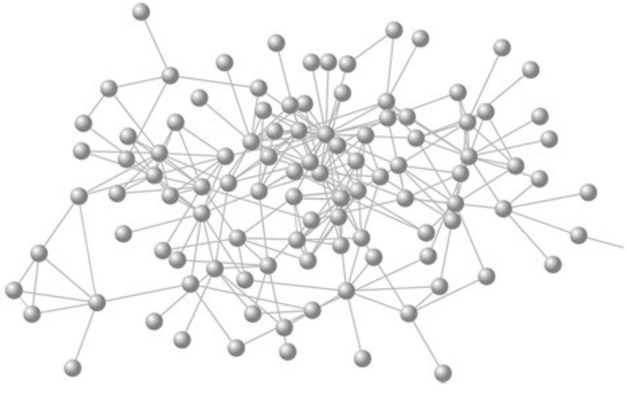
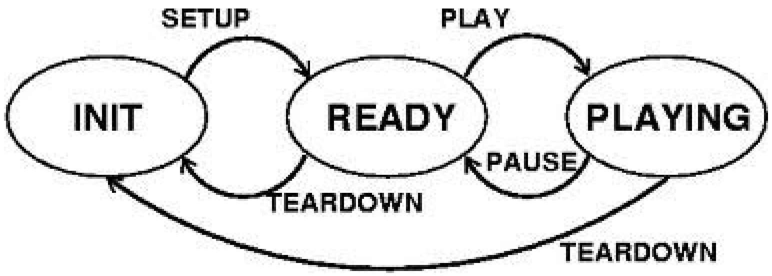
CSCD 433/533
Advanced Networks
Spring 2017

Lecture 16
RTSP and
Transport Protocols/ RTP

Read: RTP, 6.4 and RTSP, 7.4



Topics



- Multimedia Player
- RTSP Review
- RTP Real Time Protocol
  - Requirements for RTP
  - RTP Details
  - Applications that use RTP

Re-cap Multimedia

- Last time … looked at Multimedia applications
- Noted their characteristics
    - Need time guarantees
- Also looked at RTSP – Real Time Streaming Protocol
- Multimedia players and applications use RTSP and RTP for real time traffic

Multimedia Player

- **Four things it Does**
  1. Manage User Interface
  2. Handle Transmission
  3. Decompress Content
  4. Eliminate Jitter

Multimedia Player

1. Manage User Interface

- Provide buttons for Interaction
  - Uses Protocol with built-in support for users
  - Look and feel of remote control
  - Allows users to interact with media
- Makes it look attractive, cute, cool
- Users preference important

Multimedia Player

2. Handle Transmission

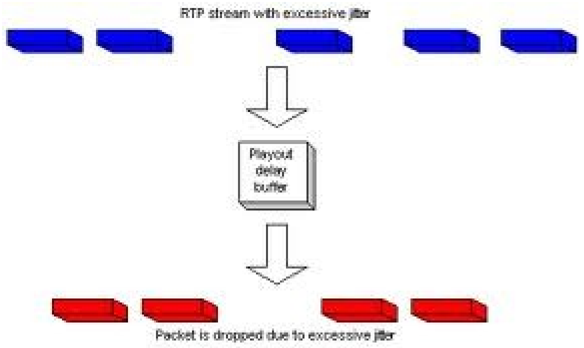
- If using **TCP**, no lost packets
  - But likely will add jitter
  - Retransmit packets not always appropriate
    - Timing off, missed playback time

- If using **UDP**, lost packets
  - Lost in small bursts
  - Must plan for this in advance
  - Will depend on type of application, stored vs. live

Multimedia Player

2. Handle Transmission continued ...

- How does the live or streaming application handle missing packets?
  - Depends on Streaming vs. Live
  - Streaming (stored)
    - Delay start time even more
    - Use a larger buffer
    - Streaming - Player, might pause
      - annoying to users!
  - Live
    - Packet will be skipped
      - Sounds and video can often mask missing packets
    - **How do they do it?**



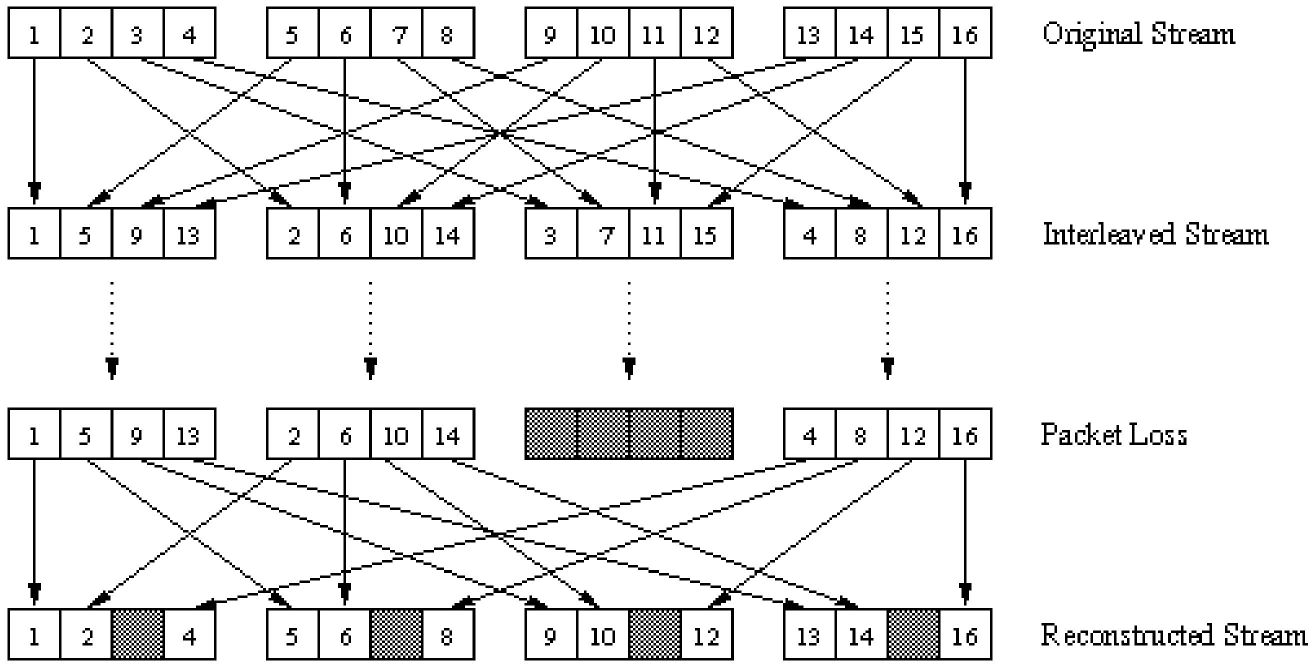
Multimedia Player

2. Missed Packets

- Two Strategies
  1. Forward Error Correction
     - Send extra data, parity packet
     - Group of 5 packets, 1 is parity
     - Lost one, then XOR of other 4 to get 5th packet
     - Disadvantage, must send extra packet no matter what
     - Other methods, an entire subject in itself
  2. Interleaving
     - Mix up packets before sending
     - Unmix packets at receiver
     - Spreads out loss over time
     - Won't notice it as much ... few lost packets intermixed vs. one missing burst of packets

Recovering from packet loss Interleaving

- Divide 20 msec of audio data into smaller units of 5 msec each and interleave
- Upon loss, have a set of partially filled chunks



Multimedia Player

3. Decompress Stream

- Must account for many compression schemes
  - MPEG, H.261, MP3, G.729 etc.
- Underlying protocols help by supporting multiple formats
  - RTP,
- Often adapt as conditions allow can switch schemes

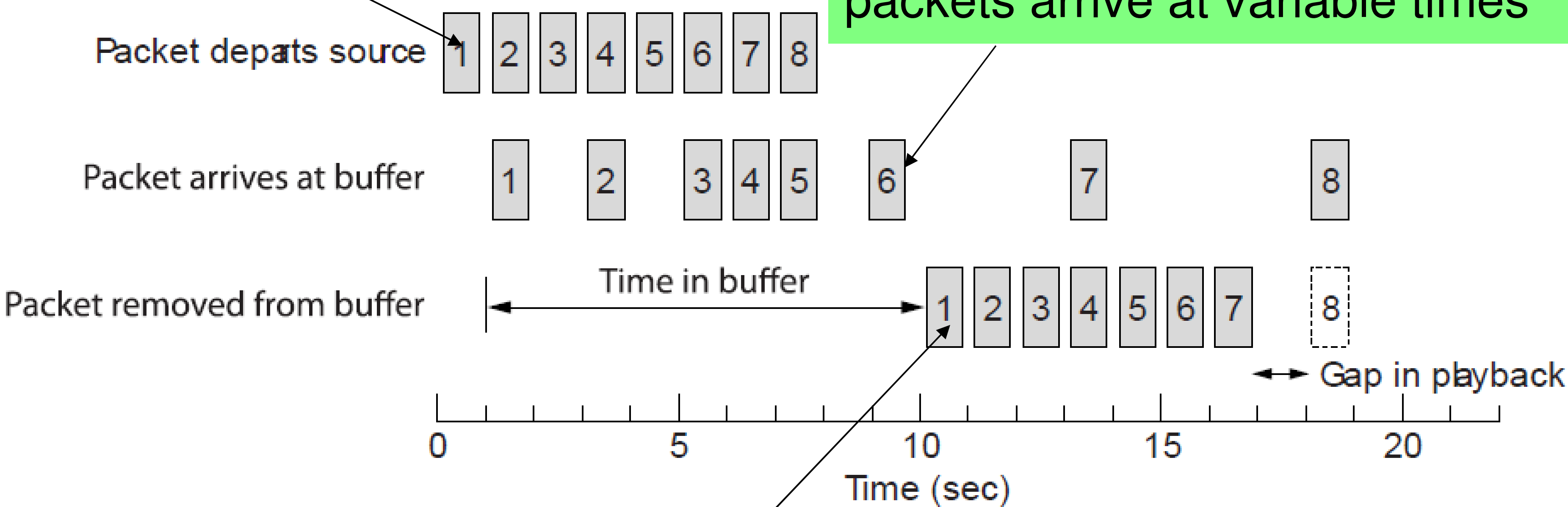
4. Eliminate Jitter

- How do players handle that?

Handling Jitter

Packets created at a steady rate

Due to network delays
packets arrive at variable times

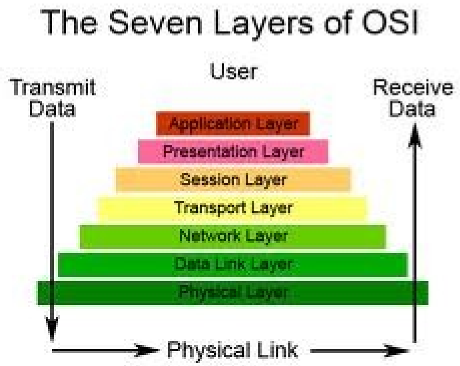


Delay playback and buffer packets before playback

Smoothing the output stream by buffering packets

Protocols for Multimedia

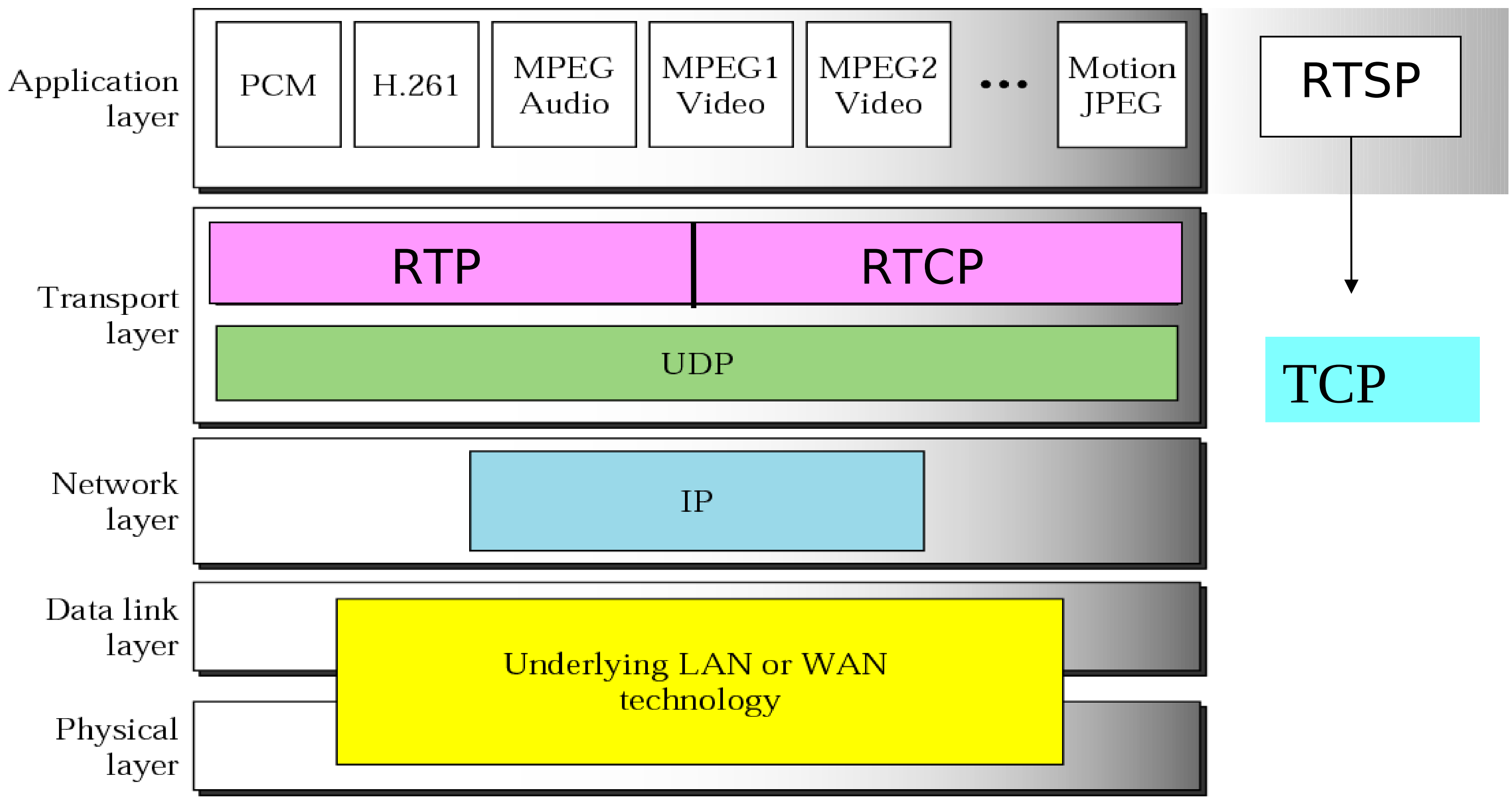
- Supporting Multimedia requires many protocols
- Multiple protocols ... not just one!
- Multiple network layers contribute to multimedia functioning over the Internet

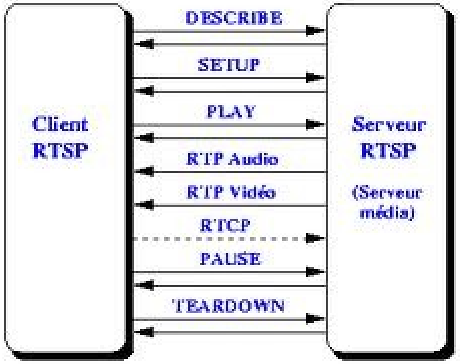


Interaction of Multimedia Protocols

- **Network layer protocols**
    - Provide basic network service support such as addressing
    - IP, network protocol
- **Transport protocols**,
    - Provide end-to-end transport functions for streaming applications
        - Includes UDP, TCP, RTP, RTCP
- **Session-control protocols, Application layer**
    - Define messages to control delivery of multimedia data during an established session
    - Examples are RTSP and Session Initiation Protocol(SIP)

Protocol stack for multimedia services





Real Time Streaming Protocol

Introduction

- RTSP is an application-level protocol for the control of real-time streaming data.
- IEFT Standard
  - RFC 2326
- It uses **RTP** as the underlying data delivery protocol and offers a VCR-like control to the user:
  - Play, Stop, Pause, FF and REW
  - Random access to any part of media clip

RTSP and HTTP

- RTSP is similar to HTTP/1.1 in terms of syntax and operation but differs in several important aspects.

- With RTSP, both client and the server can issue requests during interaction,
  - HTTP where the client always issues the requests (for documents)

RTSP and HTTP

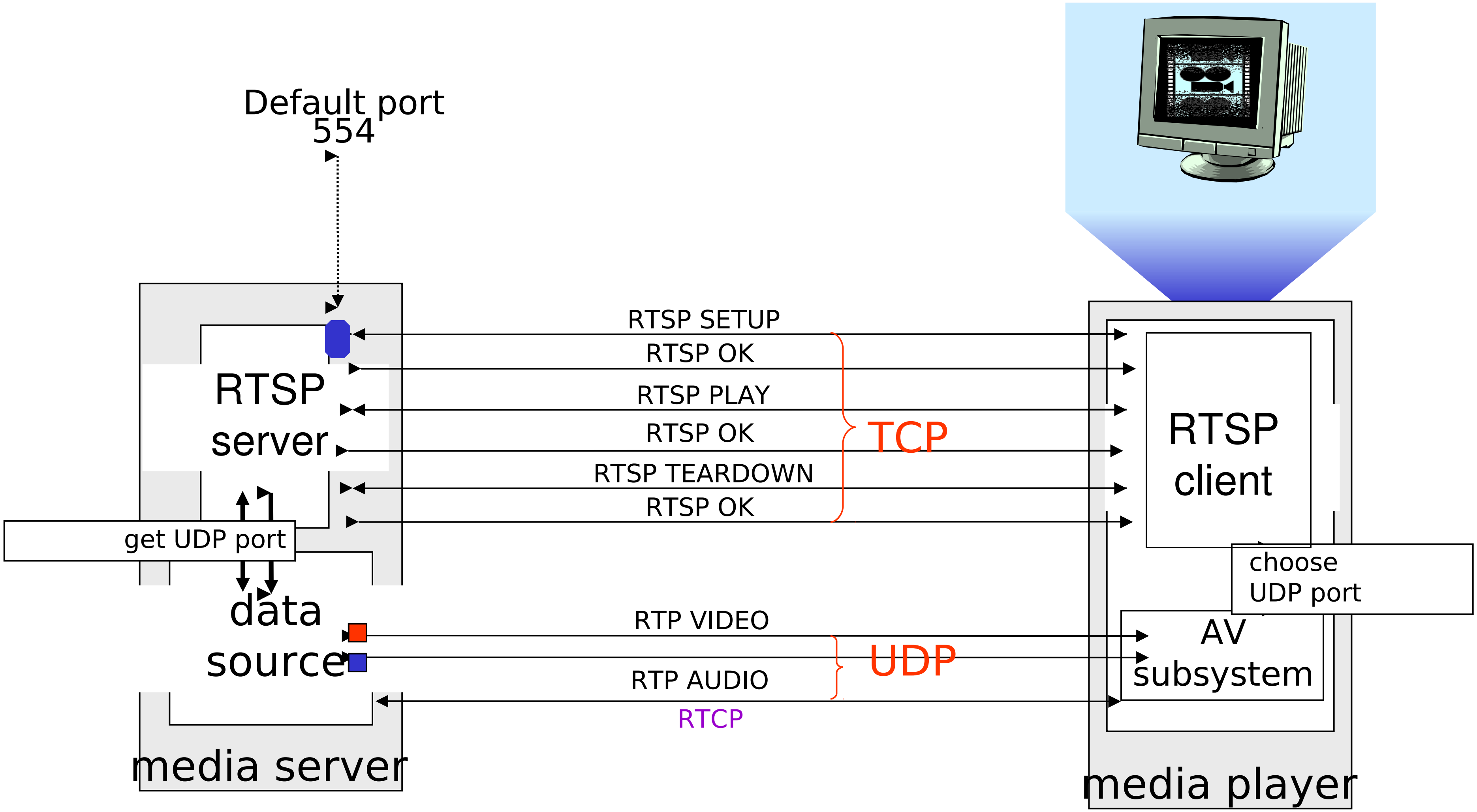
- RTSP *maintains a state* by default which happens to be very important in streaming media files.

- HTTP protocol is a *stateless protocol*
  - HTTP is unable to retain a memory of the identity of each client that connects to a web site
  - Treats each request for a web page as a unique and independent connection, no relationship to connections that preceded it

Review of RTSP

- Recall ... RTSP
  - Establishes and controls
    - Either one or several time-synchronized streams of media such as audio and video
  - RTSP
    - Acts as a "network remote control" for multimedia servers
    - Provides a framework to enable controlled, on-demand delivery of real-time data

RTSP Session



RTSP – Operation

1. Client sends a control request
   - Client constructs the request
     - Method, request URL, and protocol version number
   - Client includes a general header, a request header, and possibly an entity header (like HTTP protocol)
2. Server executes request if possible
3. Server returns a response containing a status-line and general response and entity headers
   - Status-line:
     - Protocol version, numeric status code, textual description

RTSP – Operation (cont.)

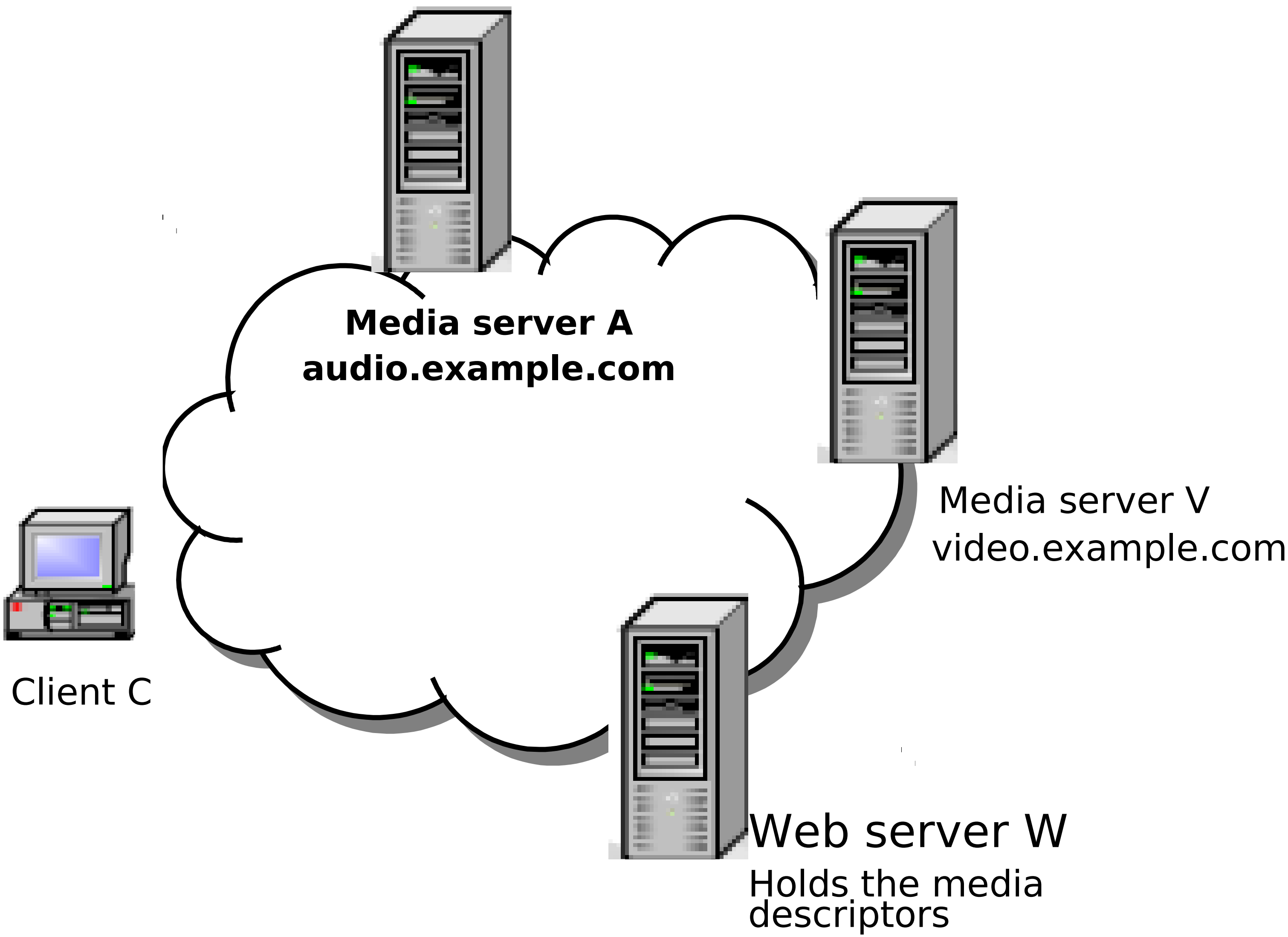
- Media streams are left unspecified
  - Could be RTP streams or any other form of media transmission
- RTSP only specifies control
  - Client and server software must maintain mapping between control channel and media streams
- Control request and responses may be sent via TCP or UDP
  - Are sequenced
  - UDP requires construction of retransmission mechanisms, so ...
  - Mostly uses TCP

RTSP Methods

- Major methods
  - SETUP: Server allocates resources for a stream and starts an RTSP session
  - PLAY: Starts data tx on a stream
  - PAUSE: Temporarily halts a stream
  - TEARDOWN: Free resources of the stream

- Additional methods
  - OPTIONS: Get available methods
  - ANNOUNCE: Change description of media object
  - DESCRIBE: Get low level descr. of media object
  - RECORD: Server starts recording a stream
  - REDIRECT: Redirect client to new server
  - SET_PARAMETER: Device or encoding control

Example:Media on demand (Unicast)

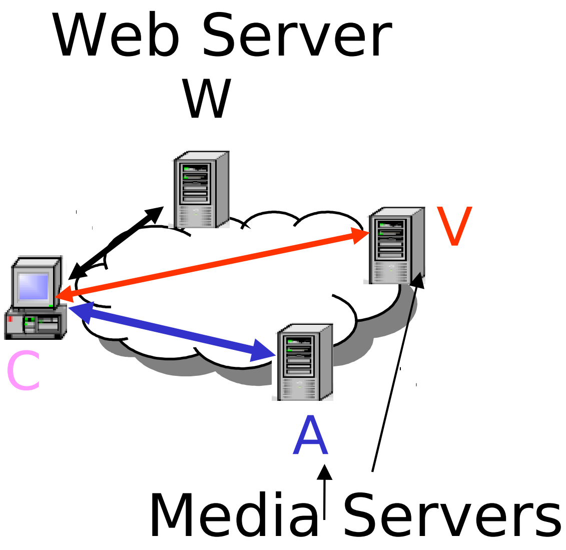


RTSP Message sequence

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

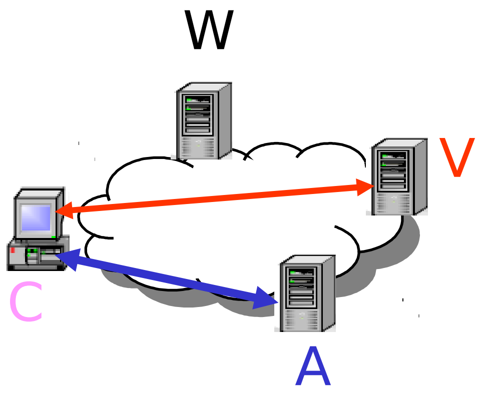
C-> A : SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0
Cseq:1
Transport : RTP/AVP/UDP;unicast;client_port=*3056-3057*
A-> C : RTSP/1.0 200 OK
Cseq:1
Session: 12345678
Transport : RTP/AVP/UDP;unicast;client_port=3056-3057
server_port=*5000-5001*

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;

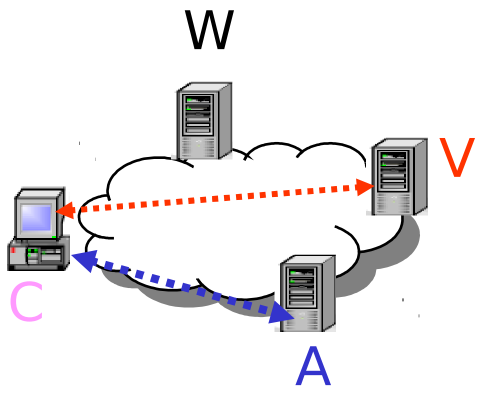
C->A: PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0
Cseq: 2
Session: 12345678

A->C: RTSP/1.0 200 OK
Cseq: 2
Session: 12345678
RTP-Info: url=rtsp://audio.example.com/twister/audio.en;
seq=876655;

RTSP Message sequence (contd.)

C->A: TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/1.0
Cseq: 3
Session: 12345678

A->C: RTSP/1.0 200 OK
Cseq: 3



C->V: TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0
Cseq: 3
Session: 23456789

V->C: RTSP/1.0 200 OK
Cseq: 3

Real-time Protocol (RTP)

RTP Overview

- One of those confusing protocols
  - Application protocol but performs transport functions
- Design Goal
  - Supports range of multimedia formats
    - Examples: H.264, MPEG-4, MJPEG, MPEG
  - Allows new formats without revising RTP standard
- Main purpose
  - Multiplex several multimedia streams onto single stream of UDP packets
  - Runs over UDP

RTP Overview

- What does it do?
- Provides end-to-end delivery service for real-time data, in unicast and multicast sessions

- **Offers synchronization services**
  - Timestamping
  - Packet identification and loss detection, sequence numbering and
  - Delivery monitoring/feedback (through RTCP)

RTP Overview

- What doesn't it do?
- Does not provide in-order and reliable delivery of packets
- Does not provide **timely delivery of packets**
- Does not ensure QoS guarantees
- Independent of the transport protocol TCP, UDP
- RTP/RTCP are usually implemented within applications, RTP libraries

RTP Overview

- Contains information so receivers can work with multimedia information

1. Packets are numbered 1,2,3,4,......

   Why do you need that?

   - If packets missing, application deals with it
     - Video - just skip that frame
     - Audio - Maybe interpolate the values
   - Can't retransmit - would arrive too late
   - RTP - has no ACK's and no mechanism to request retransmission

RTP Overview

2. Packets are time-stamped 12:34:65

Why do you need that?

- Used to enable receiver to play back received samples at appropriate intervals
- When several media streams are present, timestamps are independent in each stream
- Could be used to synchronize streams

- Timestamps in RTP packets tied to video and audio sampling clocks
  - **Not** tied to the wall-clock time!

RTP Overview

3. Packets have a payload type identifier

Why do you need that?

- Indicates format of payload and determines its interpretation by application
  - Since RTP supports multiple formats for video and audio,
  - Must tell recipient expected format

RTP and RTCP

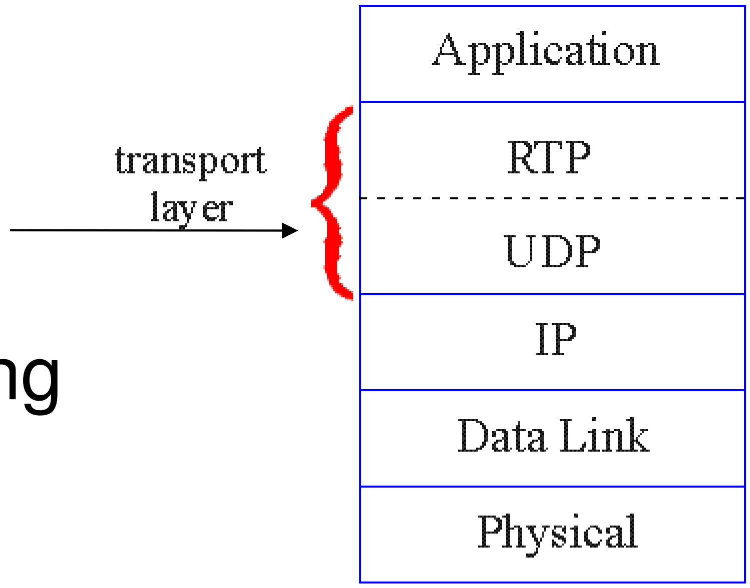
- Actually, two proposed protocols
  - RTP and RTCP – Used together
  - **Real Time Protocol and Real Time Control Protocol**
  - Pair of protocols developed by IETF 1990's
    - **RTP** – Deals with sending data
    - **RTCP** – Used to send control information associated with a given data flow
    - More about RTCP later ....

RTP runs on top of UDP

- RTP usually implemented as part of the application, libraries or direct support

- RTP libraries provide a transport-layer interface that extends UDP

  Adds
  - Payload type identification
  - Packet sequence numbering
  - Time-stamping



RTP Example

- Sending 64 kbps PCM-encoded voice over RTP

  PCM = Pulse Code Modulation

1. Application collects encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
2. Audio chunk along with RTP header forms RTP packet
3. Which is then encapsulated into a UDP segment

How RTP works

IP Header | UDP Header | RTP Header | RTP **Video** Payload

IP Header | UDP Header | RTP Header | RTP **Audio** Payload

- Video and audio payloads are sent separately
- Uses sequence number to synchronize audio and video once received

Application Level Framing (ALF)

- Two Key Ideas used in RTP
- Paper by Clark and Tennenhouse, MIT, 1990

- 1. Application Level Framing (ALF)
- 2. Integrated Layer Processing

Application Level Framing

Idea is

- Application should break data into aggregates that are meaningful to application
  - Called **Application Data Units** (ADU's)
- Frame boundaries are then preserved by lower level protocols

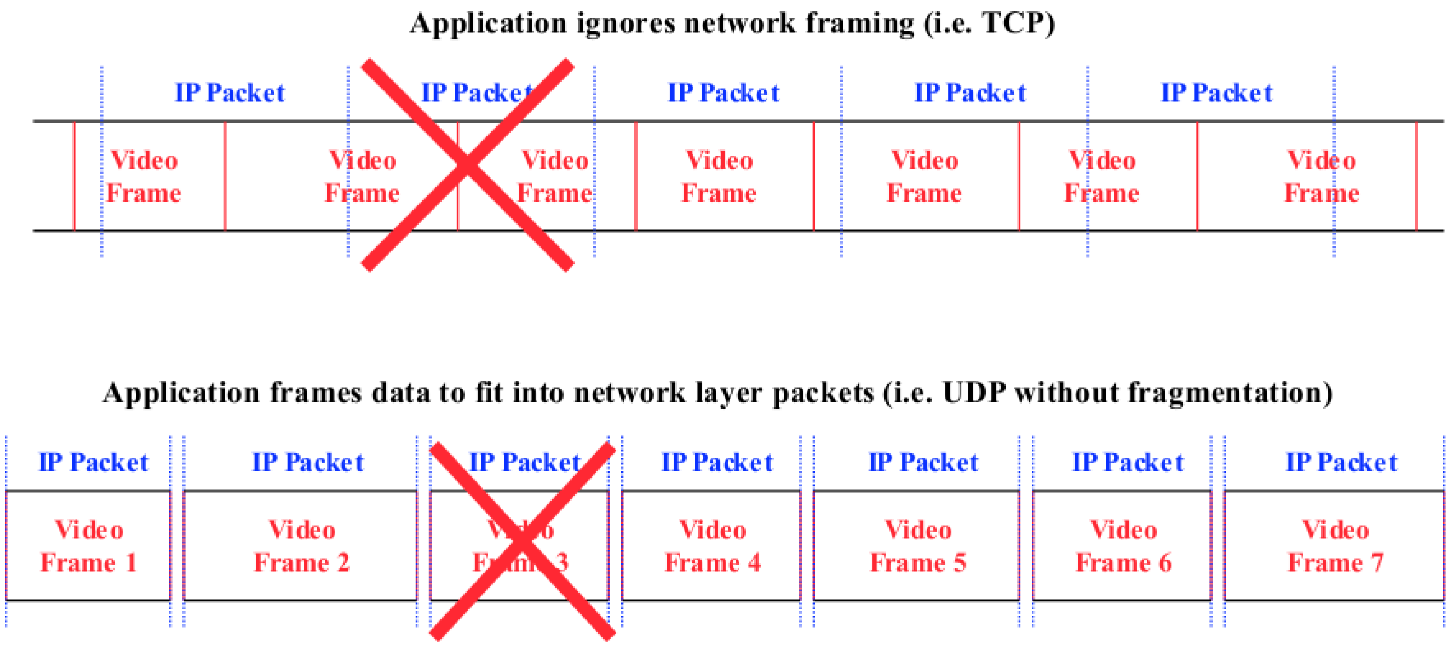
- Simply speaking, it is **application**
  - Not the **transport protocol** who determines contents of packets
  - Application-oriented packetization scheme

Application Level Framing

Idea Continued

- Sender sends enough information for reconstruction at receiver
  - Timestamps and numbering allows receiver to put back together meaningful packet stream
  - Receiver can choose to ignore lost packets
    - Error control is by application

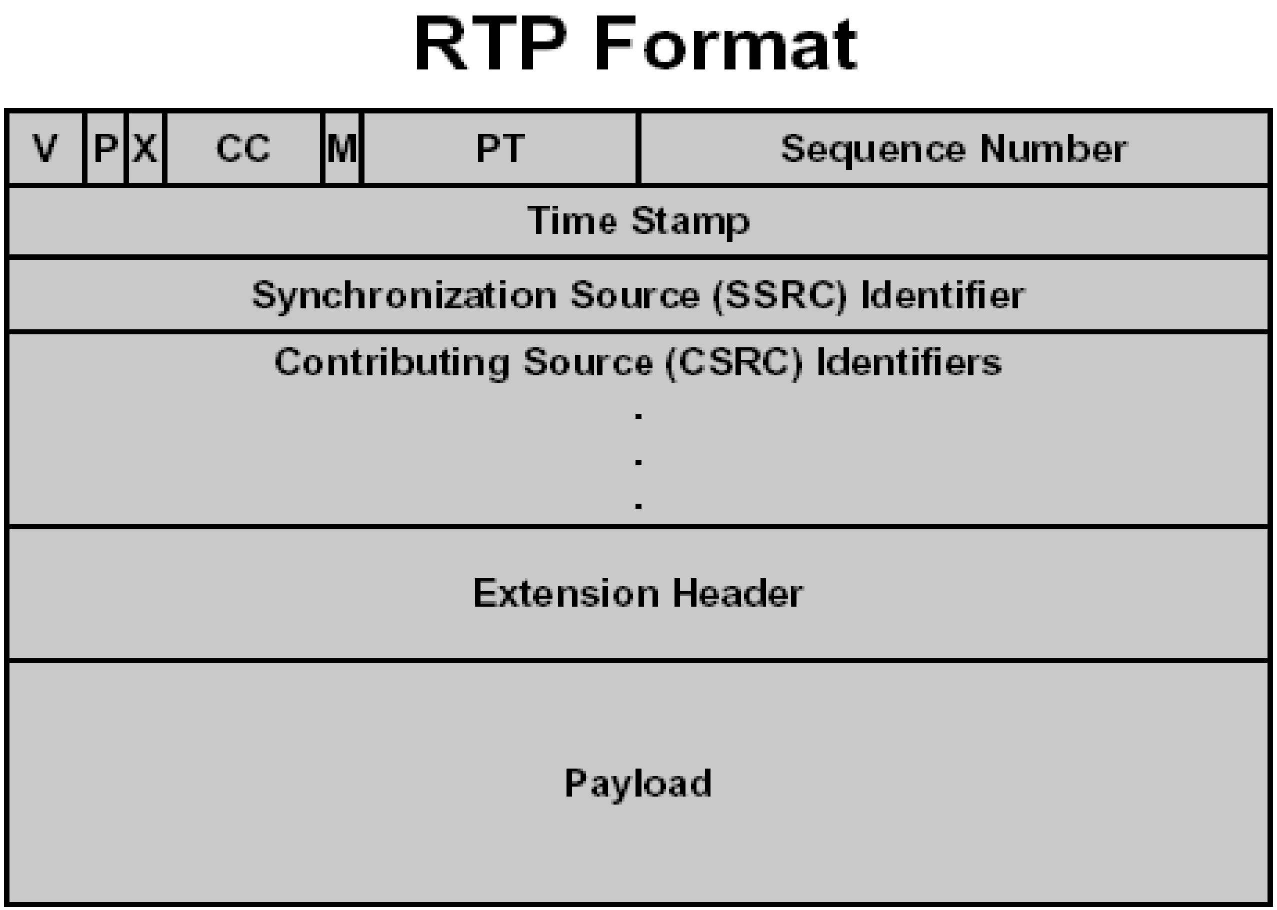
ALF Example


Application ignores network framing (i.e. TCP)
IP Packet
IP Packet
IP Packet
IP Packet
IP Packet
Video Frame
Video Frame
Video Frame
Video Frame
Video Frame
Video Frame
Video Frame
Application frames data to fit into network layer packets (i.e. UDP without fragmentation)
IP Packet
IP Packet
IP Packet
IP Packet
IP Packet
IP Packet
IP Packet
Video Frame 1
Video Frame 2
Video Frame 3
Video Frame 4
Video Frame 5
Video Frame 6
Video Frame 7

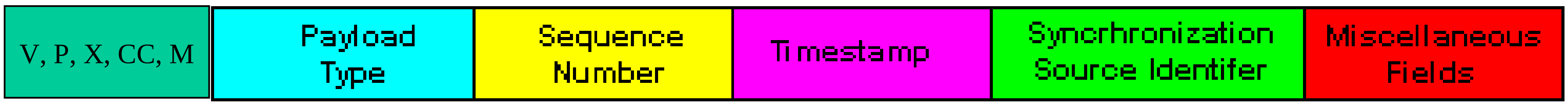
Integrated Layer Processing

- Data processing not separated into strict network layers
  - All processing of one complete ADU can be done in one integrated processing step
  - Within the application !!!!
  - Use libraries to implement protocol within application
- Engineering principle is called Integrated Layer Processing
- Moved some network layer jobs into application
- Network layers can be blurred for faster processing

RTP Packets Formats



RTP Header

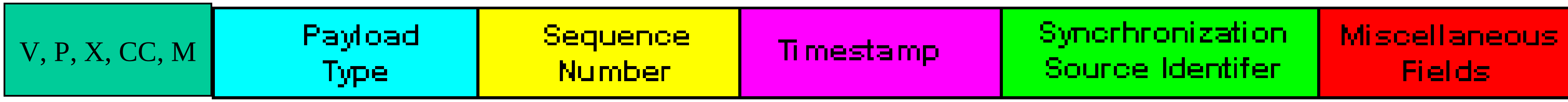


RTP Header

- V - version, currently = 2
- P - padding, 1-bit, If set, packet has one or more padding bytes. Last byte in payload is count of padding bytes
- X - extension header, 1-bit, If set, a single header extension follows fixed header. Designed for experimental additions to RTP.

RTP Header

32bits



RTP Header

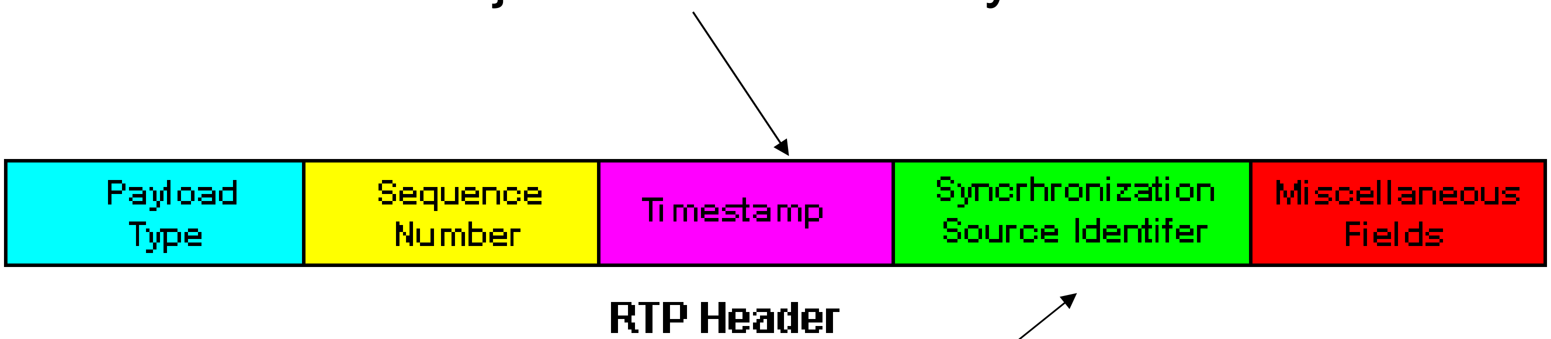
Payload Type (7 bits): Indicates type of encoding
Sender can change encoding in middle of conference, and sender informs receiver through this payload type field

- •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

Packet 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

RTP Header Continued

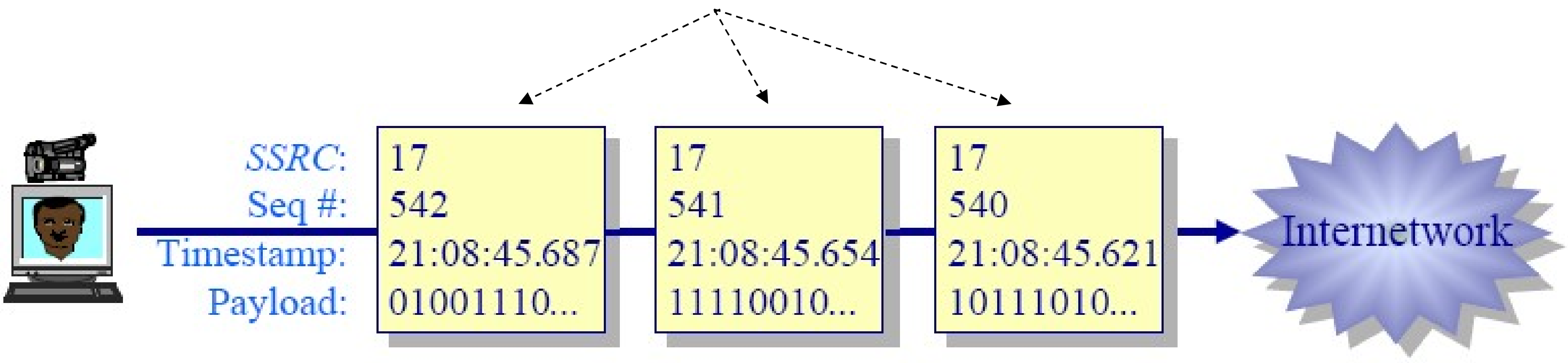
- **Timestamp field -** Reflects the sampling instant of the first byte in the RTP data packet
- Used to remove jitter introduced by the network



- **SSRC field -** Identifies the source of the RTP stream
- An id for the source of a stream.
  - Used to multiplex several streams onto single UDP stream

RTP Basics of Data Transmission

RTP Packets

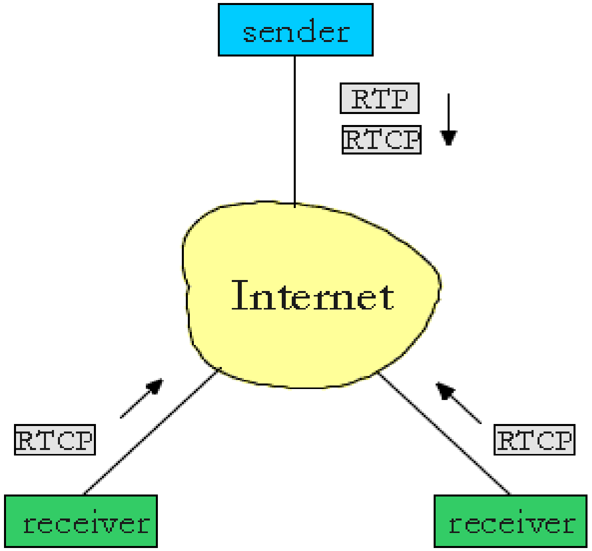


Real-Time Control Protocol (RTCP)

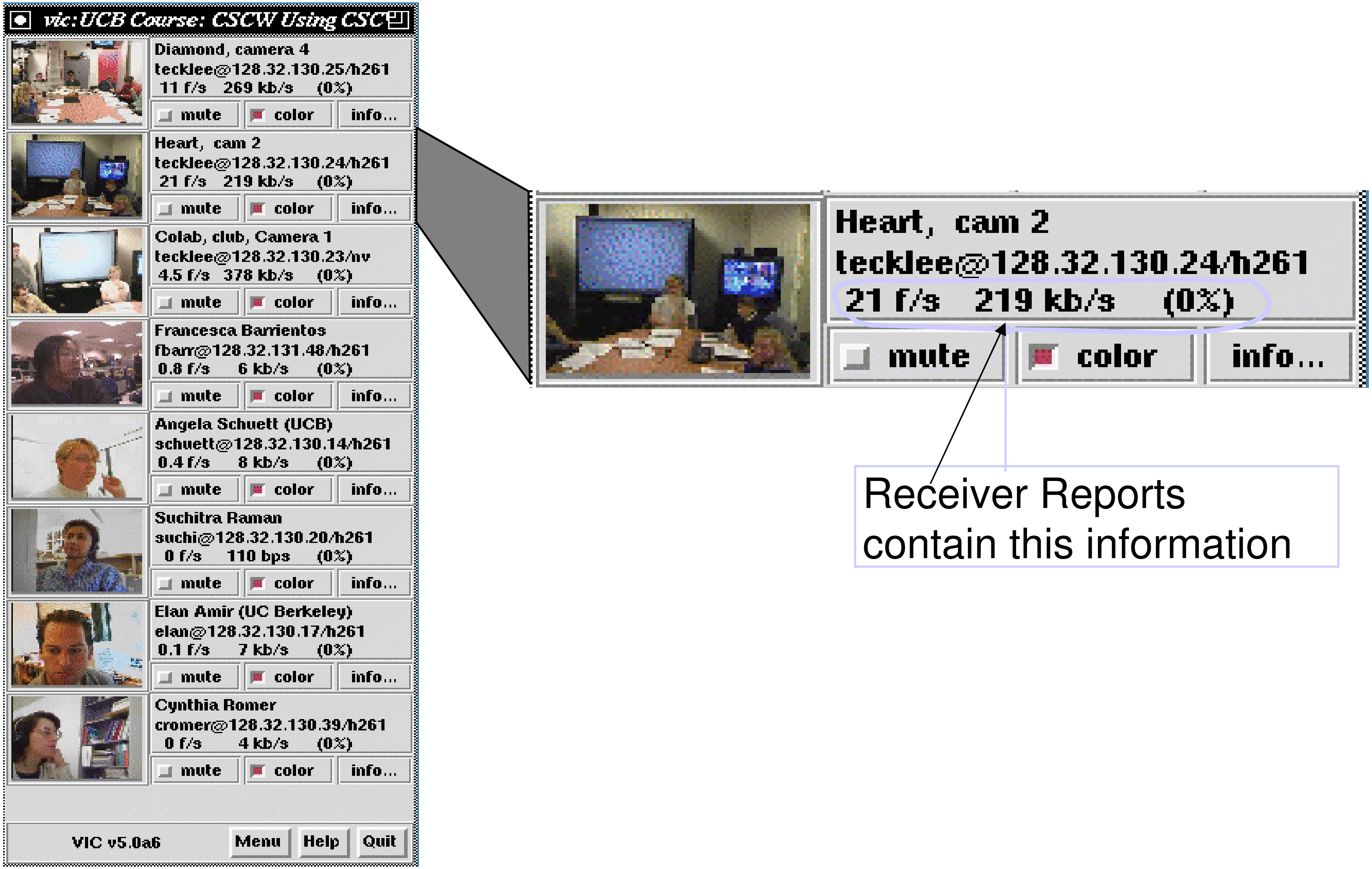
- Works in conjunction with RTP
- Each RTP session participant periodically transmits RTCP control packets to all other participants
- Each RTCP packet contains sender and/or receiver reports
  - Report statistics useful to running application

RTP Control Protocol (RTCP)

- RTCP reports packets exchanged between sources and destinations
  - Receiver reception report
  - Sender report
  - Source description report
- Reports contain statistics such as the number of RTP-packets sent, number of RTP-packets lost, inter-arrival jitter
- Used by application to modify sender transmission rates and for diagnostics purposes



RTCP Example - VIC



Receiver Reports contain this information

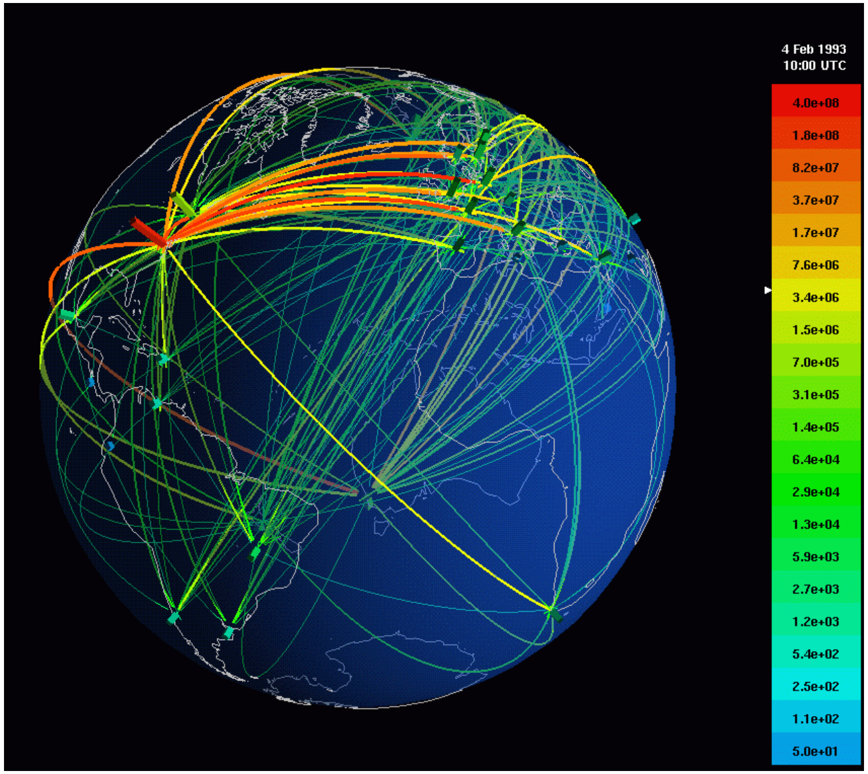
References

- Clark and Tennehouse ALF Article
  - Architectural Considerations for a New Generation of Protocols" in Proceedings of the SIGCOMM '90 Symposium on Communications Architectures, September 1990, Computer Communications Review
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.5125&rep=rep1&type=pdf
- FAQ on RTP
  - http://www.cs.columbia.edu/~hgs/rtp/faq.html
- RTP Overview - Nice
  - http://www4.informatik.uni-erlangen.de/Projects/JRTP/pmt/node30.html#SECTION0072200000000000000
- RTSP Overview - Good Summary

  http://www.javvin.com/protocolRTSP.html

Summary

- TCP/UDP good for regular applications with no timing requirements
- However, one size doesn’t always fit all
- RTP and real-time protocols good for multimedia applications that need
  - Timely delivery of data
  - Allows for data coding in different ways
  - Allows the applications to decide on packet coding to optimize it’s packet delivery
- Other protocols have developed
  - RTMP and others



Read: RTP, 6.4 and RTSP, 7.4

Re-cap Multimedia

- Last time … looked at Multimedia applications
- Noted their characteristics
    - Need time guarantees
- Also looked at RTSP – Real Time Streaming Protocol
- Multimedia players and applications use RTSP and RTP for real time traffic

Introduction

- RTSP is an application-level protocol for the control of real-time streaming data.
- IEFT Standard
  - RFC 2326
- It uses **RTP** as the underlying data delivery protocol and offers a VCR-like control to the user:
  - Play, Stop, Pause, FF and REW
  - Random access to any part of media clip

RTSP and HTTP

- RTSP is similar to HTTP/1.1 in terms of syntax and operation but differs in several important aspects.

- With RTSP, both client and the server can issue requests during interaction,
  - HTTP where the client always issues the requests (for documents)

RTSP and HTTP

- RTSP *maintains a state* by default which happens to be very important in streaming media files.

- HTTP protocol is a *stateless protocol*
  - HTTP is unable to retain a memory of the identity of each client that connects to a web site
  - Treats each request for a web page as a unique and independent connection, no relationship to connections that preceded it

RTP Overview

- What does it do?
- Provides end-to-end delivery service for real-time data, in unicast and multicast sessions

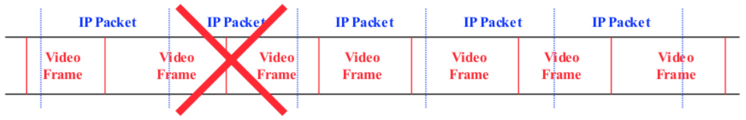
- **Offers synchronization services**
  - Timestamping
  - Packet identification and loss detection, sequence numbering and
  - Delivery monitoring/feedback (through RTCP)

RTP Overview

- What doesn't it do?
- Does not provide in-order and reliable delivery of packets
- Does not provide **timely delivery of packets**
- Does not ensure QoS guarantees
- Independent of the transport protocol TCP, UDP
- RTP/RTCP are usually implemented within applications, RTP libraries

ALF Example

Application ignores network framing (i.e. TCP)



Application frames data to fit into network layer packets (i.e. UDP without fragmentation)