Media Distribution Across Internet

- Media Distribution Category
- Media Streaming
- Streamed Media On Demand Delivery
- Streamed Media Internet Broadcast
- Streamed Media Server and Client/Player
- Streaming Service System
- RTSP (Real Time Stream Protocol)
- RTP (Real-time Transport Protocol)

Media Distribution Catalog

- Media distribution - Deliver media contents to users

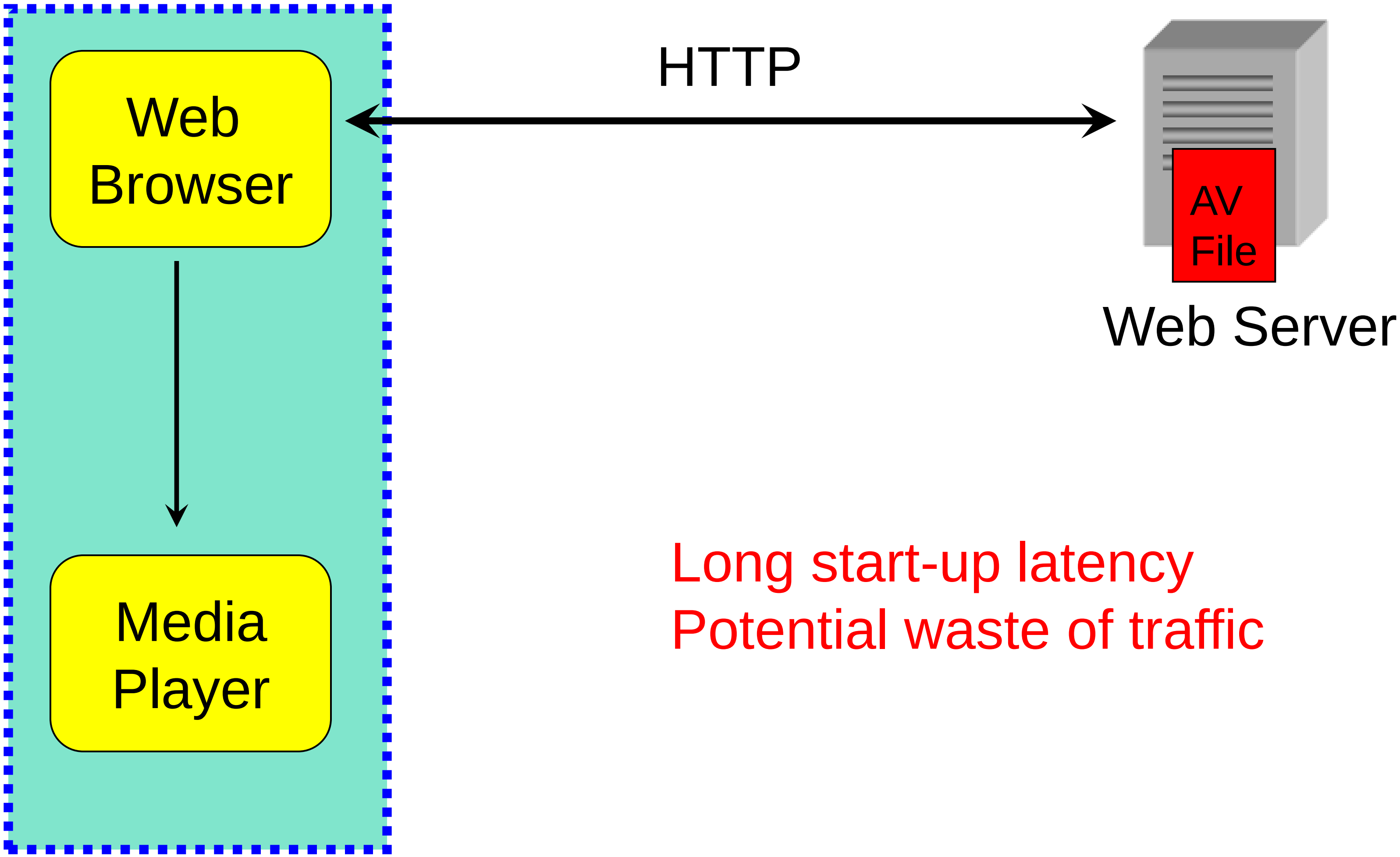
✧ Delivery via disc:
  - Merits: Large storage, high audiovisual quality
  - Demerits: long delivery time, inflexible

✧ Delivery via Internet:
  - ➢ Non realtime delivery:
    - Called download service:

      >download ***all*** data, save to disc, and play
    - Using data file transfer protocols like ftp and http via ftp or web server
  - ➢ Realtime delivery:
    - Called streaming service:

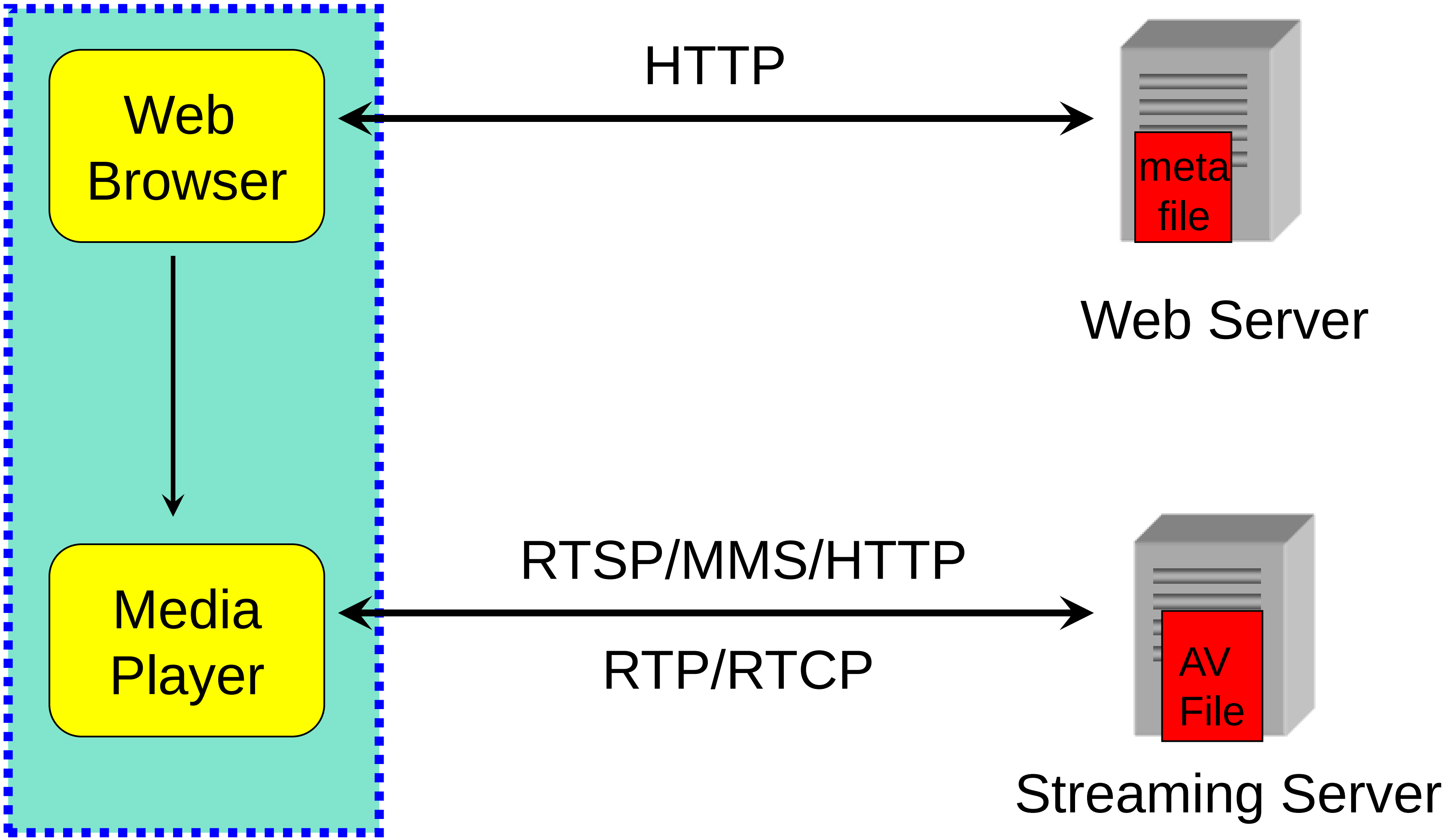
      >download & play simultaneously, partial data in buffer, no data in disc
    - May use http and web server to provide limited streaming service
    - Often use RTSP/RTP and media server for rich streaming service

Non Realtime Delivery: Download service

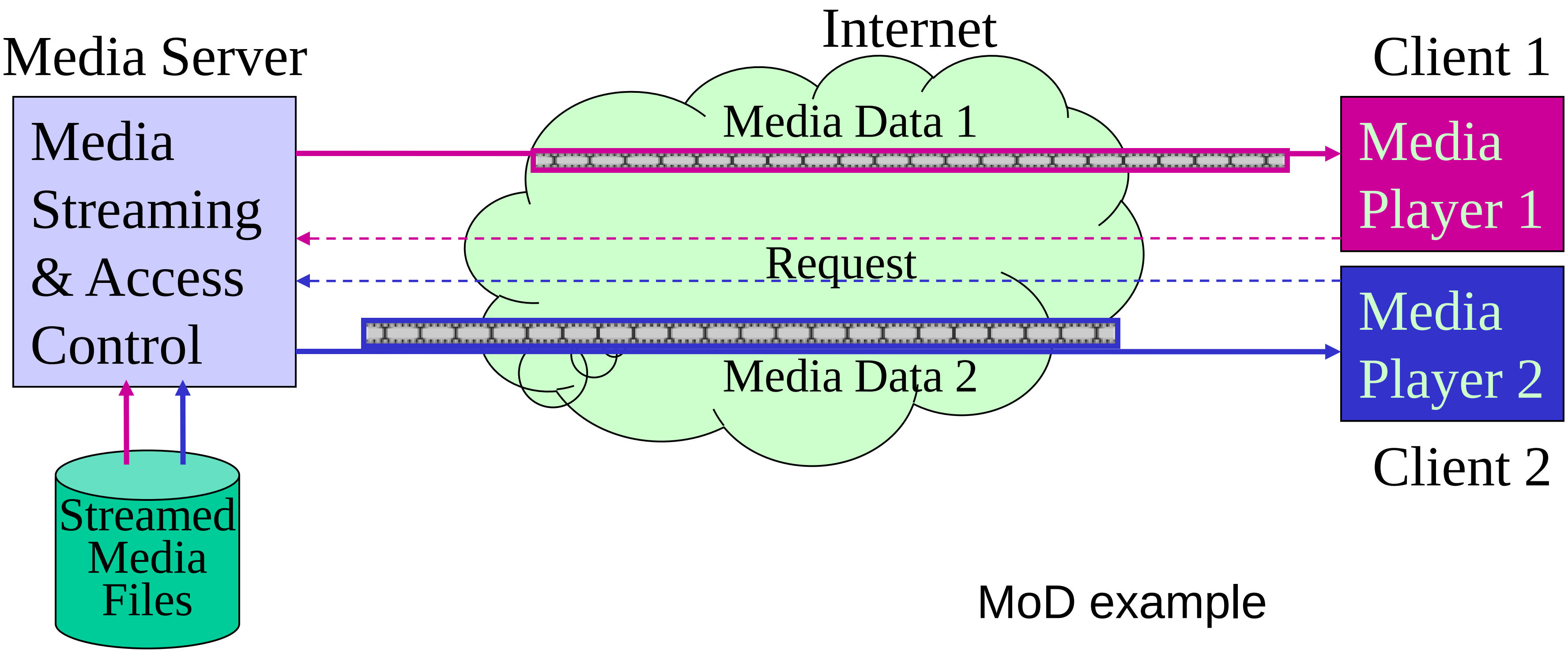


Long start-up latency
Potential waste of traffic

Realtime Delivery: Stream Service



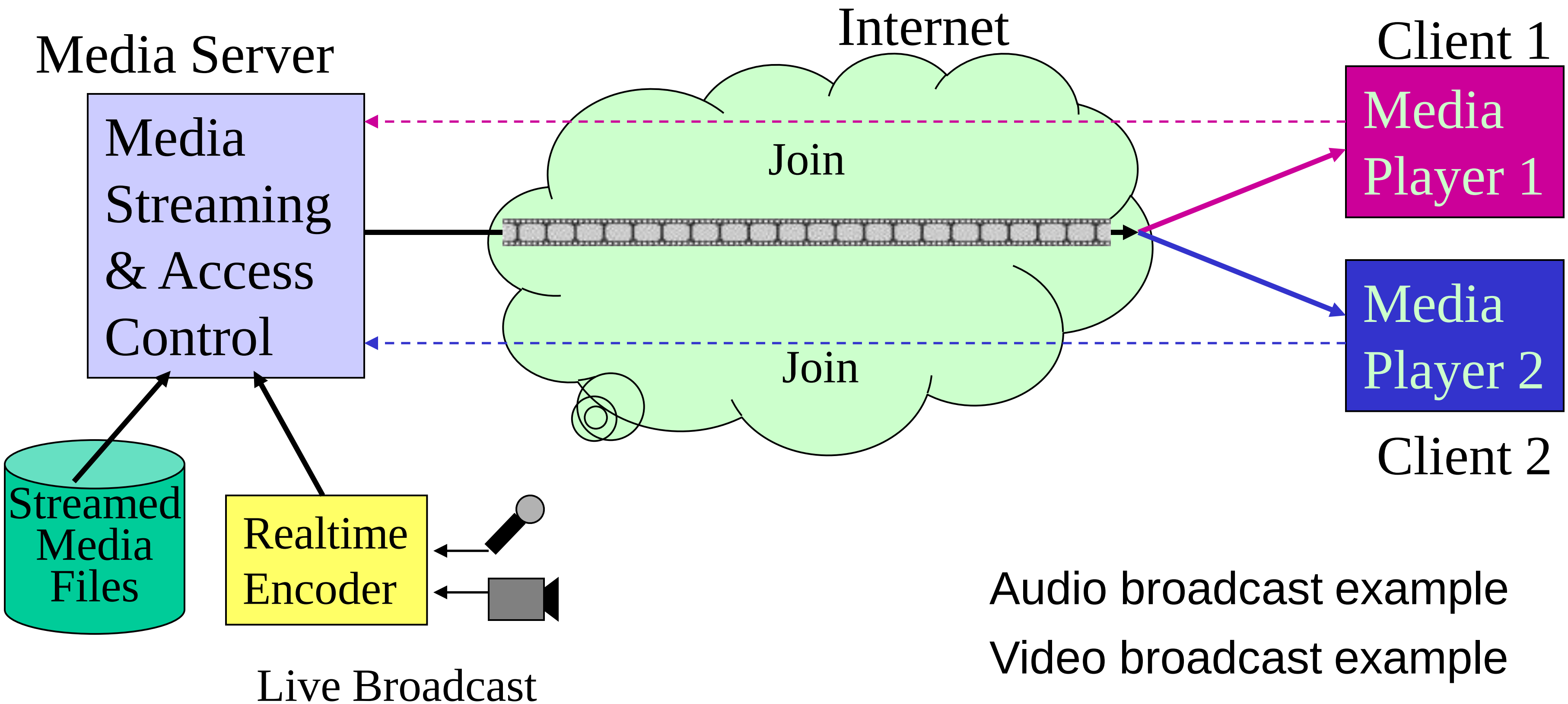
Streamed Media On Demand Delivery



MoD example

- Media on demand (MoD)
  - Streamed media are saved in media server as streamed file format
  - Clients, i.e., media player, access media contents independently
  - Media content is played from the file beginning for each client's request
  - User can control playing, such fast forward, pause, …
  - Like rent a video tape or DVD and replay it in your cassette/DVD palyer

Streamed Media Broadcast



Audio broadcast example

Video broadcast example

- Media Internet Broadcast (MIB) or Webcast
  - Media may be stored in server or captured lively and encoded in realtime
  - Clients can join a broadcast and same media content goes to all clients
  - Users watch/listen the broadcast from the current state not from beginning
  - Users can't control its playing such fast forward, stop, etc.
  - Like conventional radio and TV broadcast

Streaming Media Service History

1992

- MBone
- RTP version 1
- Audiocast of 23rd IETF mtg

1994

- Rolling Stones concert on MBone

1995

- ITU-T Recommendation H.263
- RealAudio launched

1996

- Vivo launches VivoActive
- Microsoft announces NetShow
- RTSP draft submitted to IETF

1997

- RealVideo launched
- Microsoft buys VXtreme
- Netshow 2.0 released
- RealSystem 5.0 released
- RealNetworks IPO

1998

- RealNetworks buys Vivo
- Apple announces QuickTime Streaming
- RealSystem G2 introduced

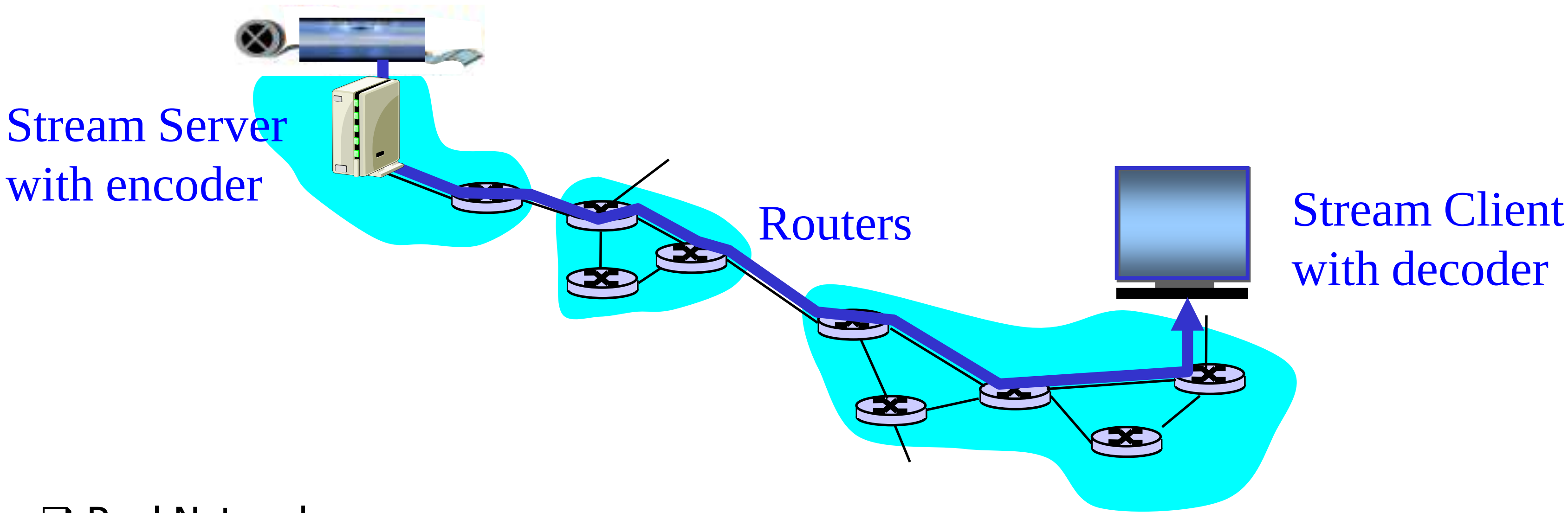
1999

- RealNetworks buys Xing
- Yahoo buys Broadcast.com for $ 5.7B
- Netshow becomes WindowsMedia

2000

- RealPlayer reaches 100 million users
- Akamai buys InterVu for $2.8B
- ***Internet stock market bubble bursts***
- WindowsMedia 7.0
- RealSystem 8.0

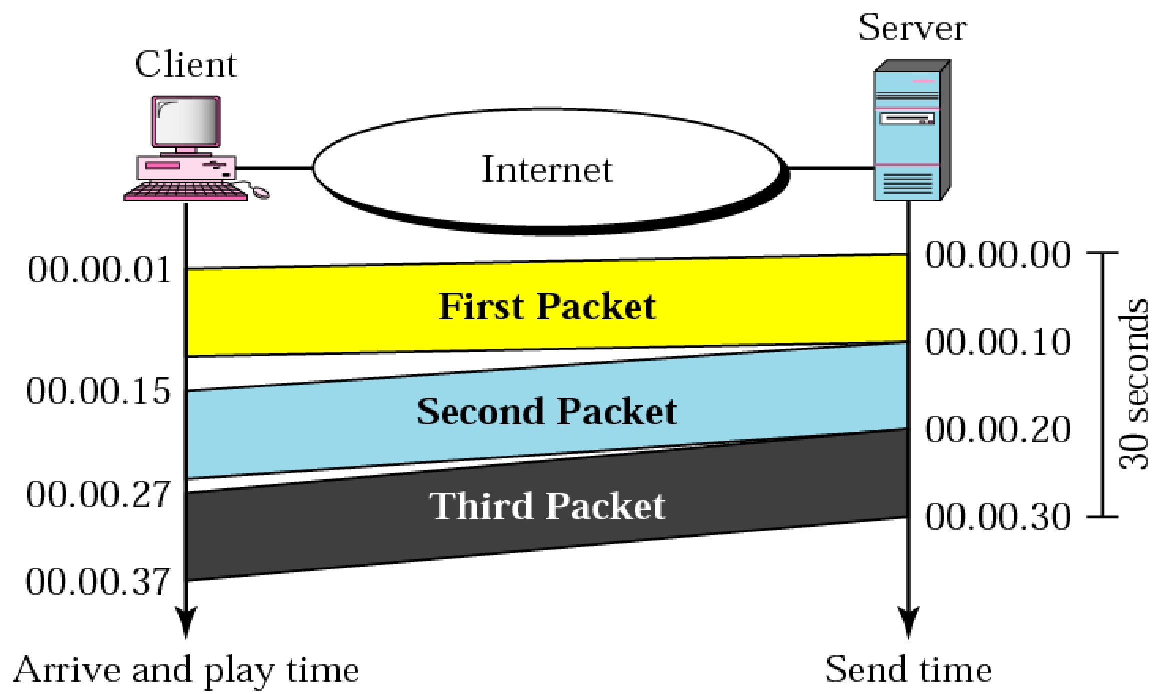
Popular Stream Media Server and Player

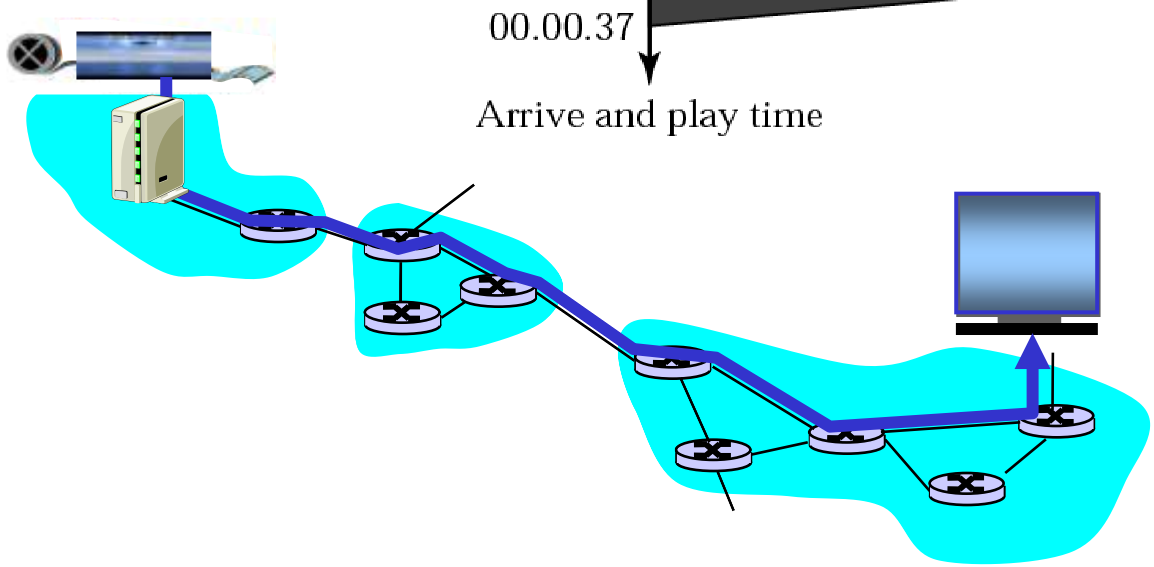


- ❑ Real Networks
  - Real Producer: create streamed media file, end with "filename.rm"
  - Real Server: streaming media to delivery across network
  - Real Player: streamed media player in RM format
- ❑ Windows Multimedia Technologies
  - Media Encoder: create streamed media file, end with "filename.asf/.wmv"
  - Media Server: streaming media to delivery across network
  - Media Player: streamed media player in ASF/WMV format
- ❑ QuickTime
  - QuickTime Pro: create streamed media file, end with "filename.qt"
  - QuickTime Streaming Server (Mac) and Darwin Streaming Server
  - QuickTime Player: streamed media player in QT format
- ❑ Audio/MP3: Liquid Audio, SHOUTcast, icecast

Key Points in Streaming Media Service

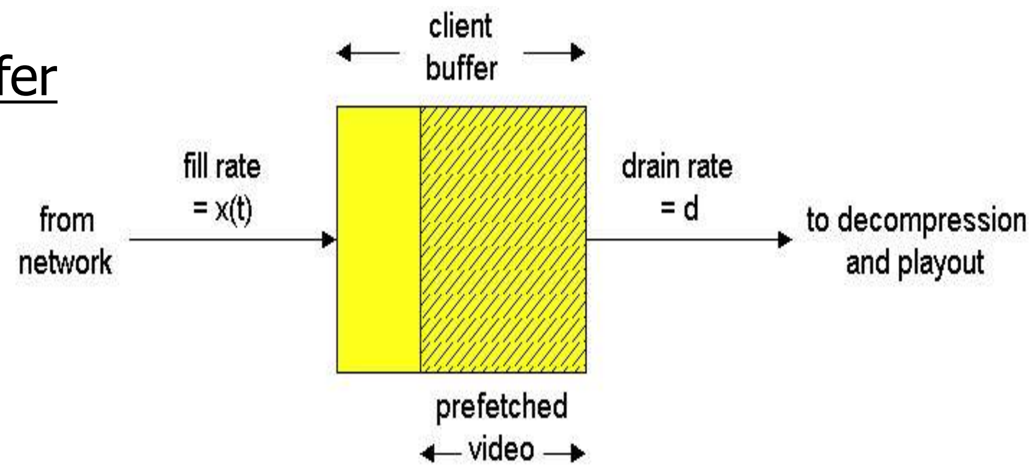
- Delay and Jitter

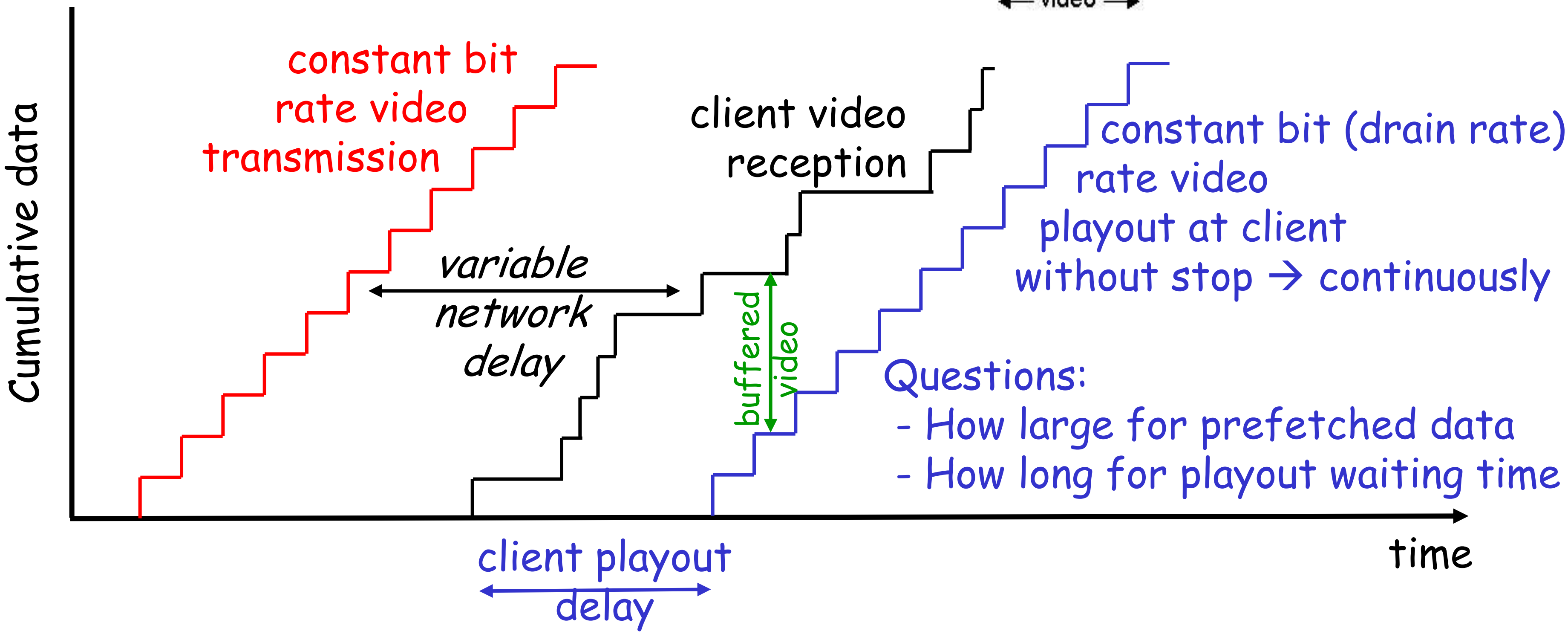




Key Points in Streaming Media Service (Cont)

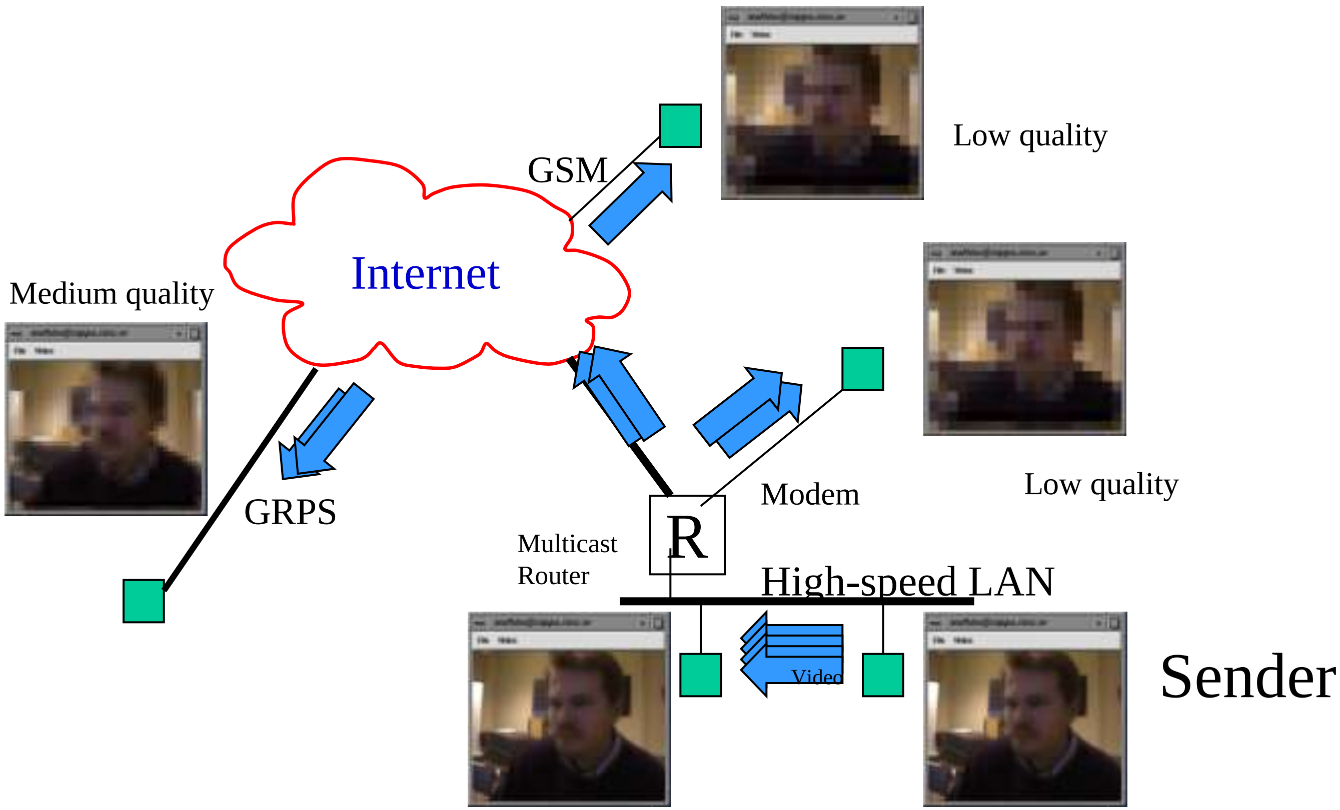
- Smooth Dealy & Jitter via buffer
  * Client-side buffering,
  * Playout delay,
  * Compensate for network delay & jitter





Key Points in Streaming Media Service (Cont)

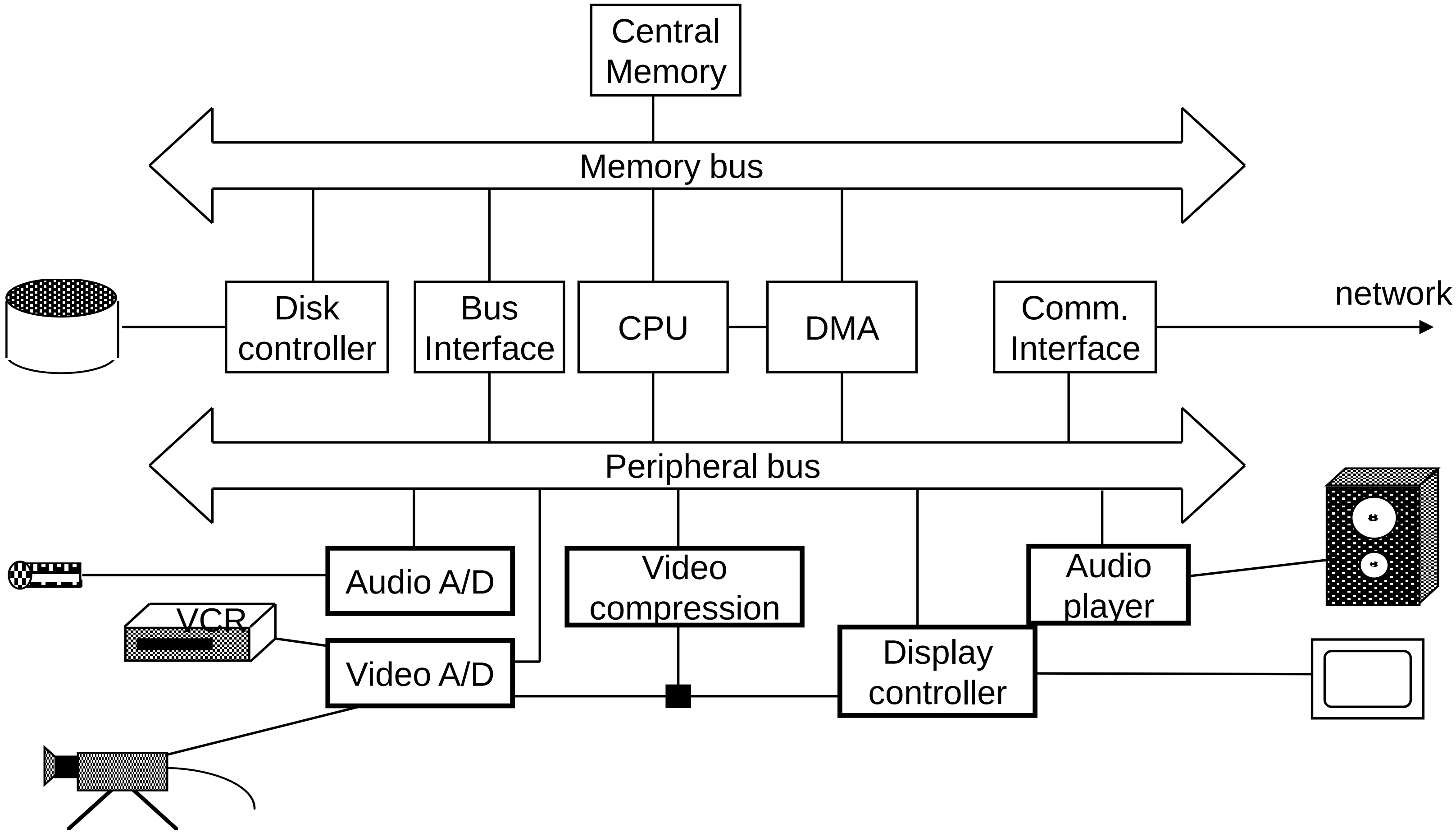
- <u>Trade-off between media quality and network bandwidth</u>
  - Data amount of continuous media, especially video, is extremely large
  - Current Internet bandwidth is relative small, 28K/56K modem, ADSL, Cable, LAN, etc.
  - Before delivery, clarify targeted users and their available bandwidth



Key Points in Streaming Media Service (cont)

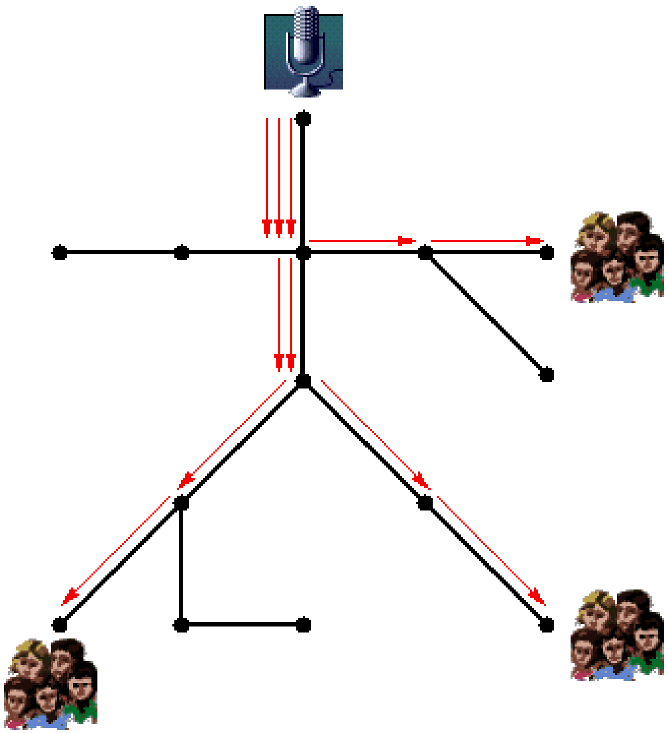
- <u>Limited server resource</u>
  - Limited computational power in processing many media steams
  - Limited storage space in saving many media data in server
  - Limited IO performance in outputting many streams to networks

→ How to serve many users simultaneously ?

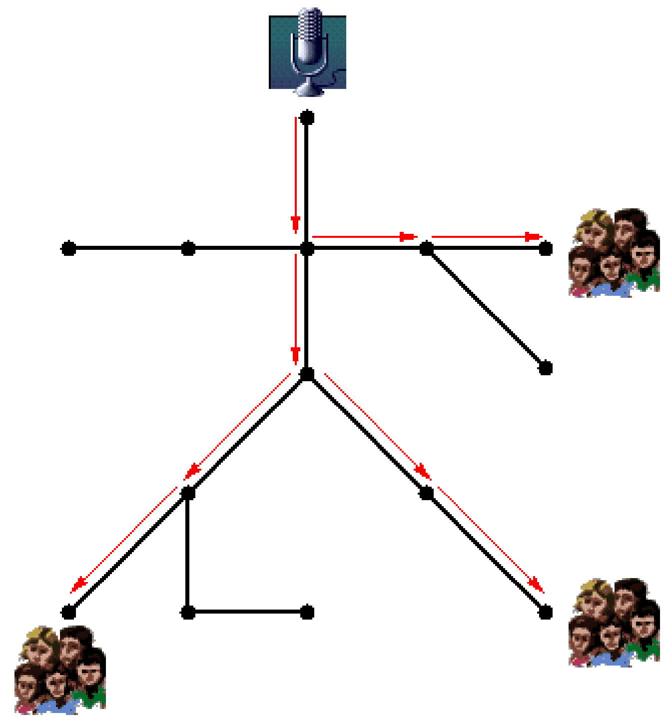


Key Points in Streaming Media Service (Cont)

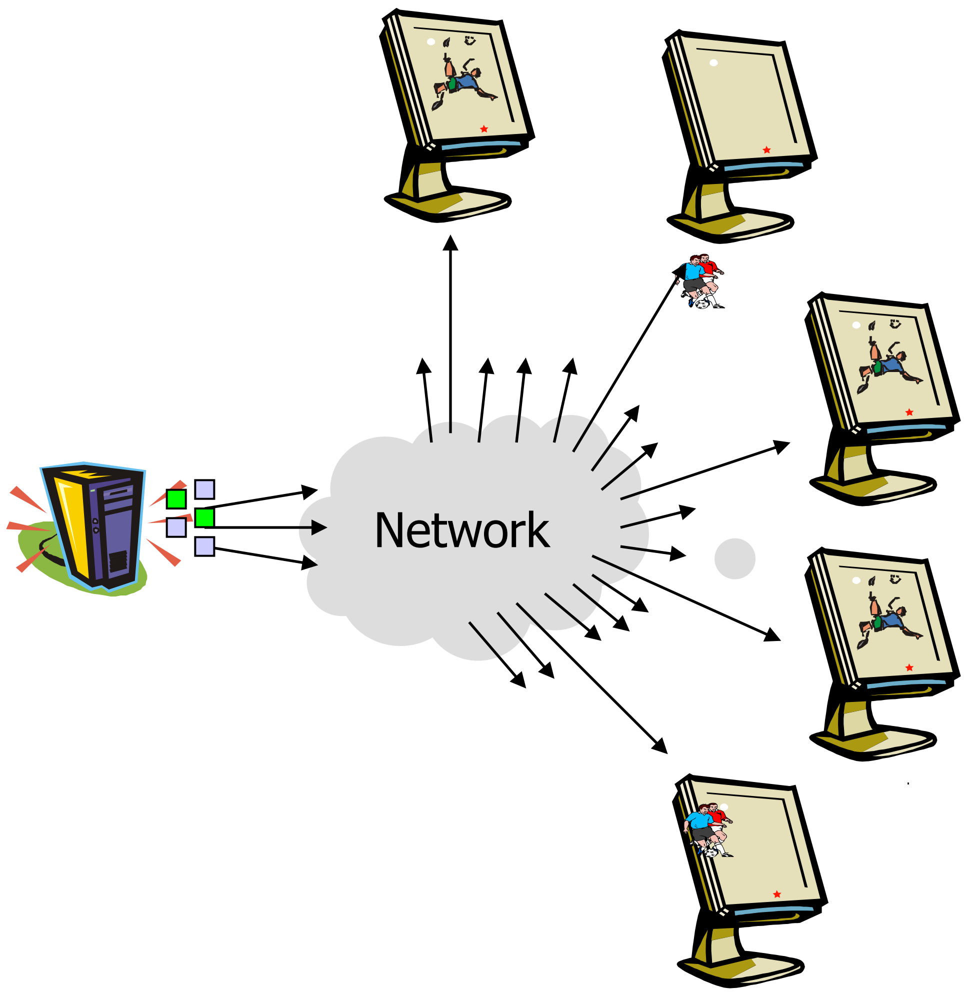
- Unicast



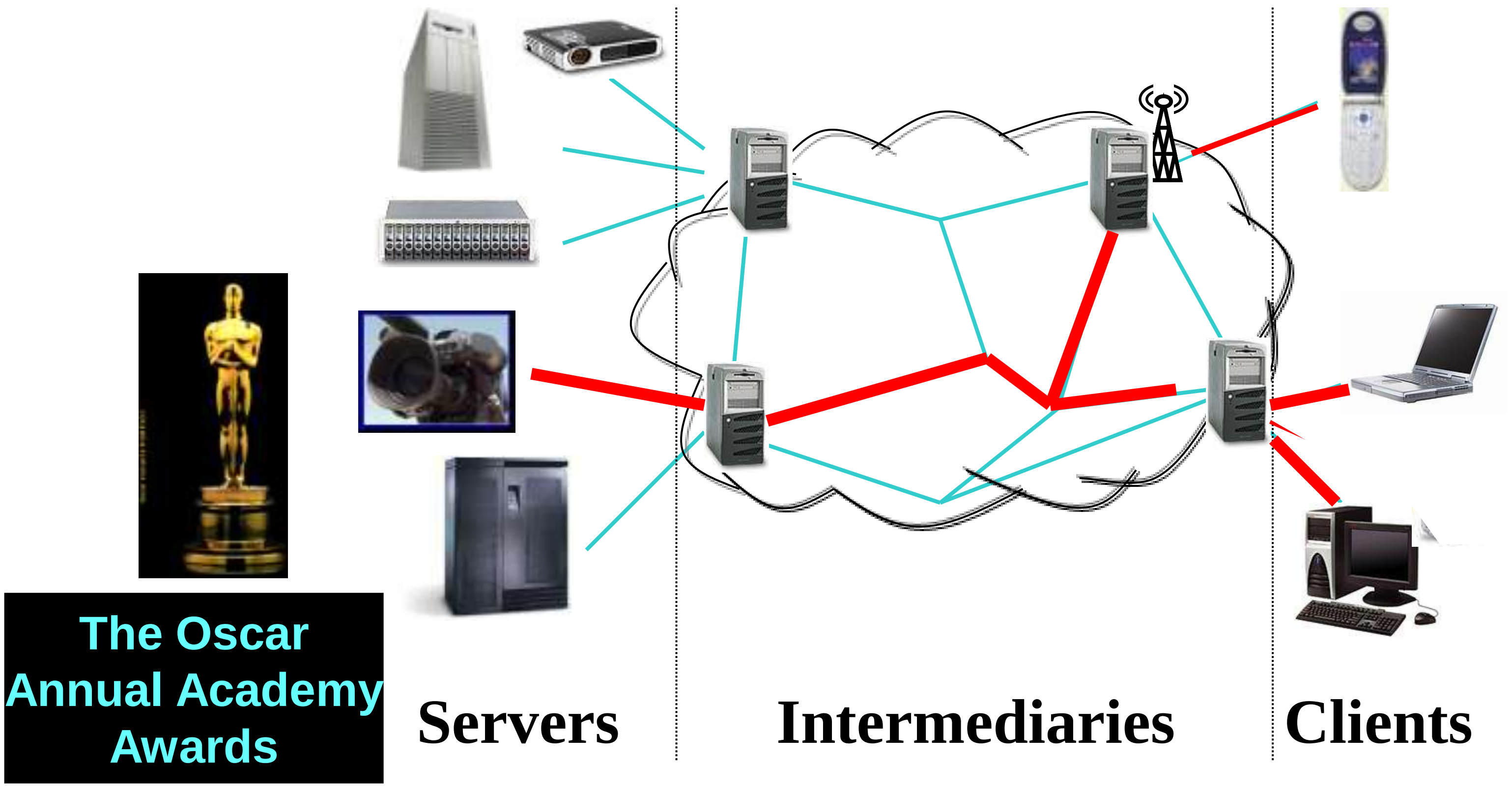
- Multicast



Unicast Example: Multiple Independent Streams

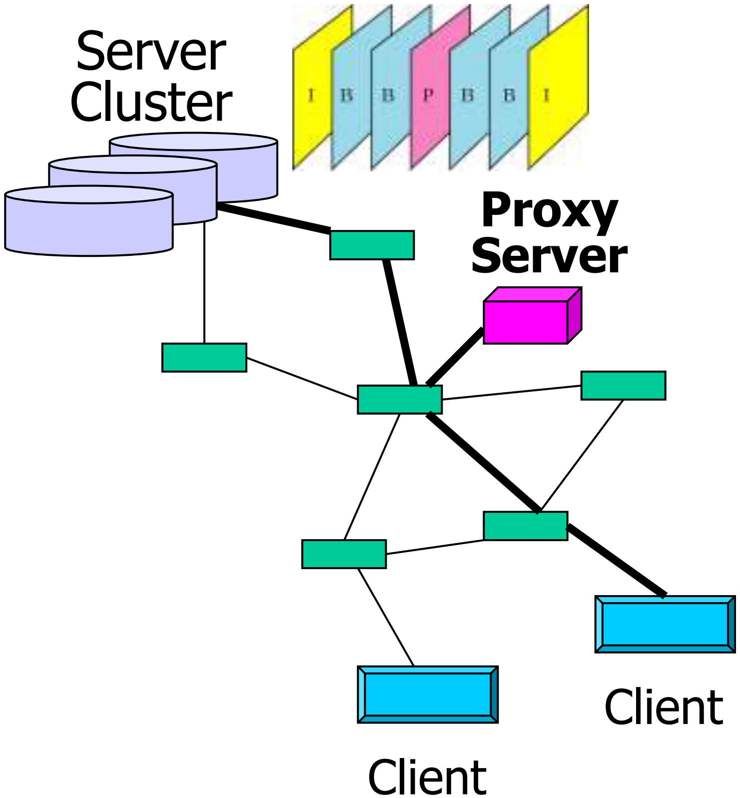


Multicast Example: Single Stream and Copy



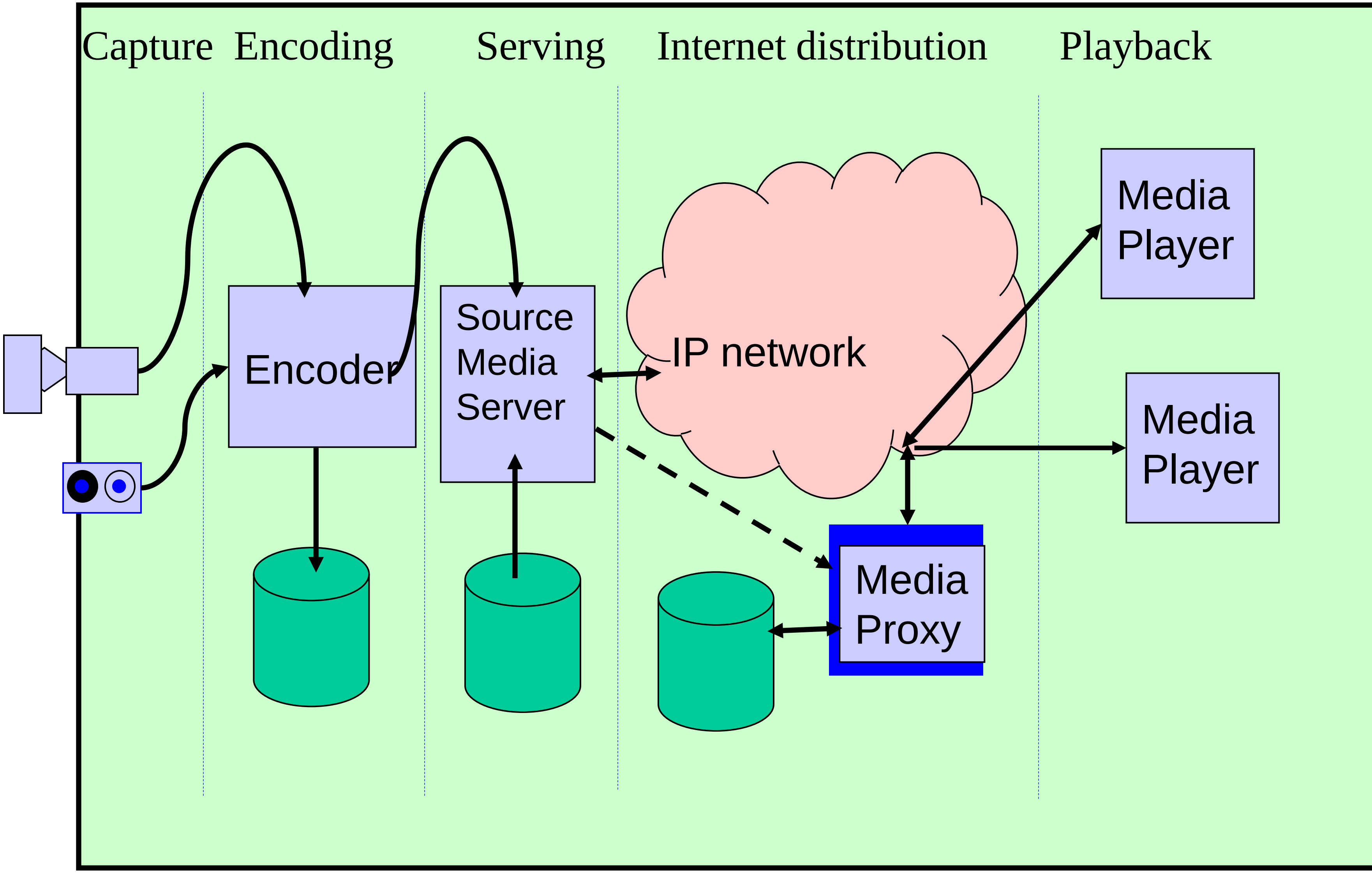
Key Points in Streaming Media Service (Cont)

- Cache technology
  - Increase IO via putting media data in memory
  - The larger memory, the better
- Distributed server cluster and proxy media server
  - Use a group of servers to improve processing performance
  - Use proxy server to reduce number of users' direct accesses to server

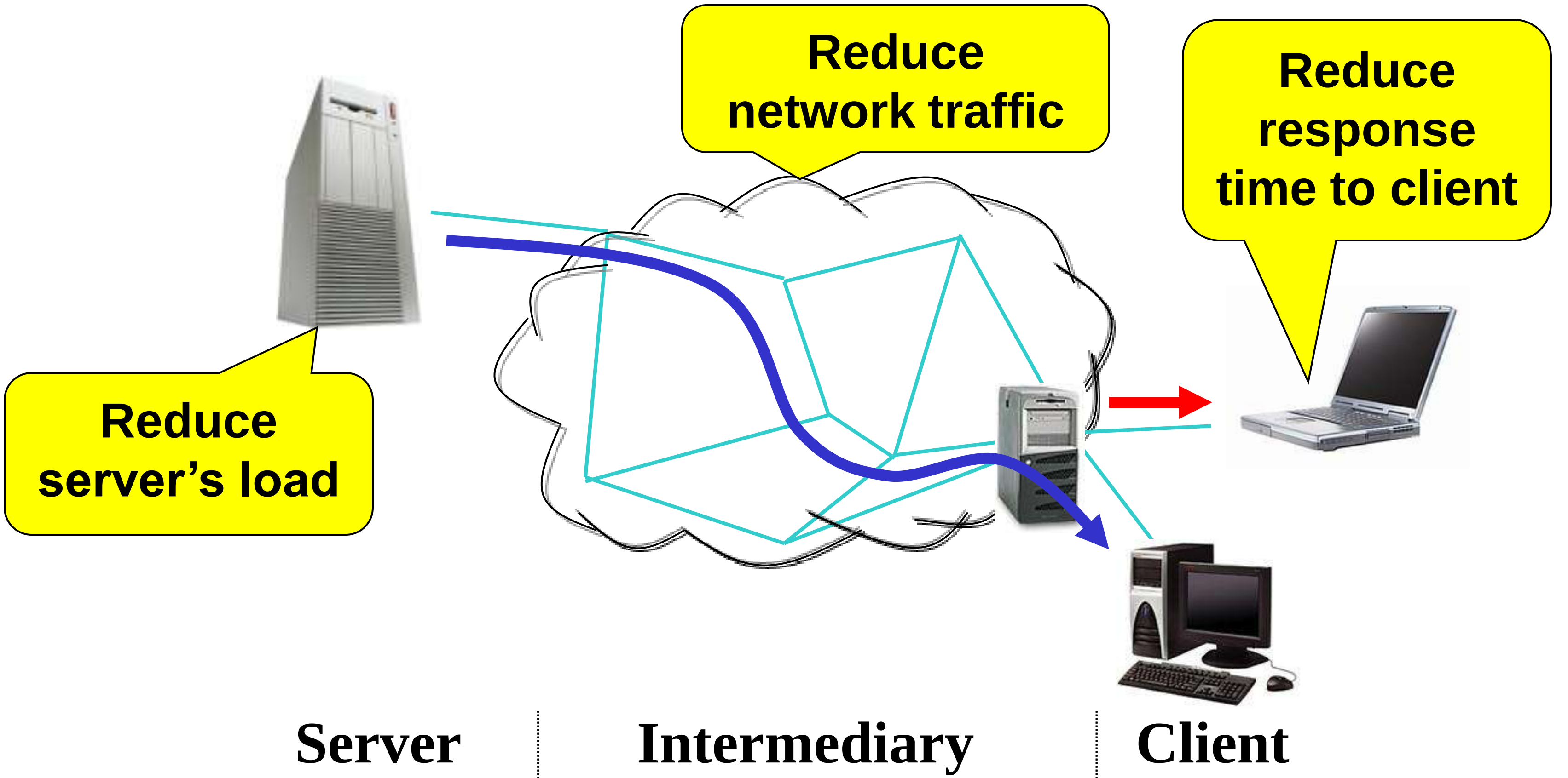


- Drop frames
  - Drop B,P frames if not enough bandwidth
- Quality Adaptation
  - Transcoding
    - Change quantization value
    - Change coding rate
- Video staging, caching, patching
  - Staging: store partial frames in proxy
  - Prefix caching: store first few minutes of movie
  - Patching: multiple users use same video

Proxy Media Server



Proxy Server: Reduce Traffic, Time, Load

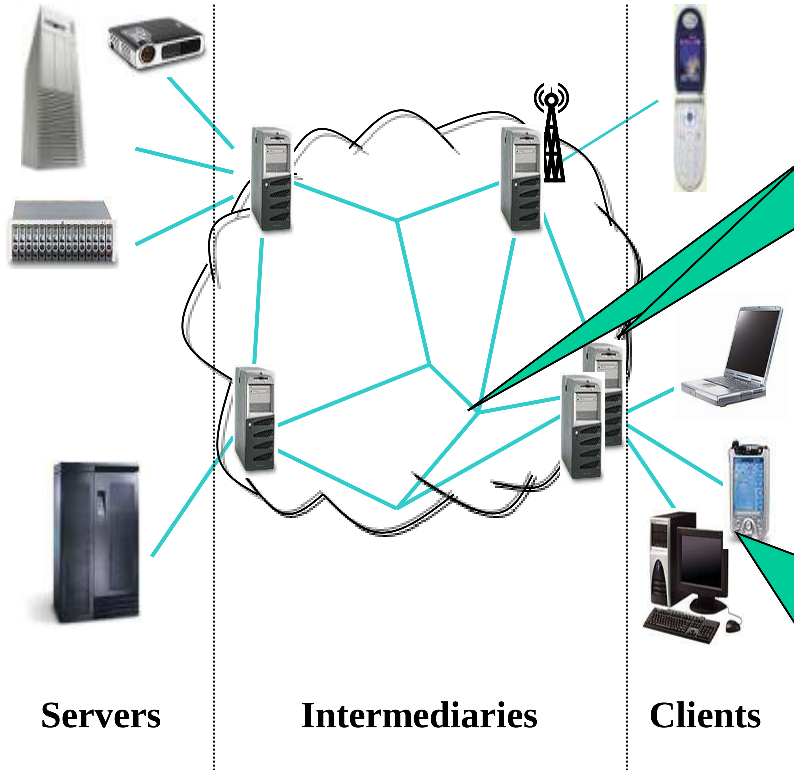


Distributed Proxy Servers

Very large sizes

Media Objects

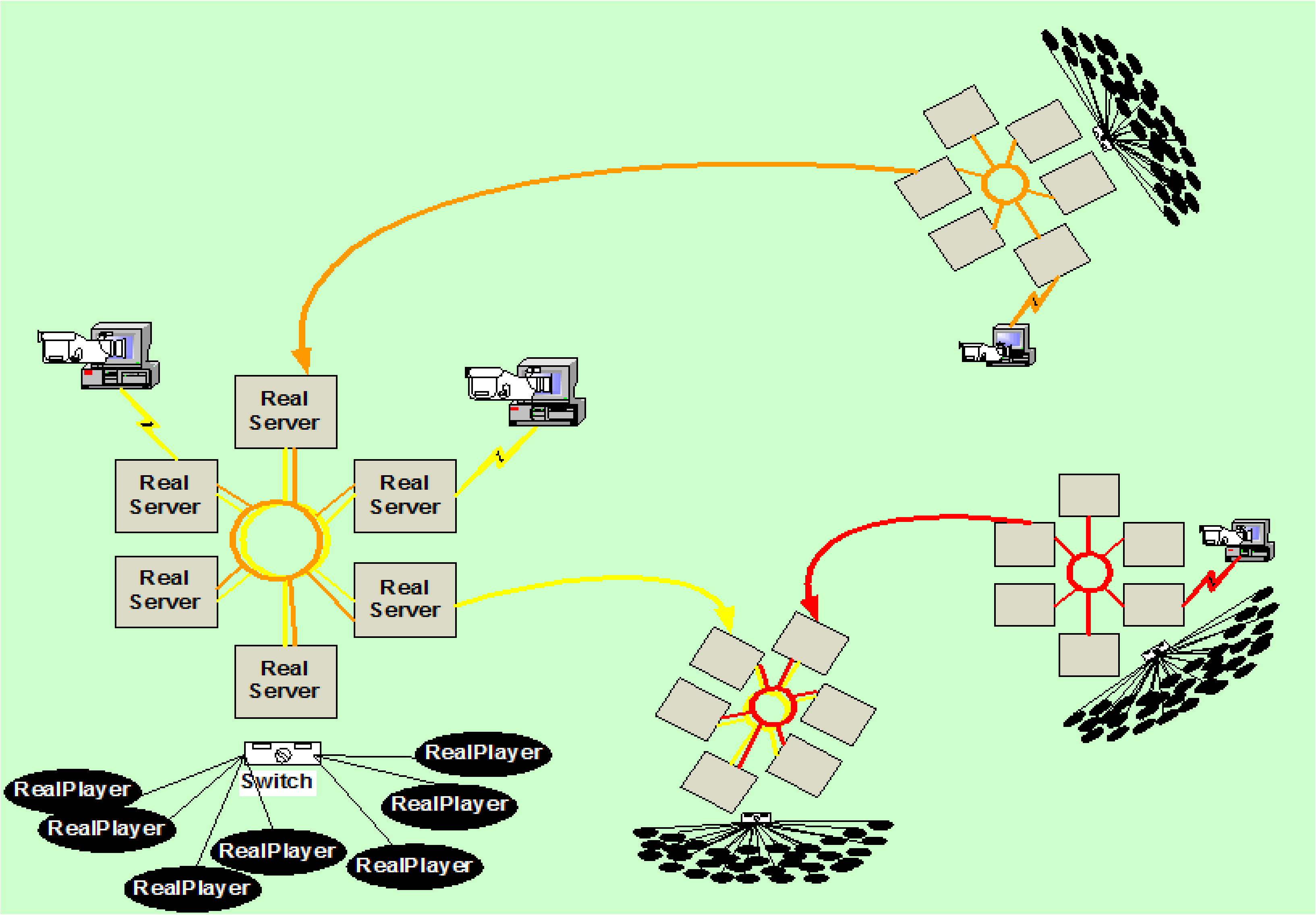
Very rigorous real-time delivery constrains: small startup latency, continuous delivery



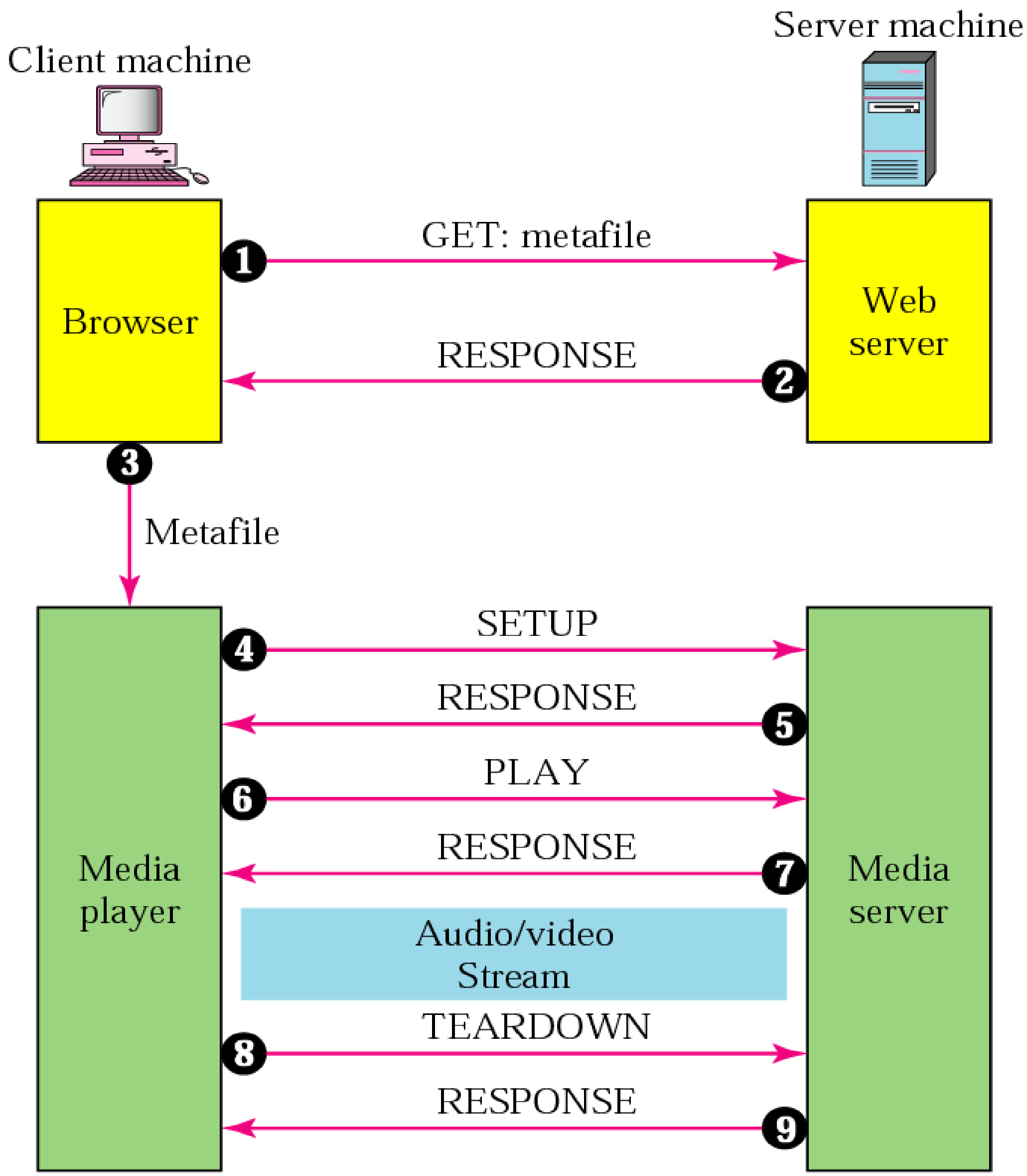
A large number of proxies with: disk, memory, and CPU cycles

Diverse client access devices: computers, PDAs, cell-phones

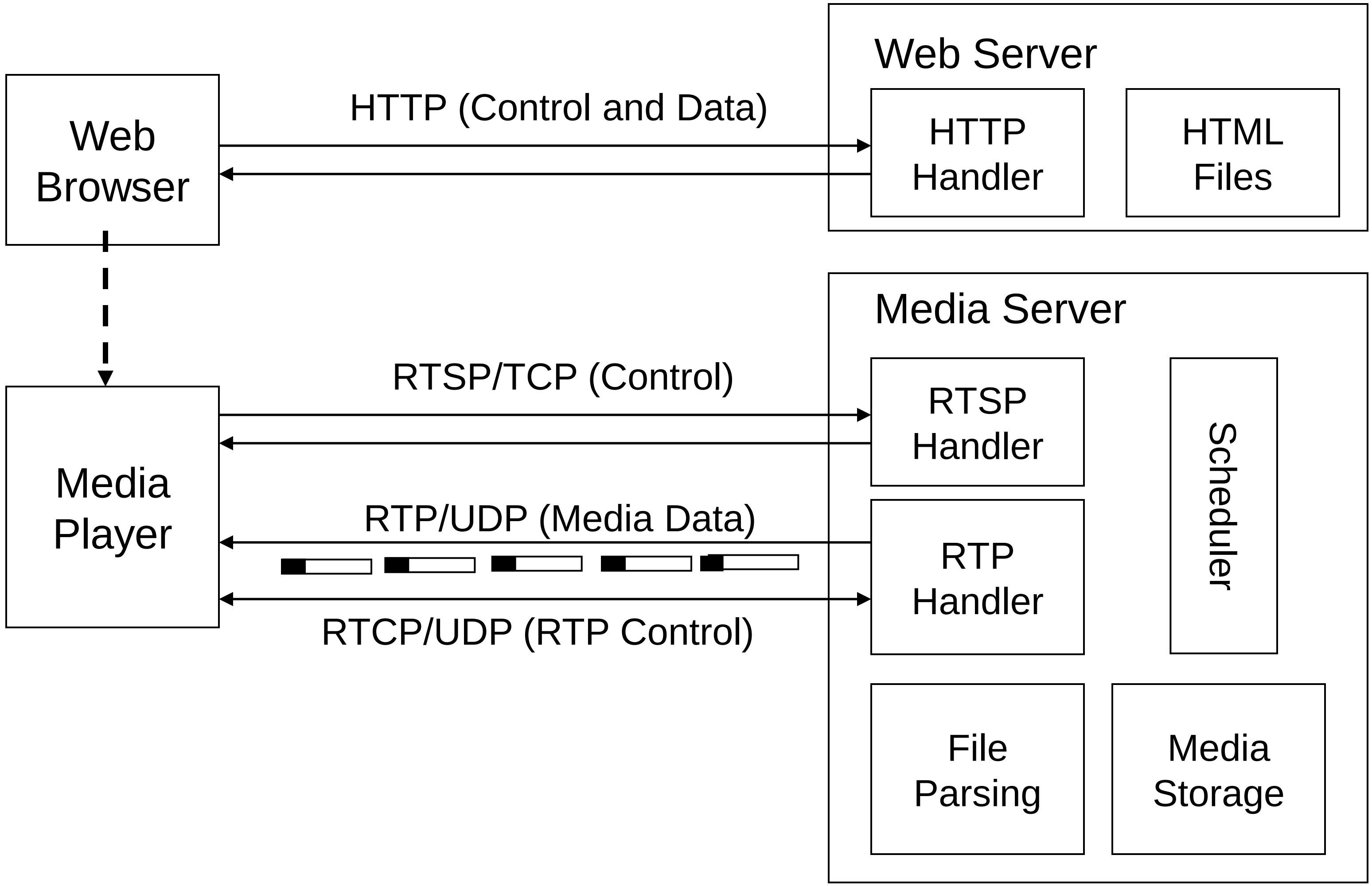
Distributed Server Clustering



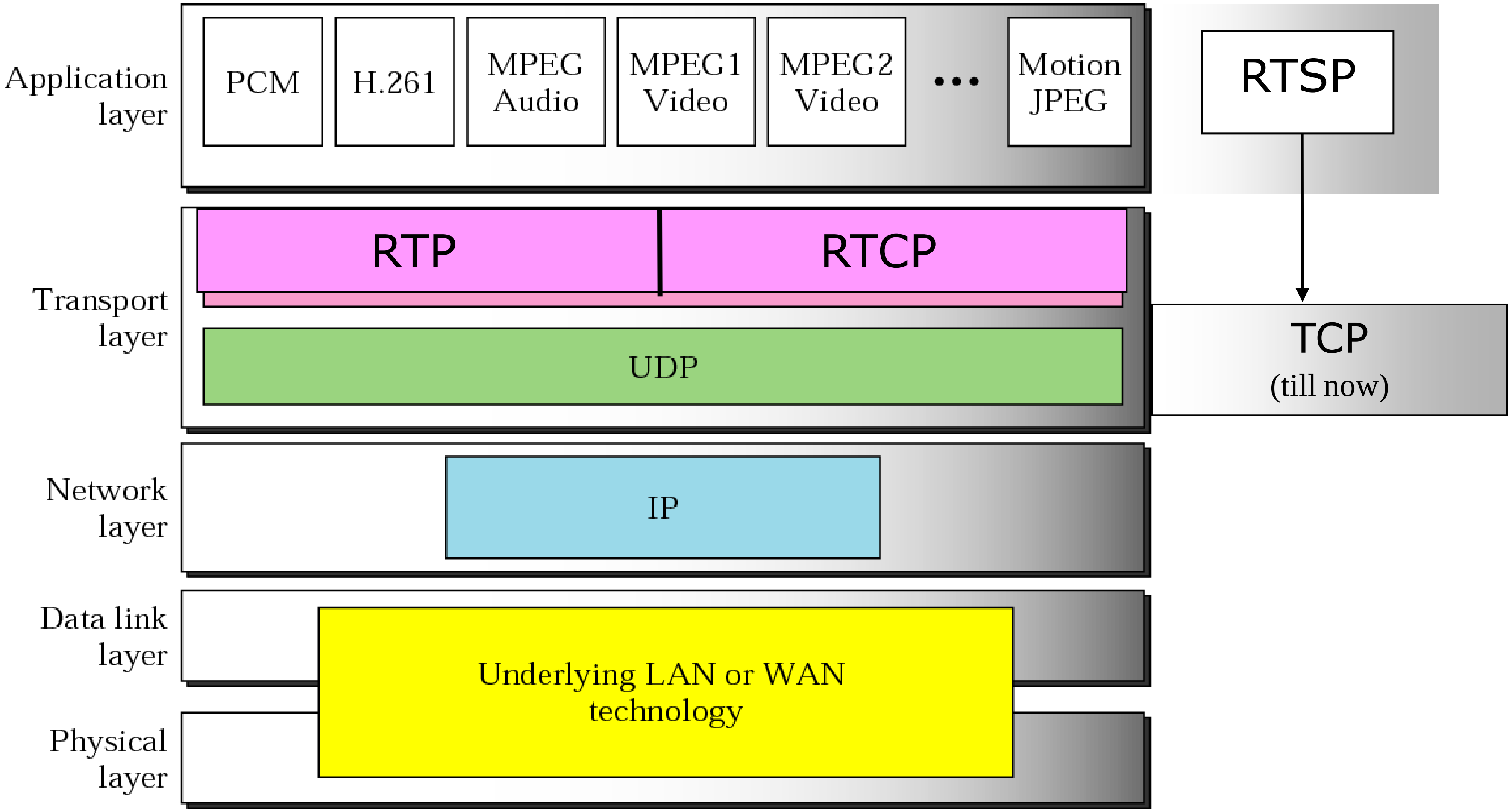
Media Streaming Service Access Process



Media Streaming Service Modules



Protocol Stack for Multimedia Services



What is RTSP?

- Real-Time Streaming Protocol (RTSP) is a standard defined in RFC 2326 by IETF in 1998
- RTSP is a control protocol intended for:
  - retrieval of media from a media server
  - establishment of one or more synchronized, continuous-media streams
  - control of such streams
- RTSP can be seen as a “network remote control”
- RTSP is not used to deliver the streams
  - use RTP or similar for that

Differences between RTSP and HTTP

- The RTSP design is based on HTTP, with the following differences:
  - new methods; different protocol identifier:

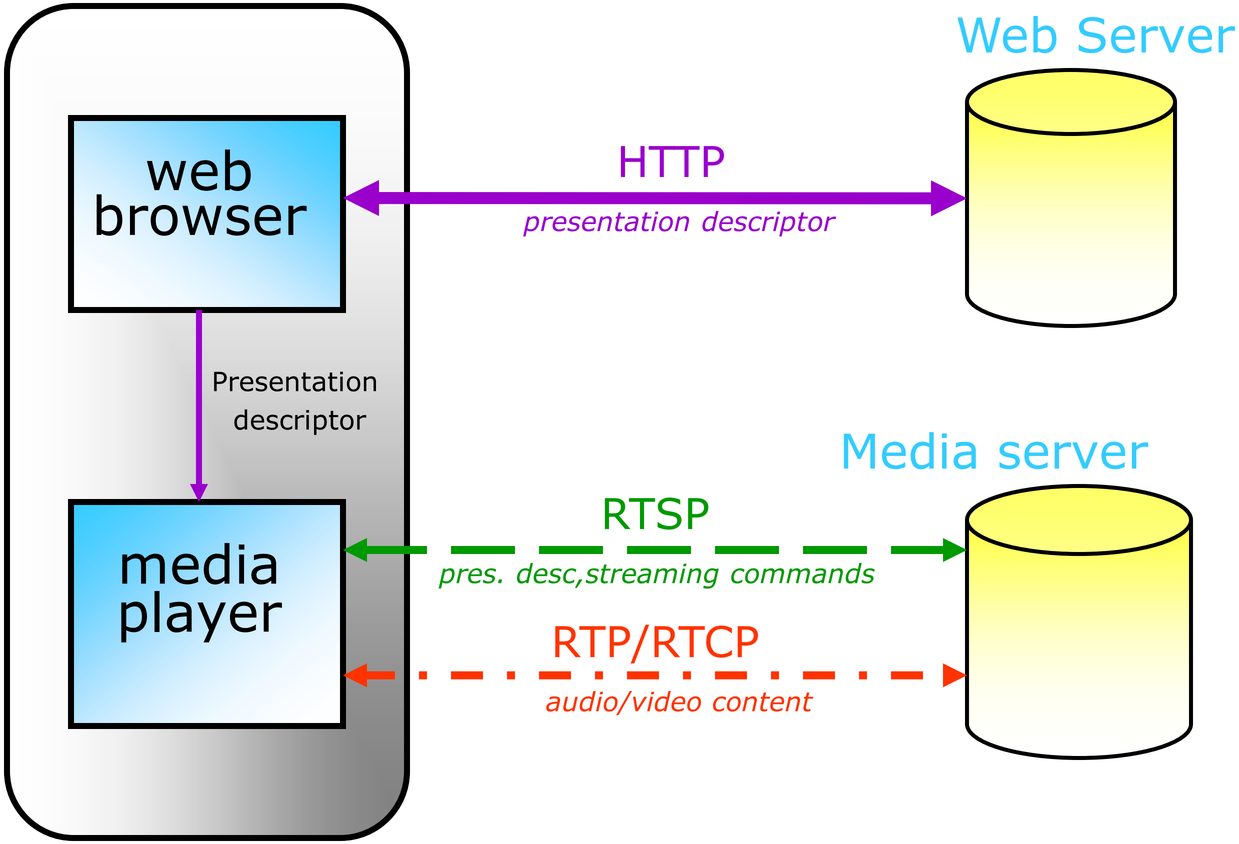
    `rtsp://audio.example.com/twister/audio.en`

    `rtsp://video.example.com/twister/video`
  - RTSP servers need to keep state while HTTP servers do not
  - Both RTSP servers and clients can issue requests
  - Data is carried by an external protocol (typically but not necessarily RTP)
  - RTSP uses UTF-8 instead of ISO 8859-1 character set
  - RTSP uses absolute request URIs
  - RTSP defines an extension mechanism

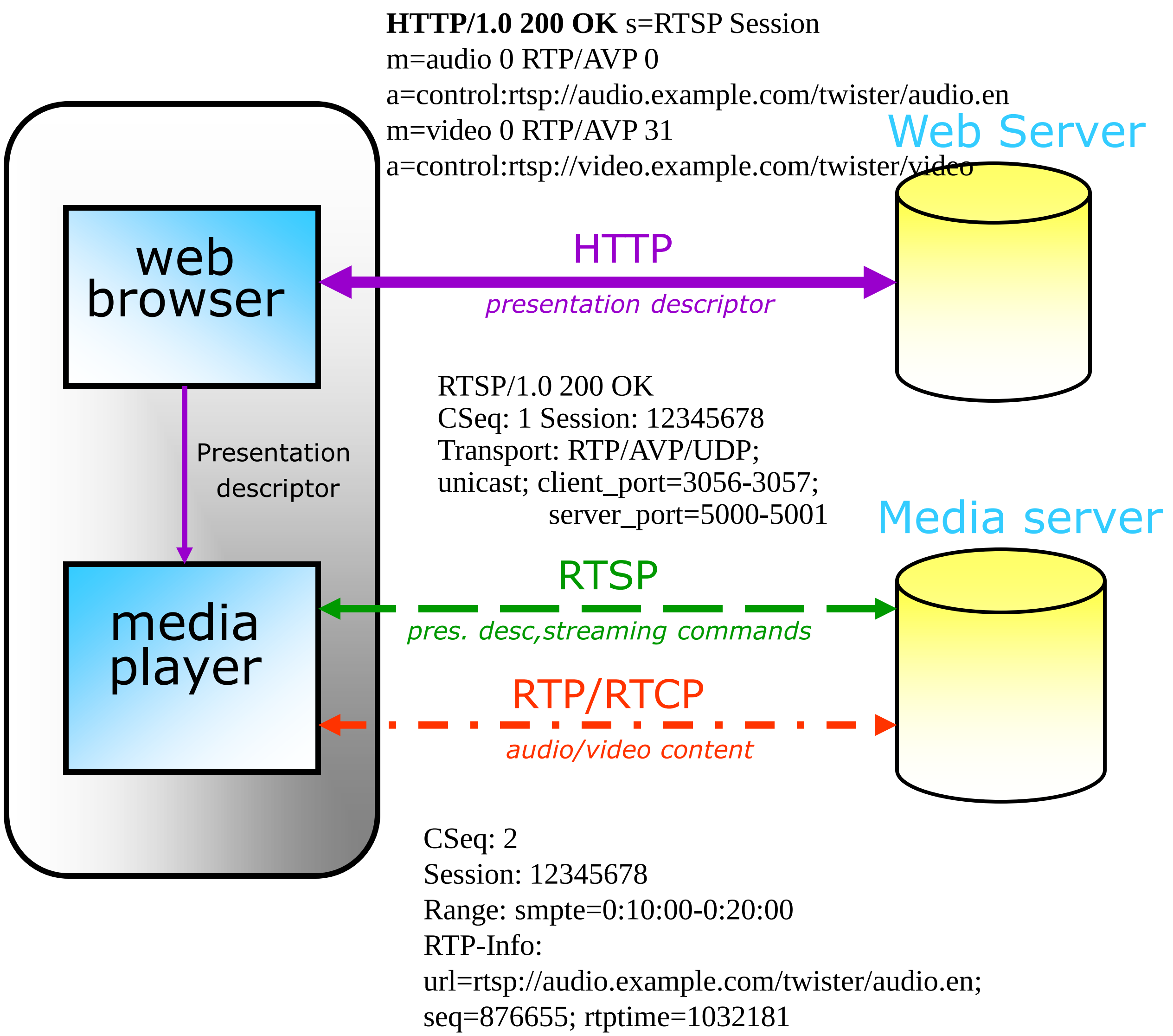
Transport independent: RTSP implements application-layer reliability and can run on top of TCP, UDP, or any other protocol. Standardized ports for RTSP:

```
rtsp          554/tcp    Real Time Streaming Control
rtsp          554/udp    Real Time Streaming Control
rtsp-alt      8554/tcp   RTSP Alternate
rtsp-alt      8554/udp   RTSP Alternate
```

HTTP and RTSP



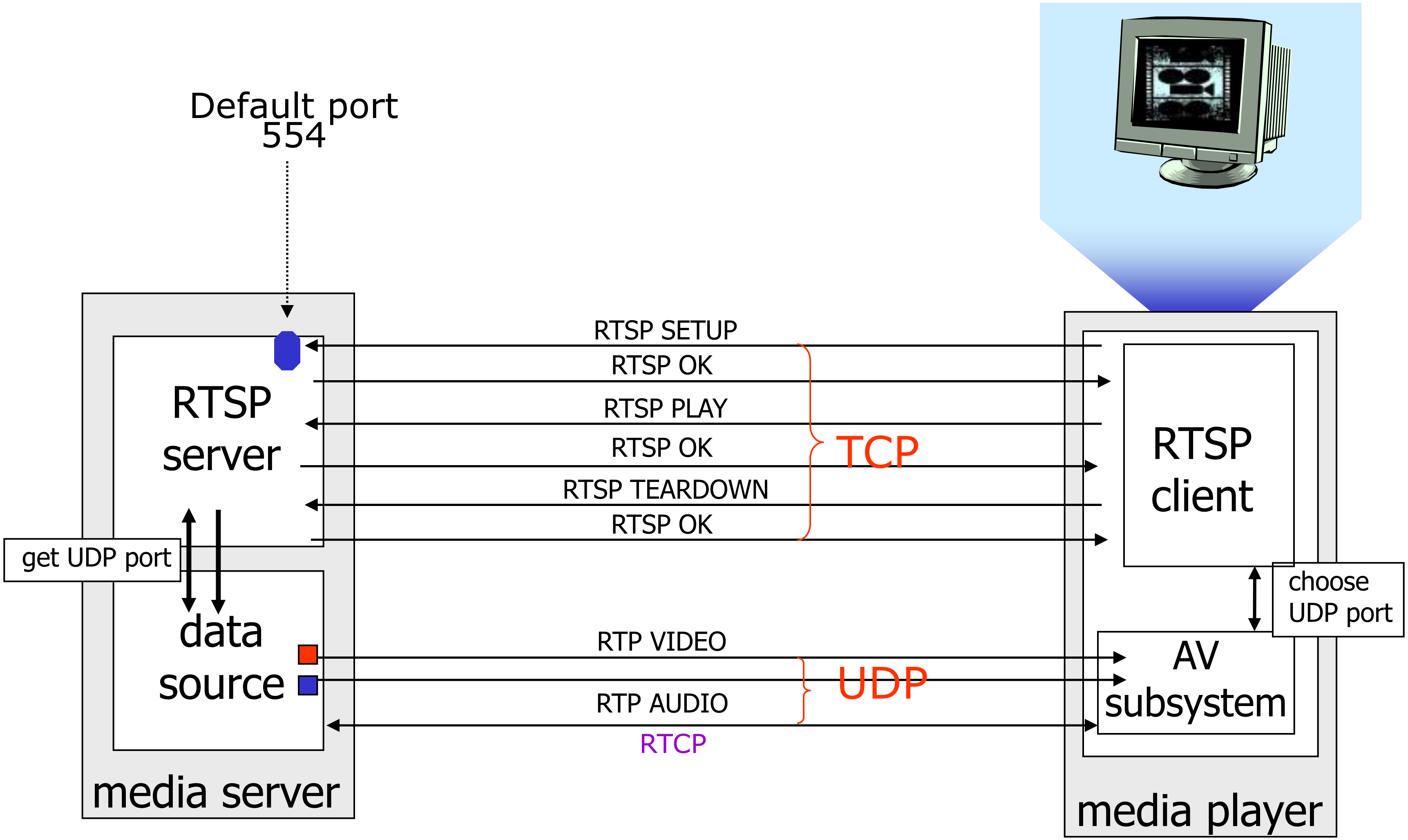
HTTP and RTSP



RTSP Methods

OPTIONS	$C \rightarrow S$ $C \leftarrow S$	determine capabilities of server/client
DESCRIBE	$C \rightarrow S$	get description of media stream
ANNOUNCE	$C \leftrightarrow S$	announce new session description
SETUP	$C \rightarrow S$	create media session
RECORD	$C \rightarrow S$	start media recording
PLAY	$C \rightarrow S$	start media delivery
PAUSE	$C \rightarrow S$	pause media delivery
REDIRECT	$C \leftarrow S$	redirection to another server
TEARDOWN	$C \rightarrow S$	immediate teardown
SET_PARAMETER	$C \leftrightarrow S$	change server/client parameter
GET_PARAMETER	$C \leftrightarrow S$	read server/client parameter

RTSP Session

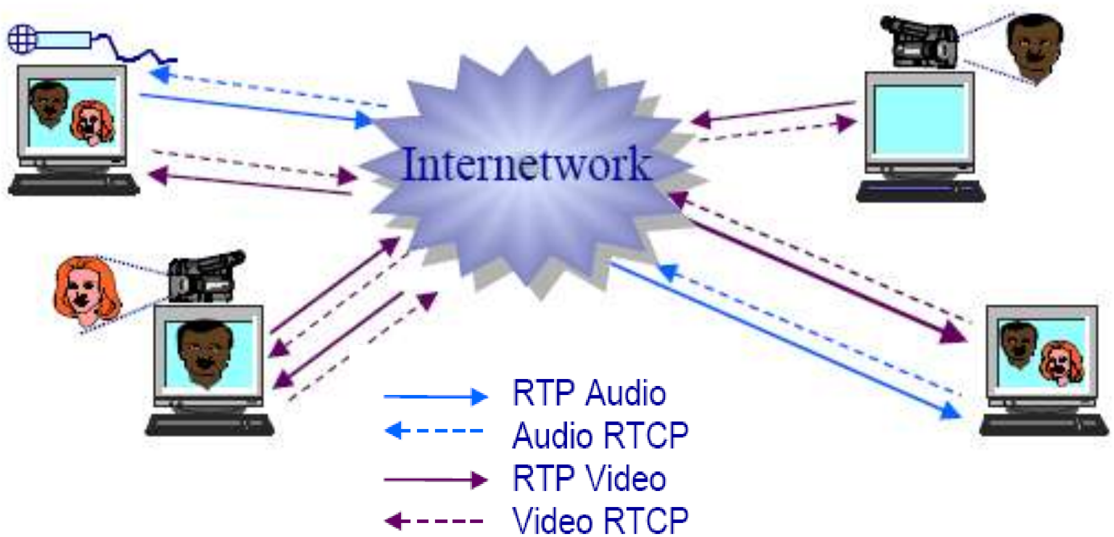


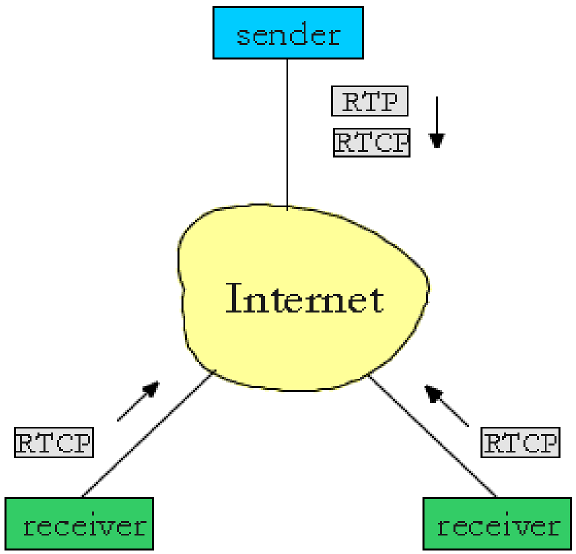
What is RTP?

- Realtime Transport Protocol (RTP) is an IETF standard
- Primary objective: stream continuous media over a best-effort packet-switched network in an interoperable way.
- Protocol requirements:
  - Payload Type Identification: what kind of media are we streaming?
  - Sequence Numbering: to deal with lost and out-of-order packets.
  - Timestamping: to compensate for network jitter in packet delivery.
  - Delivery Monitoring: how well is the stream being received by the destinations?
- RTP does not guarantee QoS (Quality of Service), i.e., reliable, on-time delivery of the packets (the underlying network is expected to do that).
- RTP typically runs on top of UDP, but the use of other protocols is not precluded

RTT, RTCP and Session

- RTP is composed of two closely-linked parts:
  - The Real-Time Transport Protocol (RTP), used to carry real-time data
  - The RTP Control Protocol (RTCP), used to:
    - Monitor and report Quality of Service
    - Convey information about the participants of a session





- Two connective ports are needed for media data transmissions
  - Even number 2n for RTP and odd number 2n+1 for RTCP
- RTP defines the concept of a ***profile***, which completes the specification for a particular application:
  - Media encoding specifications, Payload format specifications

RTP Header

Sampling instant of first data octet

- 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

Payload type

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

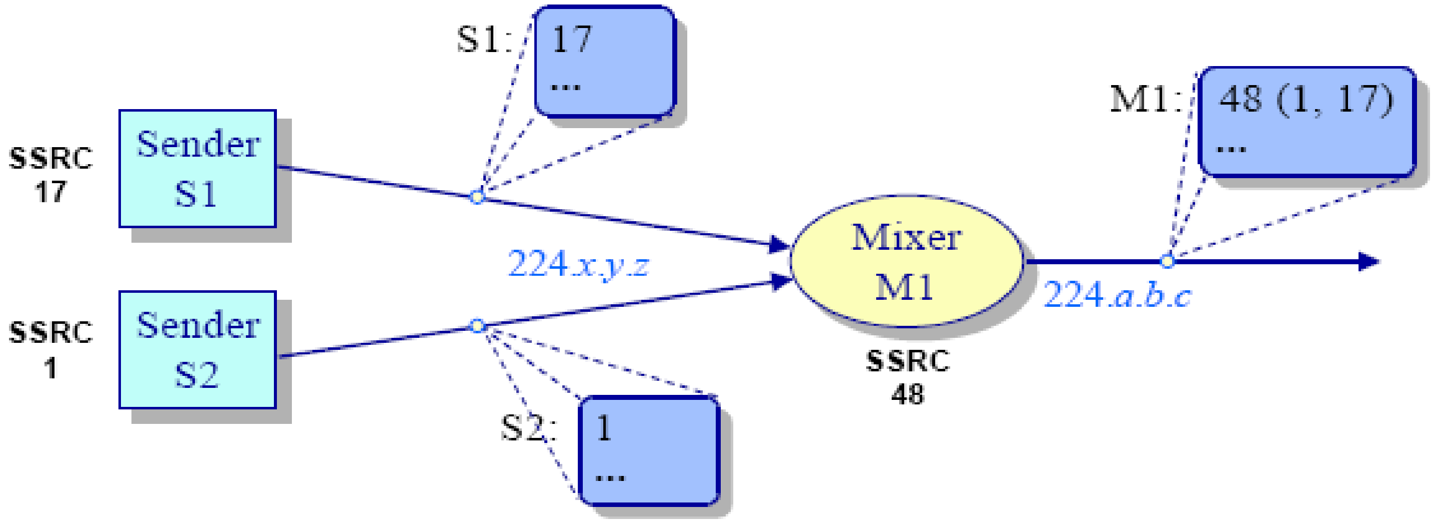
Identifies synchronization source

Identifies contributing sources

(used by mixers)

RTP Mixer

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

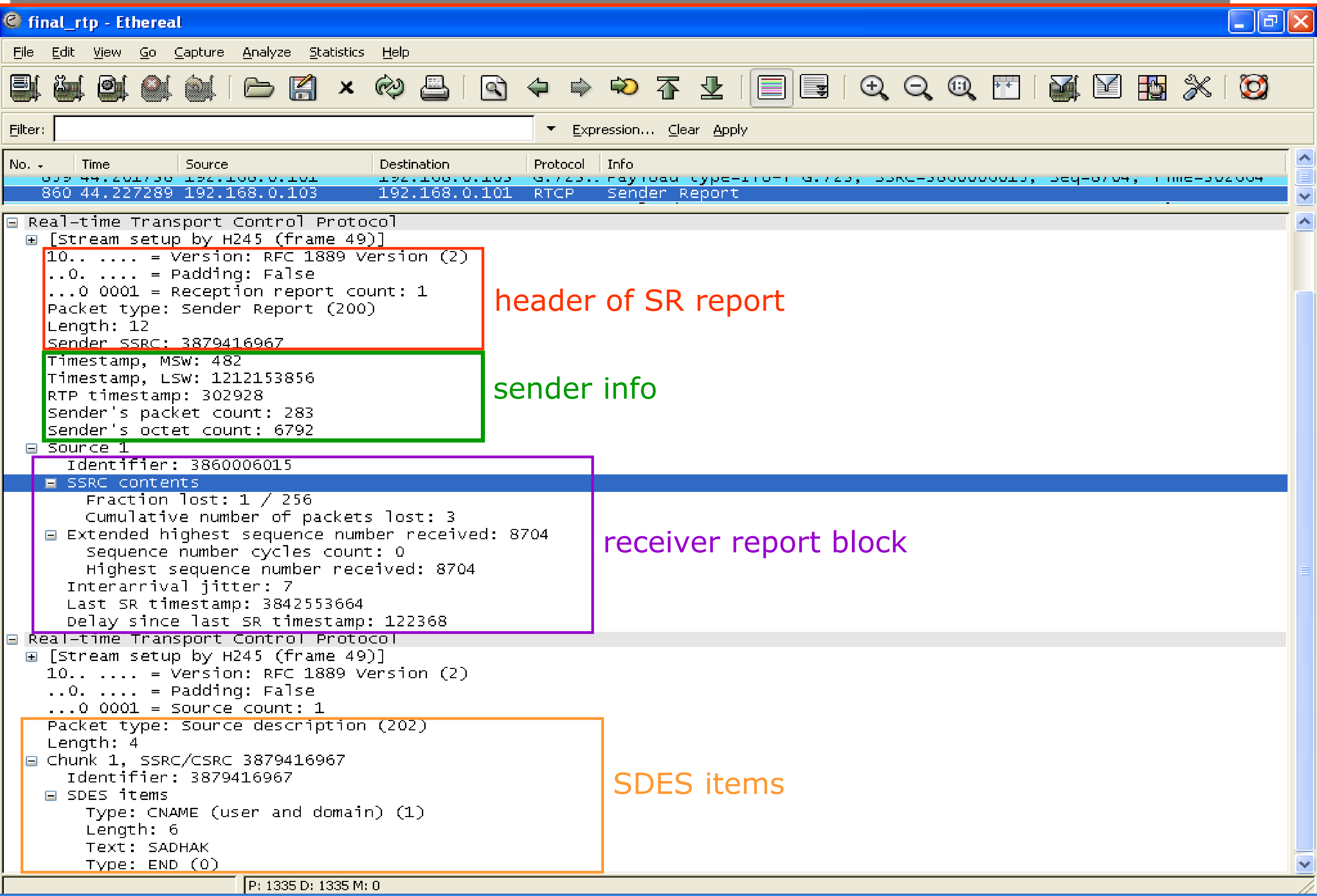


- Stream may be transcoded, special effects may be performed.
- A mixer will typically have to define synchronization relationships between streams.Thus…
    - ➢ Sources that are mixed together become contributing sources (*CSRC*)
    - ➢ Mixer itself appears as a new source having a new *SSRC*

RTCP Reports

- Cumulative counts allow both long- and short-term analysis
  - any two reports can be subtracted to get activity over an interval
  - NTP timestamps in reports allow you to compute rates
  - monitoring tools needn’t know anything about particular media encoding
- Sender reports give utilization information
  - average packet rate and average data rate over any interval
  - monitoring tools can compute this without reading any of the data
- Receiver reports give loss and round-trip information
  - extended sequence number can be used to compute packets expected
  - packets lost and packets expected give long term loss rate
  - fraction lost field gives short-term loss rate, with only a single report
  - LSR and DLSR give sender’s ability to compute round-trip time

Analyzing RTCP Reports



Demos of Streamed Audio and Video