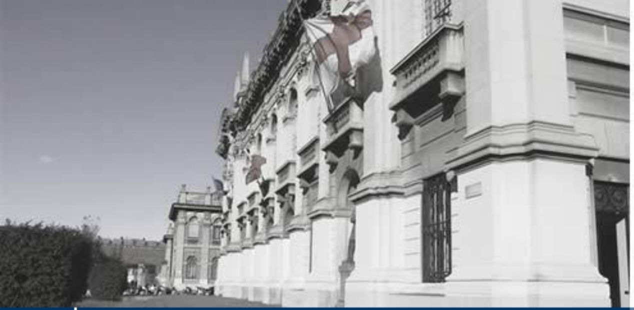


POLITECNICO DI MILANO

IPTV: Real-time Transport Protocol (RTP) and Real Time Streaming Protocol (RTSP)

OVERVIEW

The Real-Time Transport Protocol (RTP)

- An internet protocol standard
  - Specifies a way for programs to manage the real-time transmission of multimedia data over either unicast or multicast network services
  - Provides end-to-end network transport functions
  - Originally specified in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1889

The Real-Time Transport Control Protocol (RTCP)

- RTP combines its data transport with a control protocol (RTCP)
- Makes it possible to monitor data delivery for large multicast networks
  - Packet loss detection
  - Delay jitter compensation

RTP & RTCP

- Both protocols work independently of the underlying transport layer and network layer protocols

The Real-Time Streaming Protocol (RTSP)

- RTSP is a network control protocol for use in entertainment and communications systems to control streaming media servers
- establishes and controls media sessions between end points
- developed by the MMUSIC WG of the Internet Engineering Task Force (IETF) and published as RFC 2326 in 1998

IPTV COMMUNICATION MODEL (IPTVCM)

IPTVCM

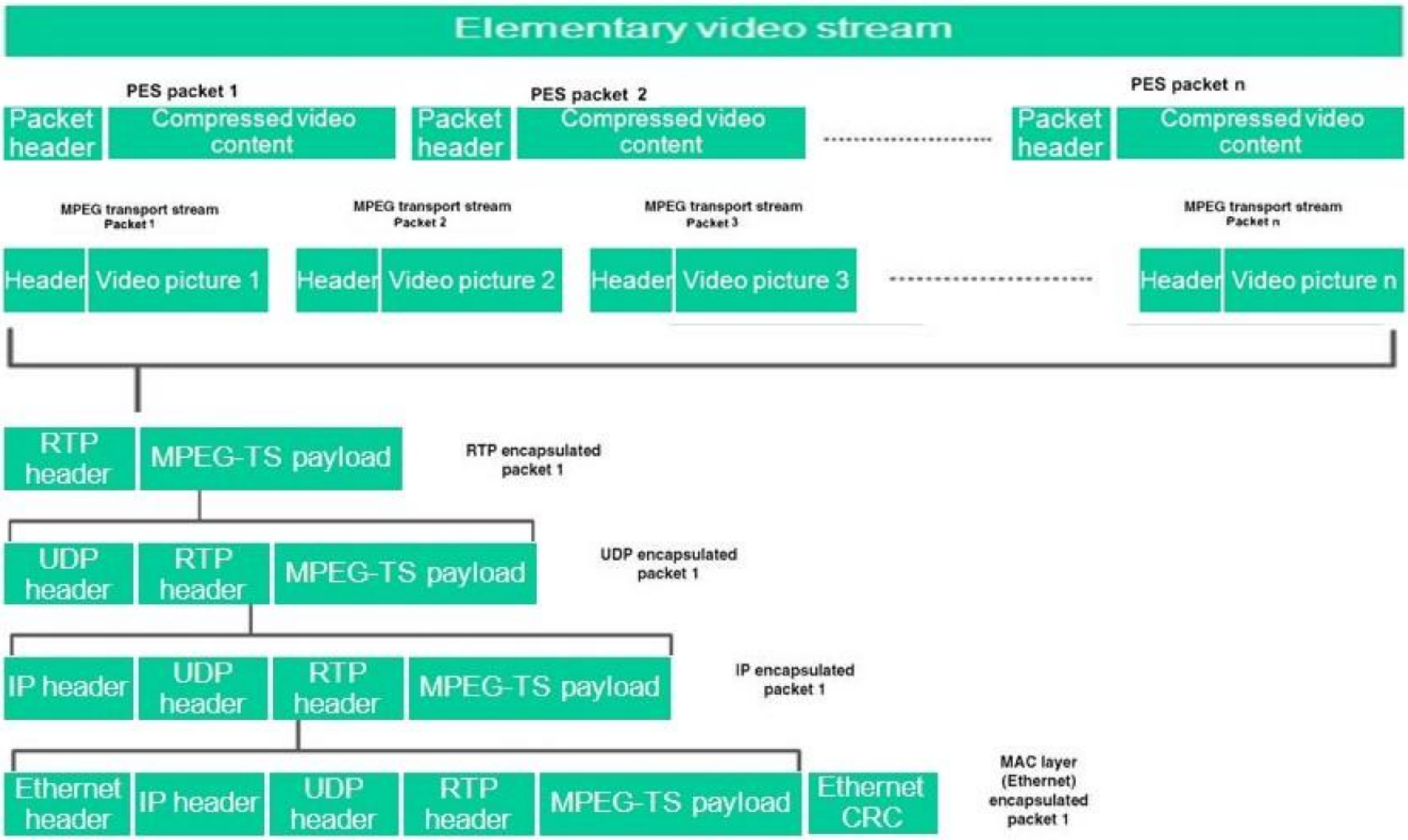
- A conceptual environment that consists of seven and potentially eight layers that are stacked on top of each other, each specifying particular network functions

- One of them is RTP layer (optional)

IPTVCM encapsulation layers (1/2)

1. Physical
2. Data link
3. IP
4. Transport
5. RTP layer (optional)
6. Transport stream construction
7. Video packetizing layer
8. Video encoding layer

IPTVCM encapsulation layers (2/2)



RTP Layer

- Used by a wide variety of IPTV applications
- Intermediary between the H.264/AVC, MPEG-2, or VC-1 encoded content in the higher layers and the lower sections of the IPTVCM
- The RTP protocol represents the core of this layer

Video Delivery

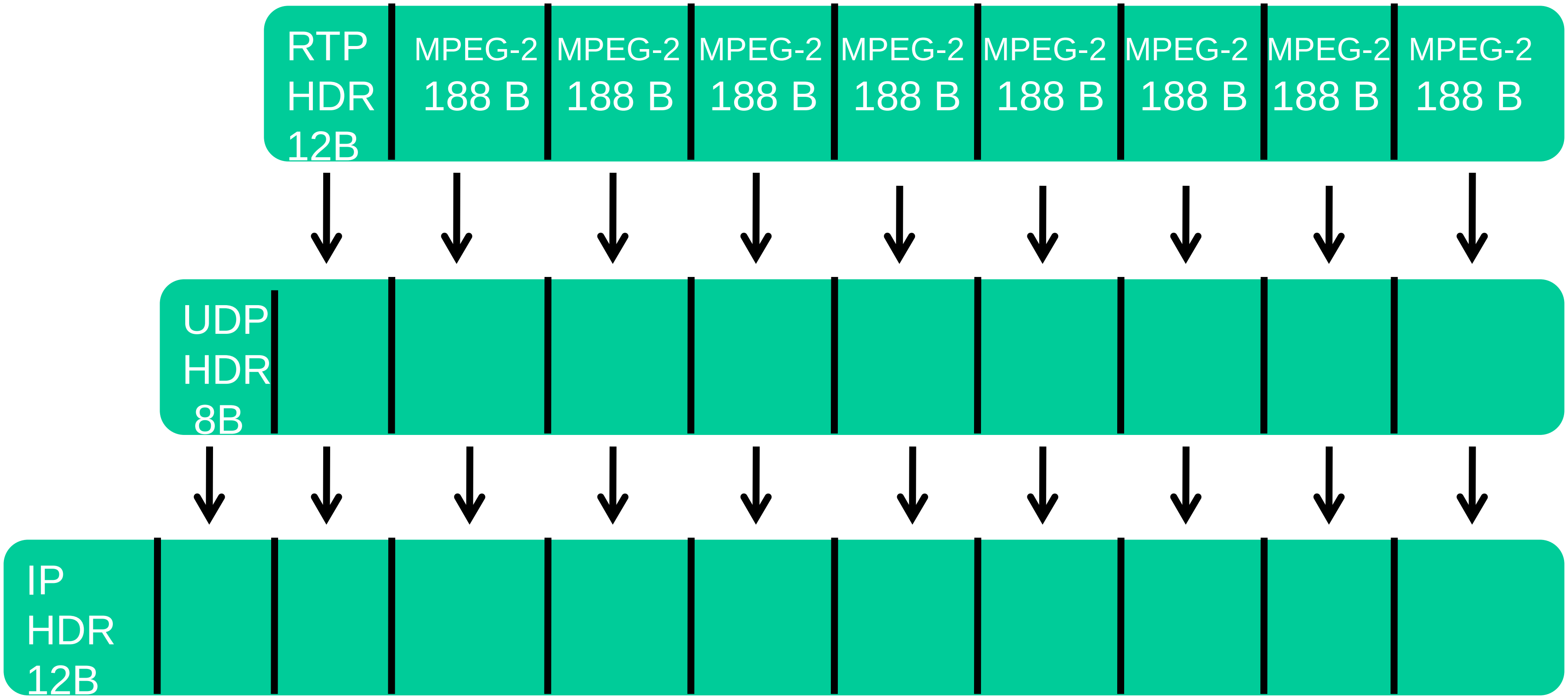
- Transporting MPEG-2-based IPTV is usually done by the use of RTP
  - UDP/RTP
- It is important to note that video can also be transported directly in UDP packets without the use of RTP
  - UDP/RAW

UDP/RTP

- detection of several additional error conditions beyond those detectable when UDP/RAW is used:
  - Determining packets received out of order
  - Detecting duplicate packets
  - Determining if a packet is lost
  - Determining packets that have an incorrect size

RTP - TECHNICAL ARCHITECTURE

The use of RTP for transporting MPEG-2-based IPTV



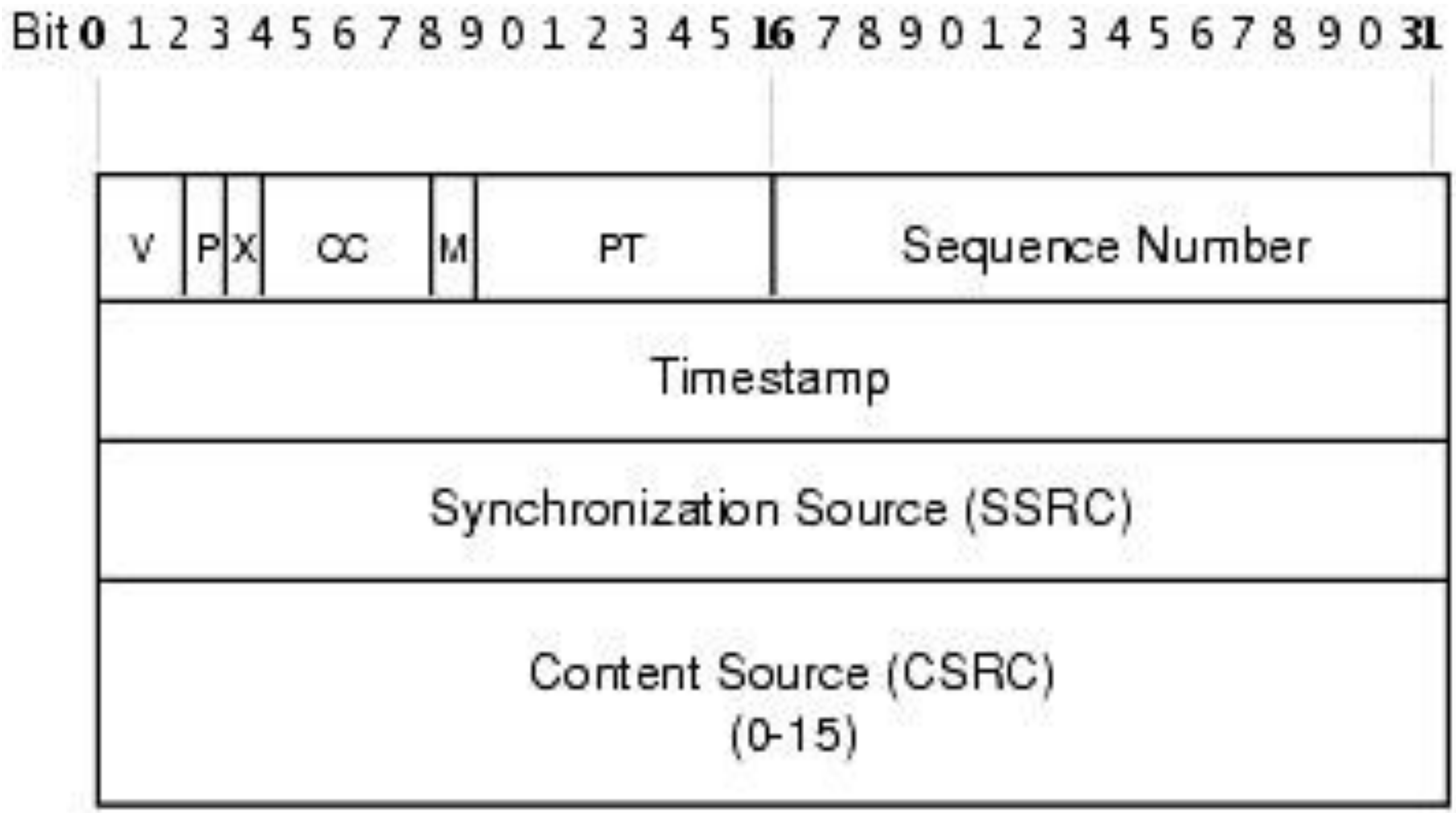
RTP Overhead

Comparing Overhead of UDP/RAW and UDP/RTP

Protocol	*Network Layer*	*Data Link Layer*
UDP/RTP	3.0	5.3
UDP/RAW	2.1	4.4

- The addition of RTP to the protocol stack does not result in an excessive amount of overhead (less than 1 percent)

RTP header


Bit 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 16 7 8 9 0 1 2 3 4 5 6 7 8 9 0 31
V
P
X
CC
M
PT
Sequence Number
Timestamp
Synchronization Source (SSRC)
Content Source (CSRC)
(0-15)

RTP header fields

- Version (Ver)
- Padding (P)
- Extension (X)
- CSRC (CC) Count
- Marker (M)
- Payload Type (PT)
- Sequence Number
- Time-Stamp
- Synchronization Source (SSRC)
- Contributing Source (CSRC)

Time-Stamp Computations

- For video applications, the time stamp depends on the ability of the application to determine the frame number
- When a frame is transmitted as a series of RTP packets, the time-stamp value in each header will be the same

Port Number Utilization

- Under the RTP specification, RTP data is to be transported on an even UDP port number
- RTCP packets are to be transported on the next-higher odd port number

RTP Main Benefits

- It adds a sequence number to the packet to help both the server and the IPTV customer devices (IPTVCD) to detect lost packets. Additionally, this number may also be used by the IPTVCD decoder to reorder packets that arrive from the IP network in the wrong sequence

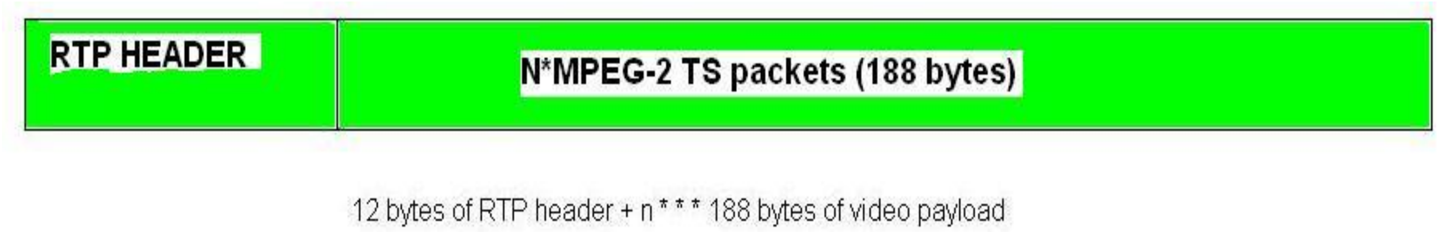
- The timestamp field helps to tackle issues such as jitter and incorrect clock synchronization between source and destination

RTP Drawbacks

- Does not have flow and congestion control to guarantee time of delivery and cannot gurantee other standards of service
- Does not support multiplexing
- Also out of the purview of RTP:
  - Non-delivery of packets
  - Out-of-order delivery
  - Sequence of delivery

RTP with different compression formats (1/6)

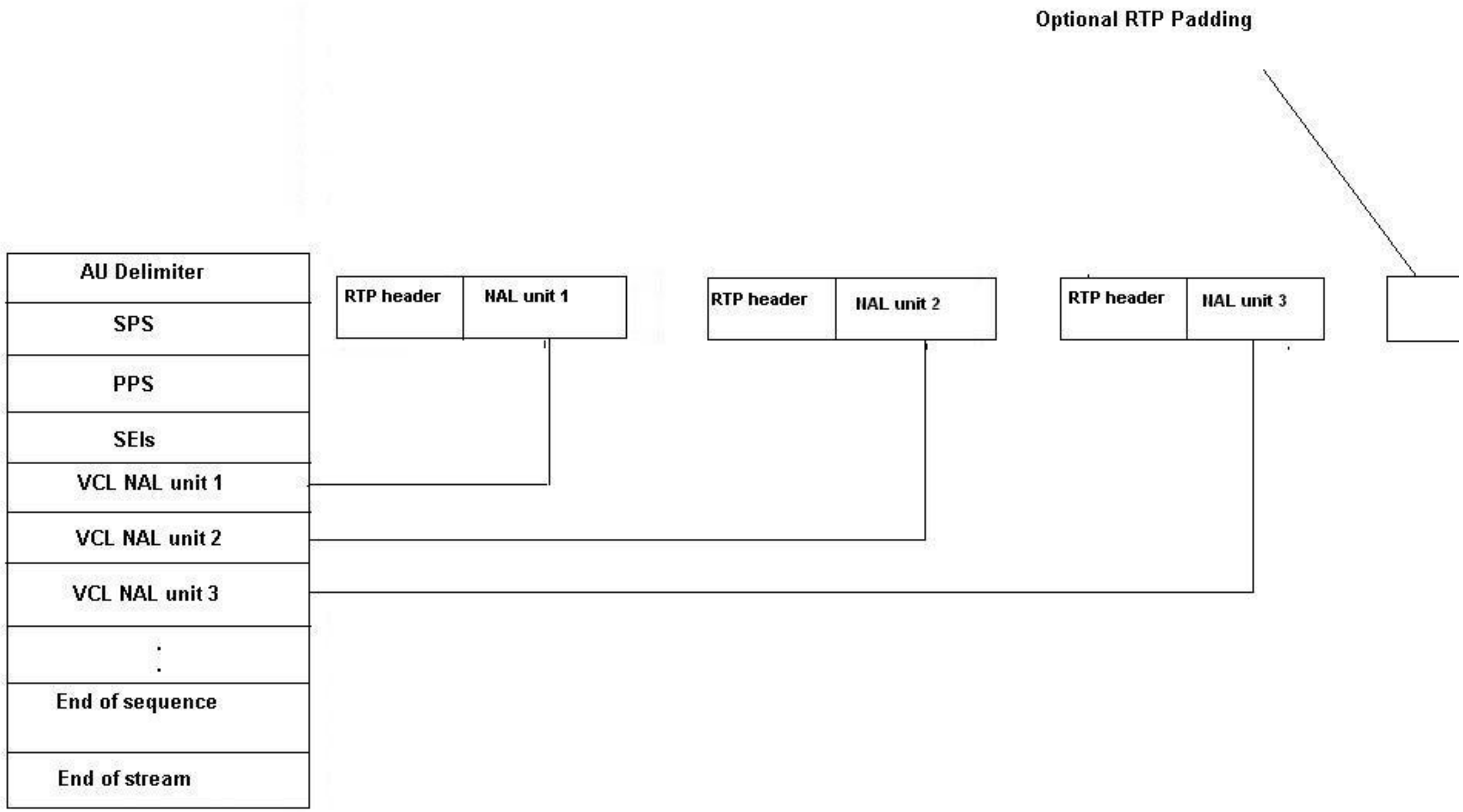
- MPEG-2



RTP with different compression formats (2/6)

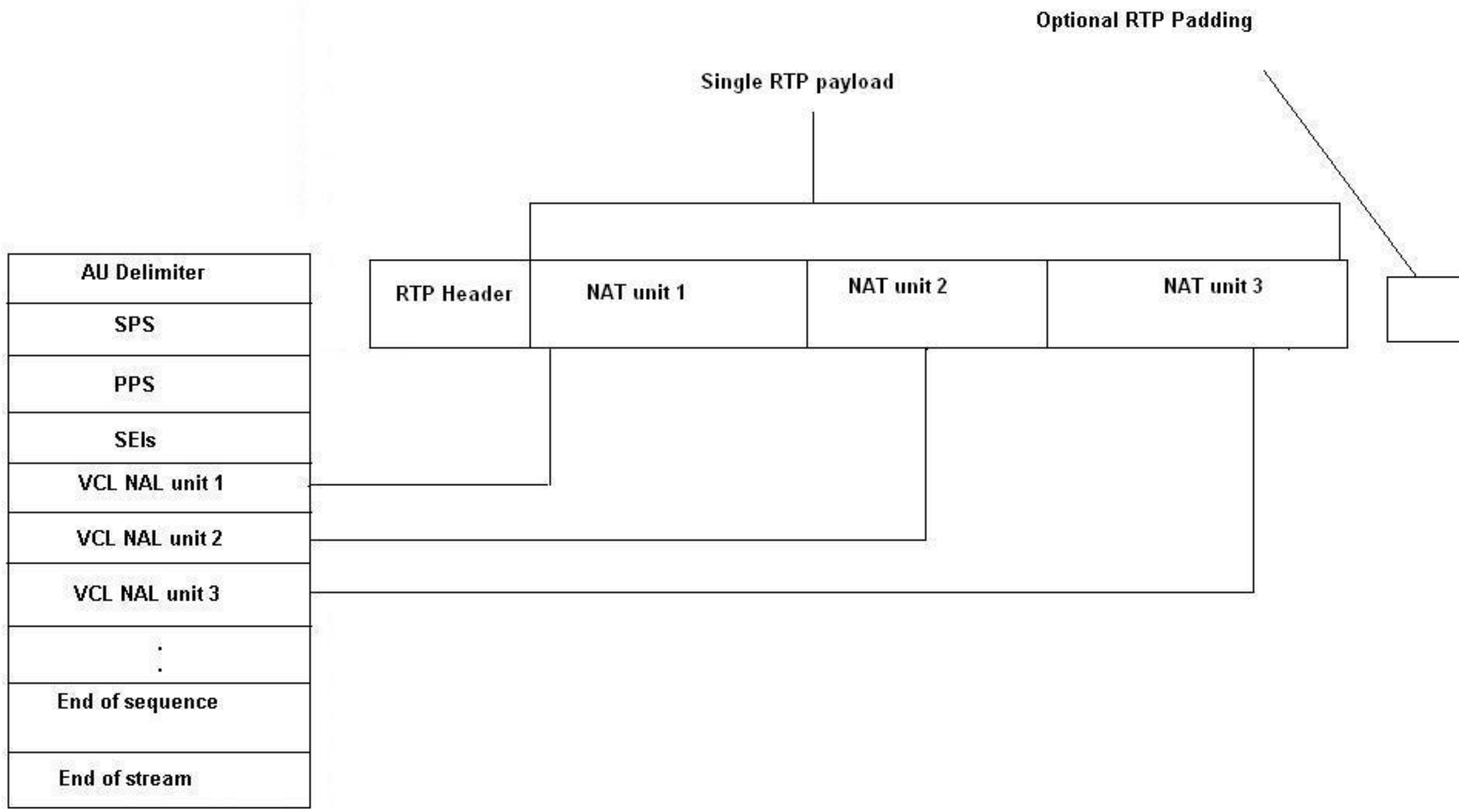
- H.264/AVC
  - three mechanisms for inserting NAL units into the RTP payload:
    - Single NAL unit packet
    - Aggregation NAL unit packet
    - Fragmentation NAL unit packet

RTP with different compression formats (3/6)



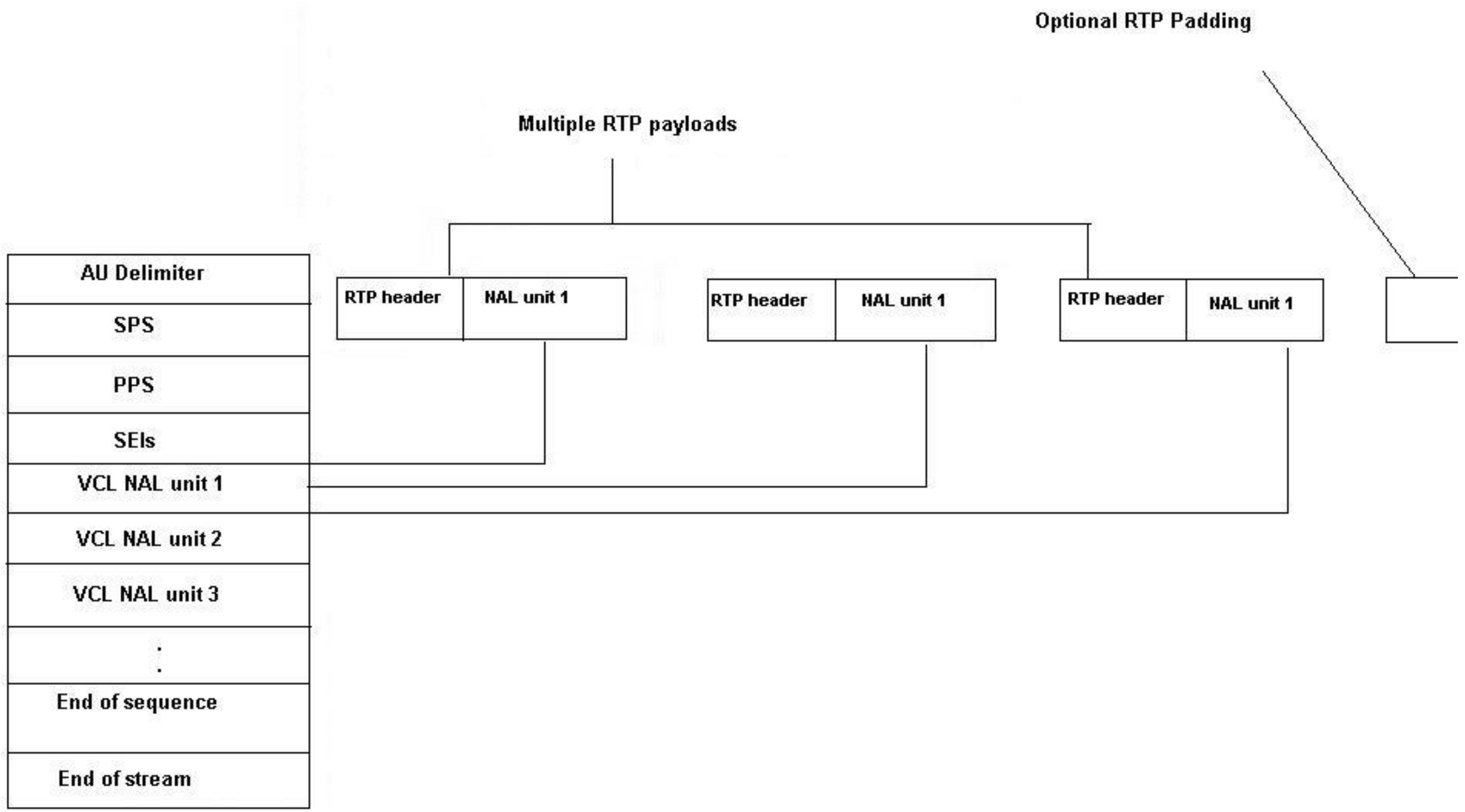
- Mapping H264/AVC content as single NAL units directly into a RTP payload

RTP with different compression formats (4/6)



- Mapping H264/AVC content as multiple NAL units directly into a single RTP payload

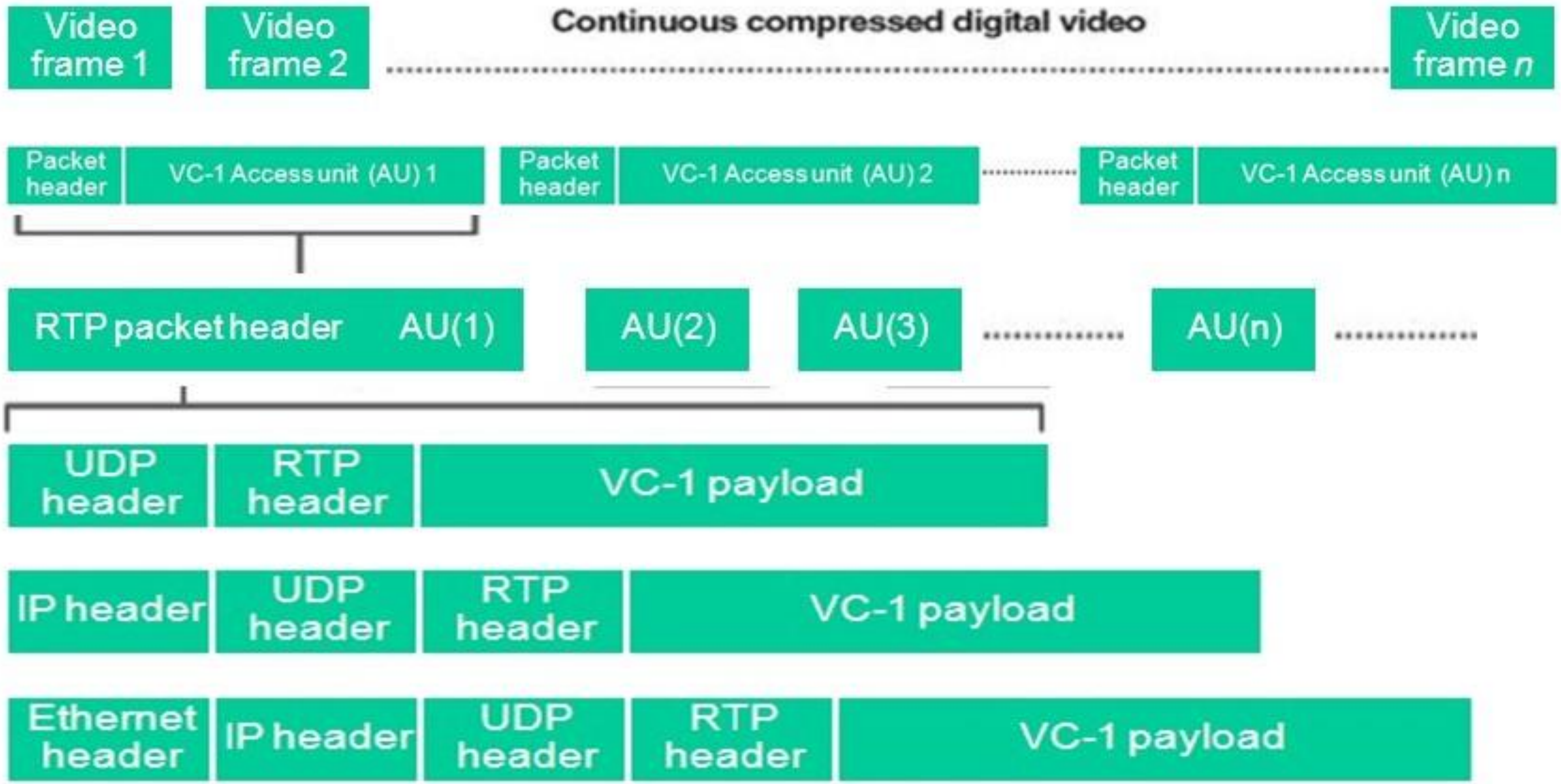
RTP with different compression formats (5/6)



- Mapping a single H264/AVC NAL unit directly into multiple RTP payloads

RTP with different compression formats (6/6)

- VC-1



VC-1 encapsulation layers

RTP QoS

- RTP encapsulation mechanisms are generally deployed across networks that cannot guarantee sufficient levels of QoS for delivering IPTV services

- Although RTP helps to improve the likelihood of streams arriving at their destination points in good order, it has not been designed to guarantee QoS levels.

RTP & RTCP Technical Architecture

- Two linked parts:
    1) Data Part (RTP)
    2) Control Part (RTCP)
- RTP passes issues to the higher layers in the protocol stack and Interactive IPTV application decides which solution to imply:
    1. Slowing down the transmission or
    2. Using a higher compression rate

RTCP

RTCP

- Provides information about the participants in an ongoing session and as well as a mechanism to monitor the quality of service
- Communicates by messages
- Functions as an integral part of RTP's role as a transport protocol

RTCP Protocol Functions

- Gathers statistics on quality aspects of the media distribution and transmit this data
- RTCP provides canonical end-point identifiers (CNAME) to all session participants(unique identification)
- Includes session bandwidth management (dynamic control of the frequency of report transmission)

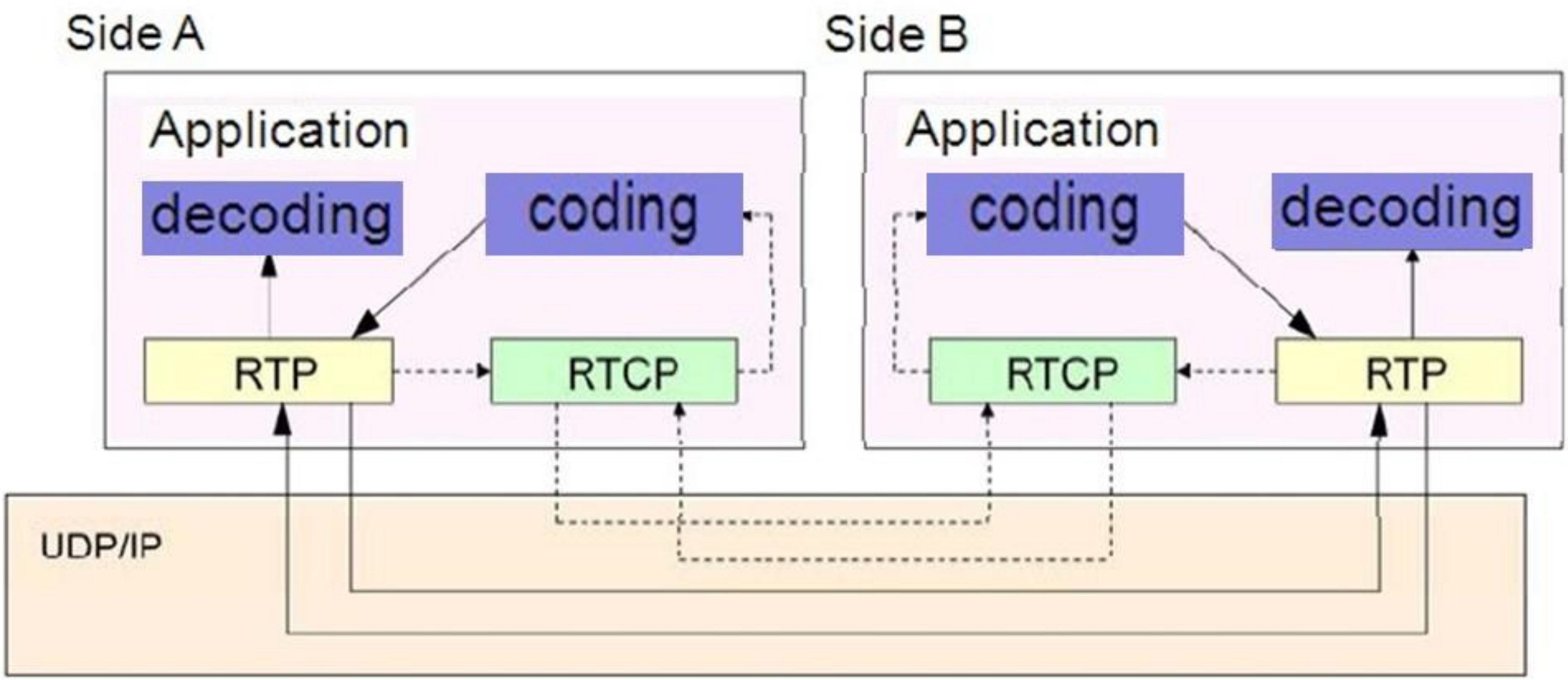
RTCP Message Types

- RTCP distinguishes several types of packets:

1) Sender Report (SR)
2) Receiver Report(RR)
3) Source Description (SDES)
4)Bye (BYE)
5)Application-Specific Message (APP)

- A standards-based extension of RTCP is the Extended Report packet type introduced by RFC 3611

Example: Transmission of Data and Control



Lines:

——— data

- - - - - control

RTP – IN USE

Using RTP

- RTP can be very useful when:

1)Packets Are Arriving Out of Order

2)Packets Are Dropped And Lost

Packets Are Arriving Out of Order

- Packet buffering is one technique that is often used to address the issue of packets arriving out of sequence
- Network protocols such as UDP however are also adding the problem of packets arriving out of order
- Control protocols such as RTP and RTCP can be used to identify issues associated with IP video packets arriving out of order

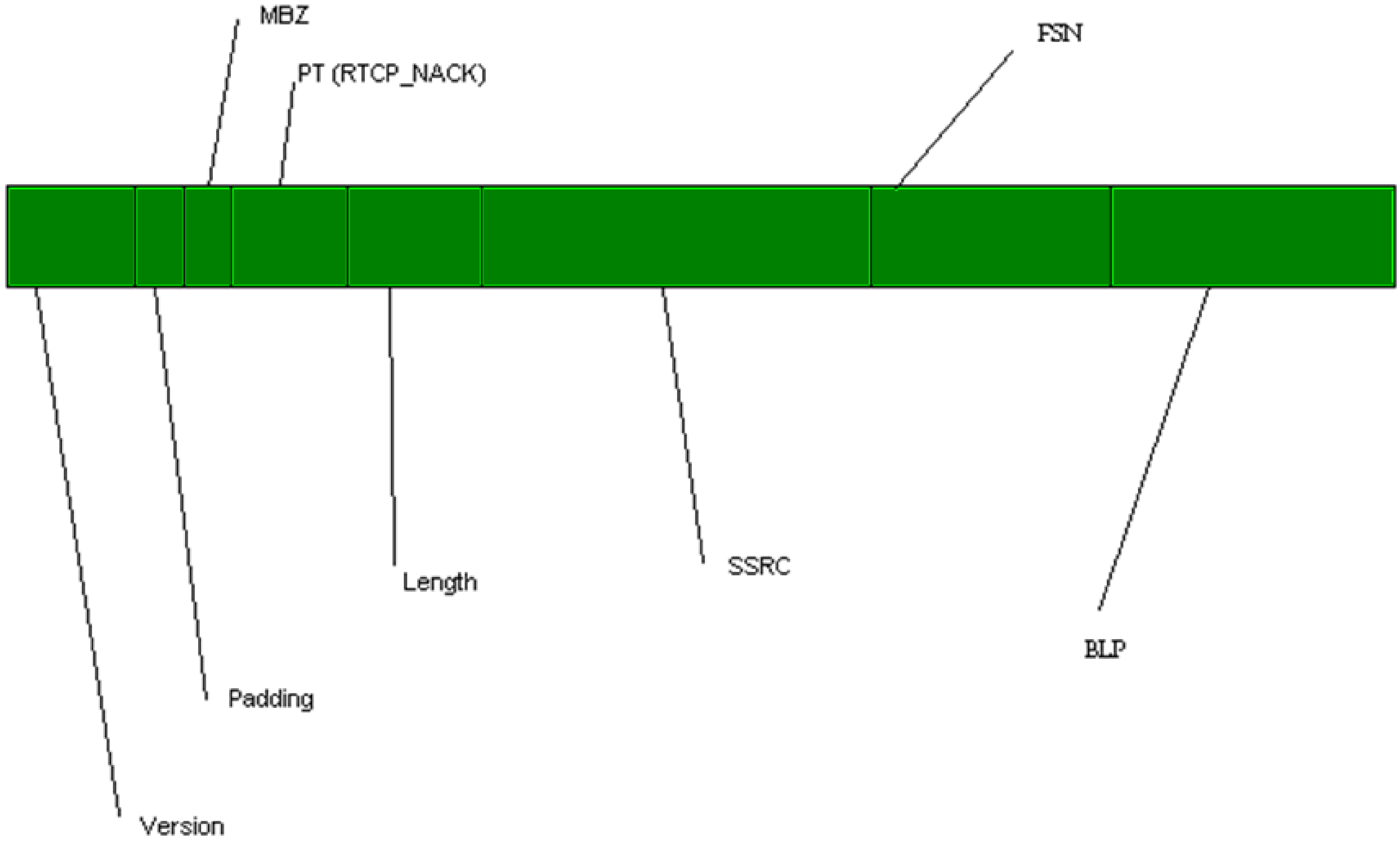
Dropped and Lost IP Packets

- Some techniques to reduce the affects of dropped packets on an IPTV service are:
  1) Using Retransmissions
  2) Using ARQ with TCP
  3) Using RTP to recover from packet loss
  4) Using RUDP and others...

Using RTP to recover from packet loss

- Requires feedback from the receiving IPTVCD to identify lost packets by identifying gaps in the RTP packet numbering sequence
- If a missing packet is identified server receives negative acknowledgement (NACK) message
- Server resends the missing packets

RTCP NACK packet



Secure Real-time Transport Protocol

- Intends to provide encryption, message authentication and integrity, and replay protection to the RTP data in both unicast and multicast applications
- Developed by Cisco and Ericsson and first published by IETF in March 2004 as RFC 3711
- SRTP also has a sister protocol, called Secure RTCP (SRTCP)
- SRTCP to RTCP is as SRTP is to RTP

Jitter & Jitter Buffer

- Jitter is the displacement of packets transporting frames from their original position
- the term packet delay variation is often preferred over jitter.
- Usually jitter buffer at the reciever handles the jitter
- The most popular use of jitter buffers is in VoIP applications

Multiuser Access (1/2)

- RTP sessions
  - Session participants can send, receive, or do both
  - Each media type is transmitted in a separate session, enabling participants to choose which media types they want to receive. For example, a user may just want the audio portion of a streaming music video

- Possible PROBLEMS!

Multiuser Access (2/2)

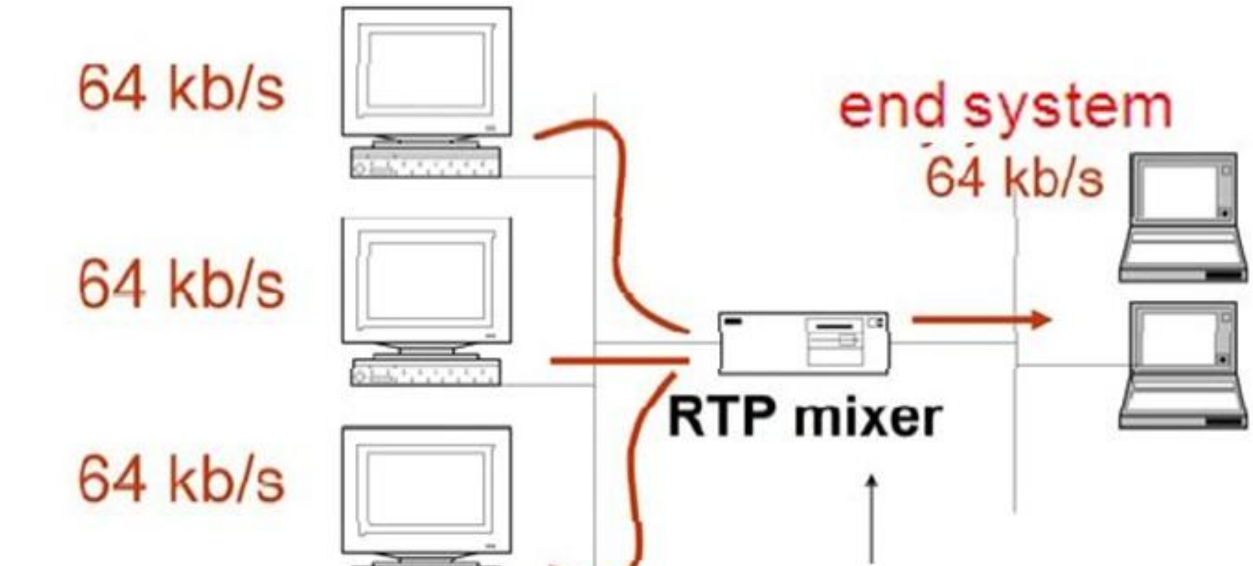
- Possible problems:
  - Not all users want to recieve the same media format
  - There may be differences in terms of access network
  - There may be differences in terms of end-system (terminal)
- For adjustment we use RTP mixer and RTP translator

Example 1: Network Limitations

- Majority of the users is in a high-speed network, and some users are in the network with a slower connection
  - Bad solution: all users use reduced bandwith
  - Better solution: we put a RTP mixer toward the slower part of the network

- RTP mixer is suitable only for audio!

RTP mixer



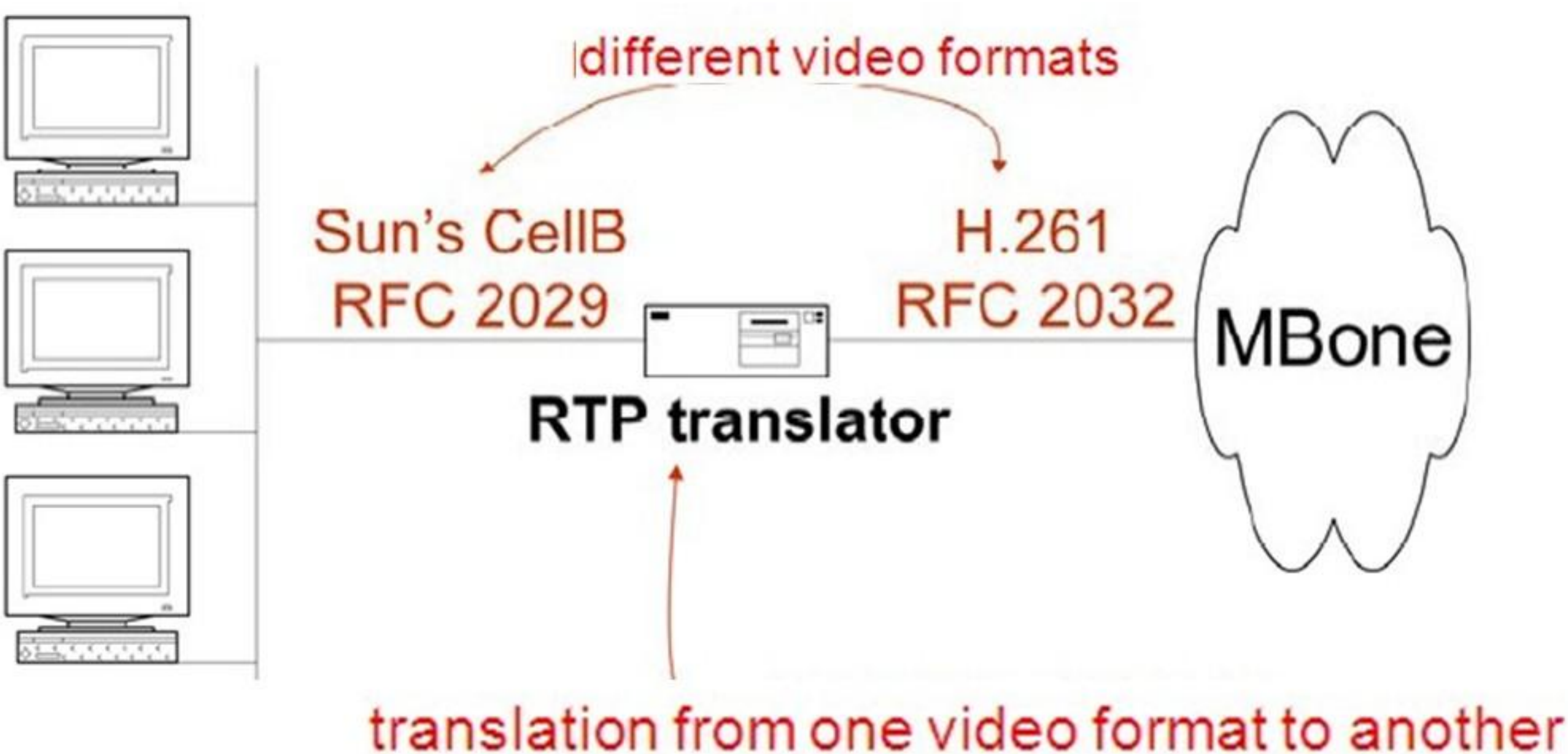
(3 x 64 kb/s = 192 kb/s)

- Collects streams from more sources
- If necessary changes format
- Combines streams in one stream
- Forwards combined stream (in RTP header CSRC list includes some SSRC)

Example 2: Network Limitations

- All users in in a high-speed network, but using different formats
  - No need for combining streams
  - Problem of adaptation of formats is solved with RTP translator

RTP translator



REAL-TIME STREAMING PROTOCOL

Introduction RTSP (1/2)

- Real-Time Streaming Protocol (RTSP) is an application level protocol belonging to the IP communication model that enables IPTVCDs to establish and control the flow of IPTV streams
- The specification for RTSP is RFC 2326
- It can be tranfered with various protocols such as UDP,TCP,RTP and others

Introduction RTSP (2/2)

- Its goal is to offer robust protocol that can stream multimedia over multicast and unicast in "one to many"-applications
- The idea in RTSP is that it acts as a "network remote control" for multimedia servers
- It allows IPTVCDs to issue VCR style commands to an IPTV streaming server

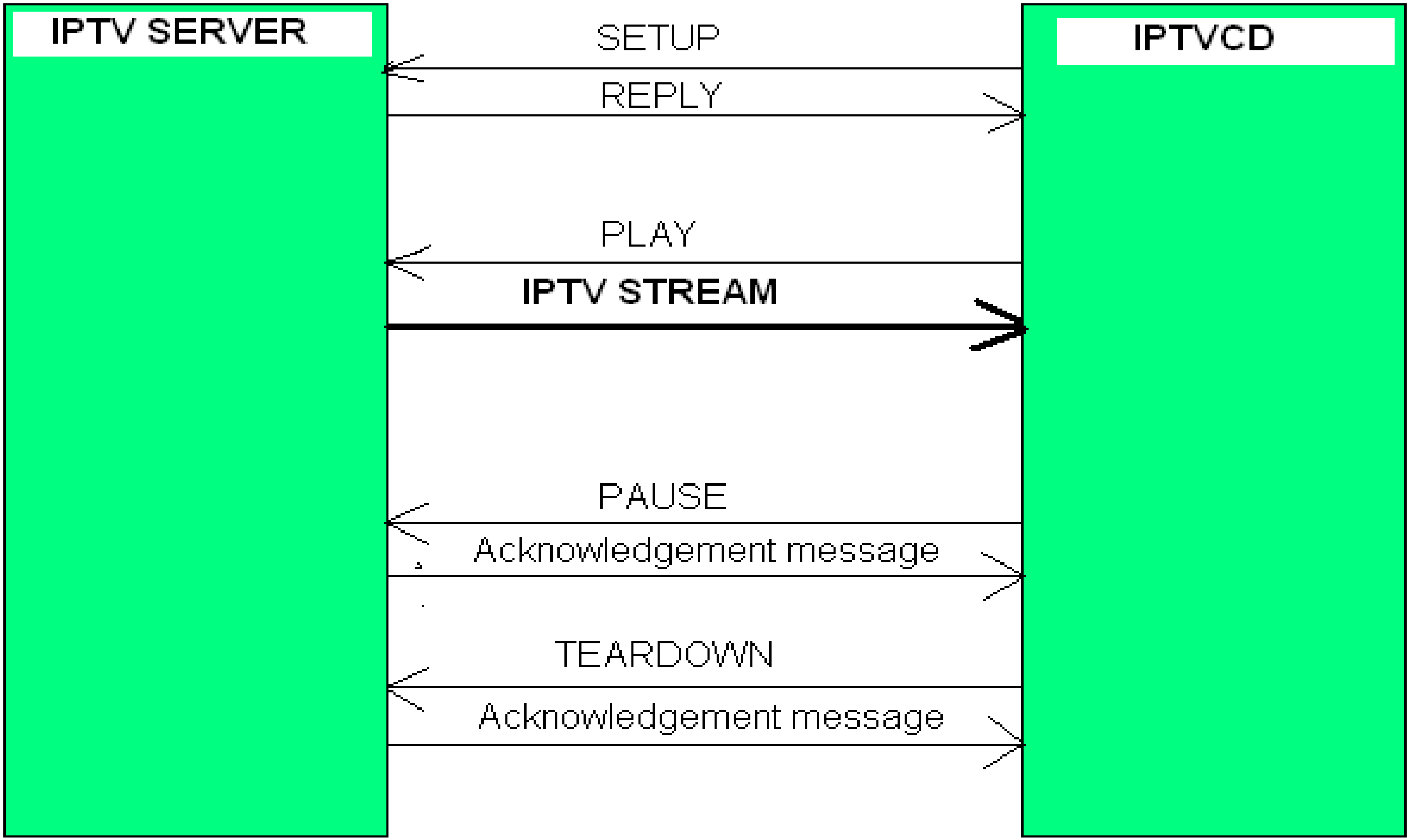
Key Characteristics

- Client-Server Computing Model
- Similar Operational Functionality to HTTP
- Support for Both Unicast and Multicast Traffic
- Independent of Underlying Transport Protocols
- Works in Conjunction with RTP
- RTSP Message Formats

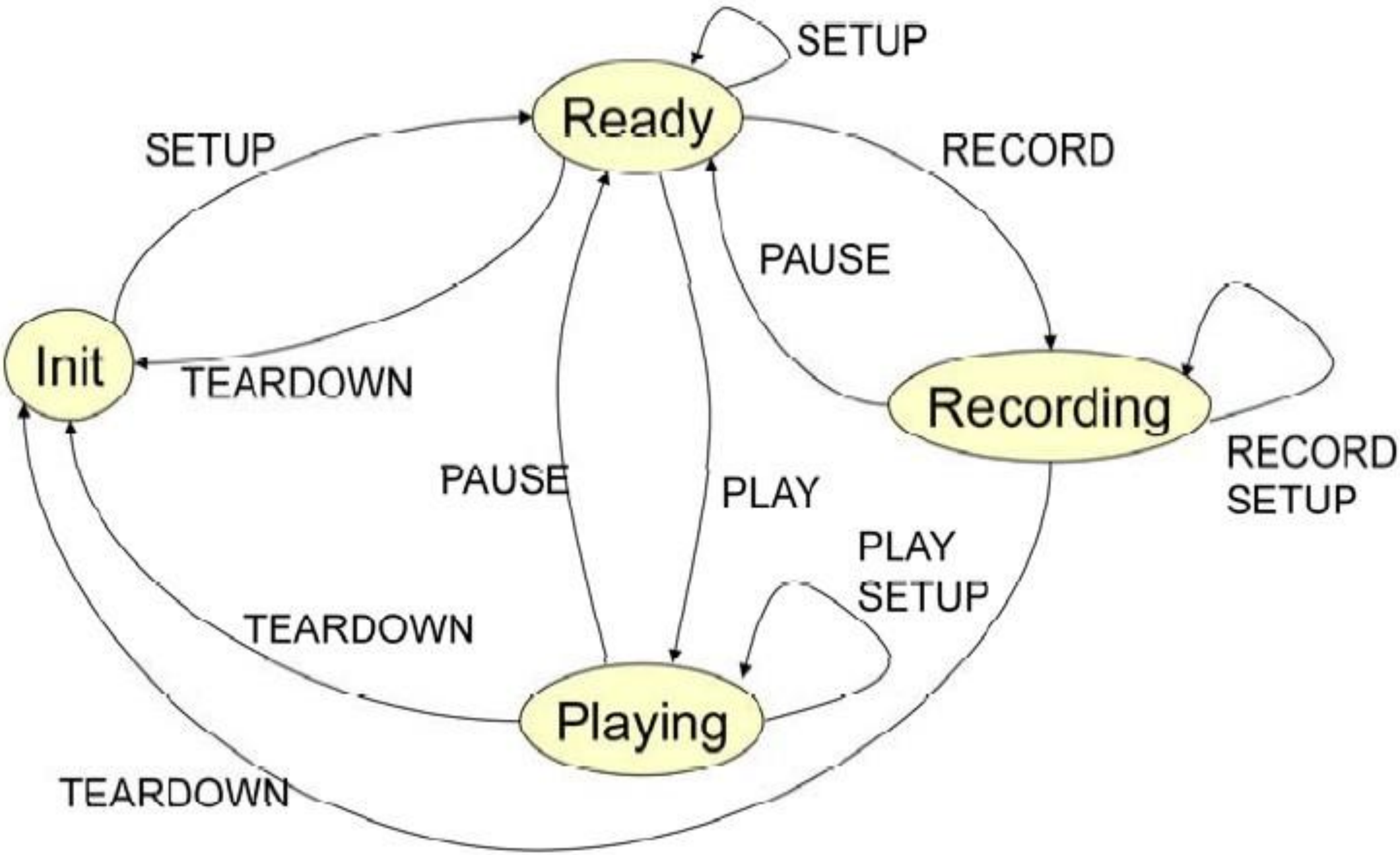
RTSP Commands

- SETUP
- PLAY and RECORD
- PAUSE
- RECORD
- TEARDOWN
- ANNOUNCE
- DESCRIBE

Use of commands in RTSP session



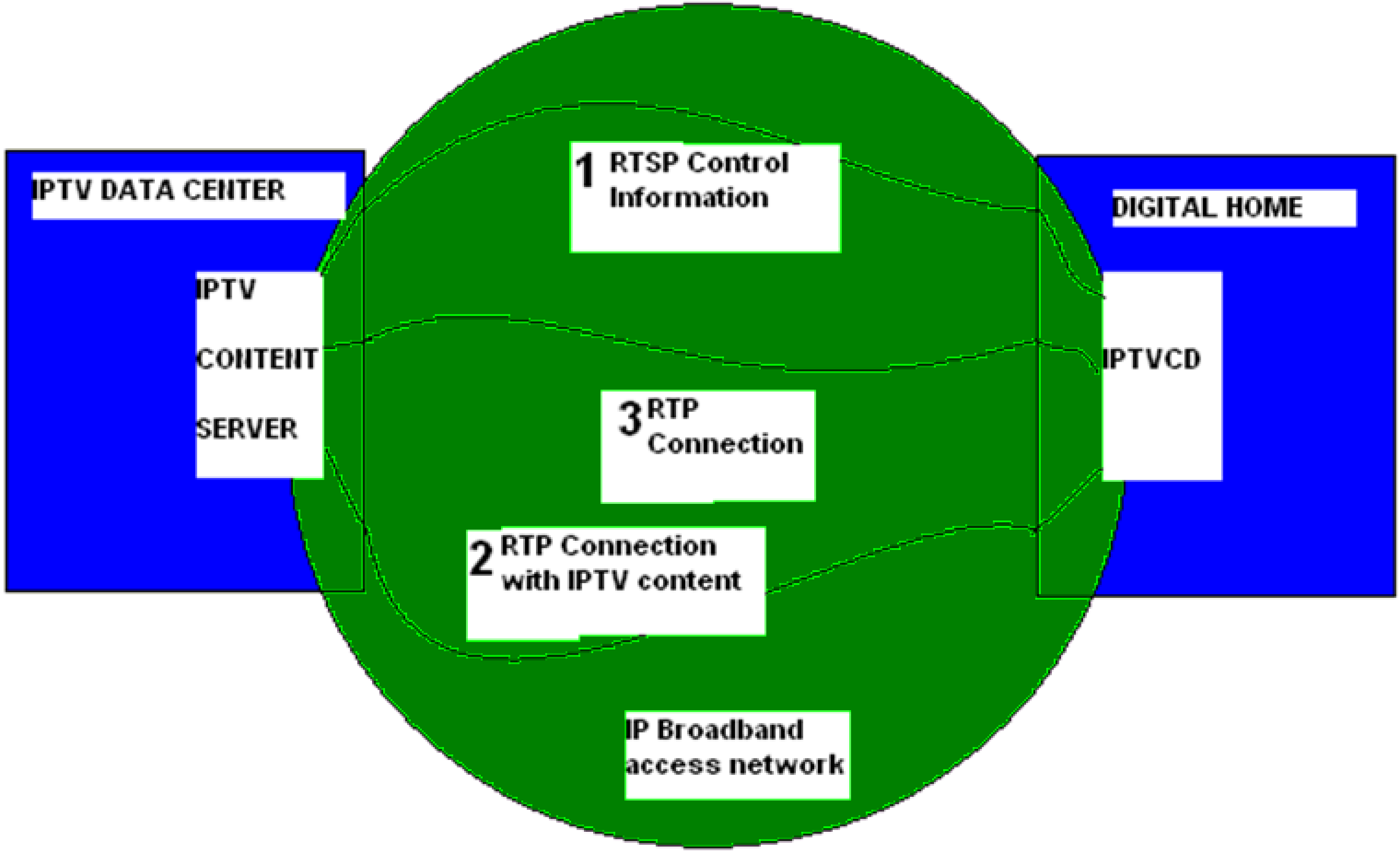
Client and Server Behavior



Client-Server computing model (1/2)

- Three seperate connections are established to provide communications between the RTSP client running on an IPTVCD and the IP-VoD server
  1. An out-of-band connection is established to carry RTSP control information.
  2. A separate RTP over UDP connection is established to carry encoded IPTV content.
  3. The third connection carries RTCP over UDP synchronization information

Client-Server computing model (2/2)



Similar Operational Functionality to HTTP

- The RTSP is intentionally similar in syntax and operation to HTTP/1.1. However, it differs in a number of important aspects from HTTP from which most importantly this three areas:
  1. The protocol identifier is different.
  2. The state of operation
  3. Unlike RTSP HTTP is primarily an asymmetric protocol

RTSP URL

- RTSP URL is in form *rtsp://media.example.com:554/twister/audiotrack*, where *rtsp://* is the identifier for TCP rtsp scheme (*rtspu://* is used for UDP scheme)
- 554 is the assumed port for Real-Time Streaming Protocol
- twister is the name of the presentation
- audiotrack is the name of certain stream in the presentation (this is optional)

RTSP Message Formats

- The messages used by RTSP may be broadly classified into two categories: requests and responses
- The general syntax for an RTSP request message is:

{method name} {URL} {Protocol Version} CRLF {Parameters}

- The general syntax for an RTSP response message is:

{Protocol Version} {status code} {reason phrase} CRLF{Parameters}

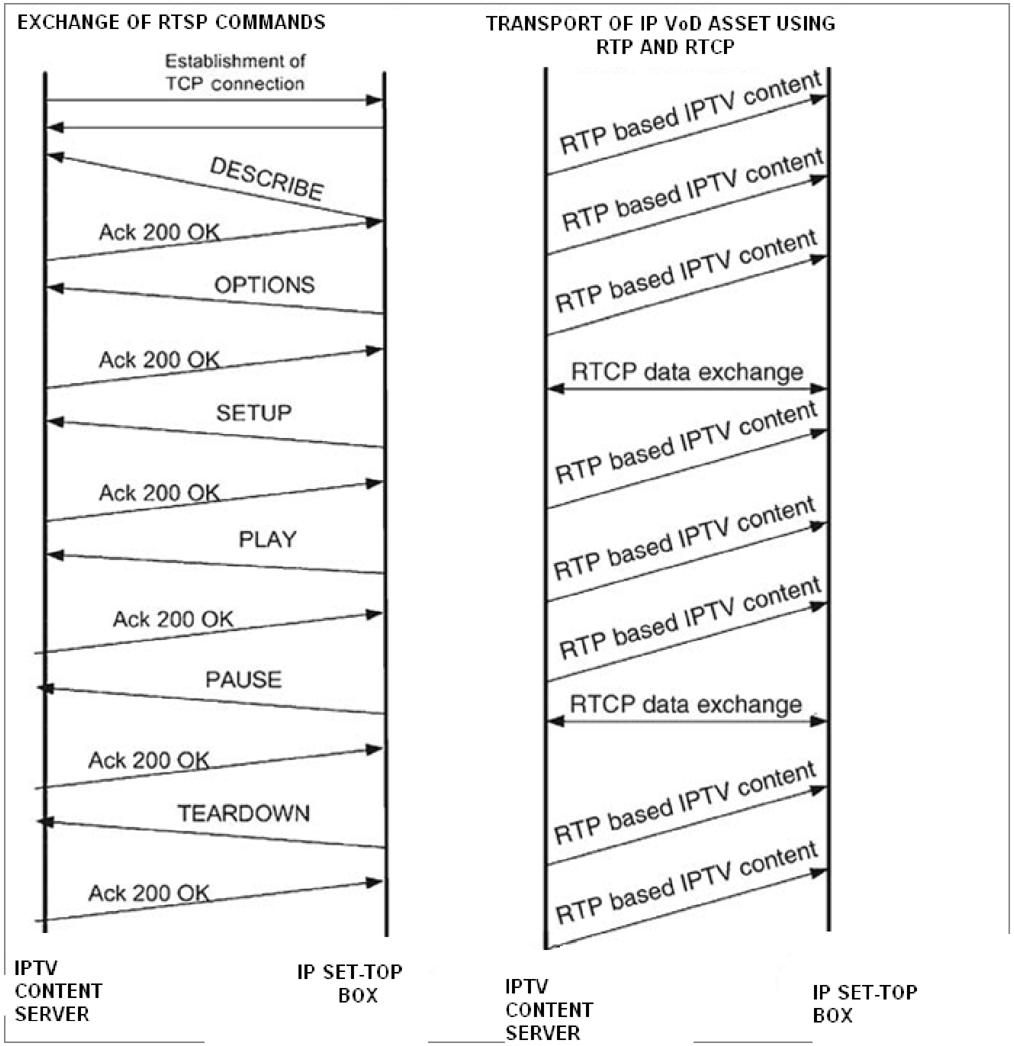
RTSP Methods

- OPTIONS
- SETUP
- ANNOUNCE
- DESCRIBE
- PLAY
- RECORD
- REDIRECT
- PAUSE
- SET PARAMETER , GET PARAMETER
- TEARDOWN

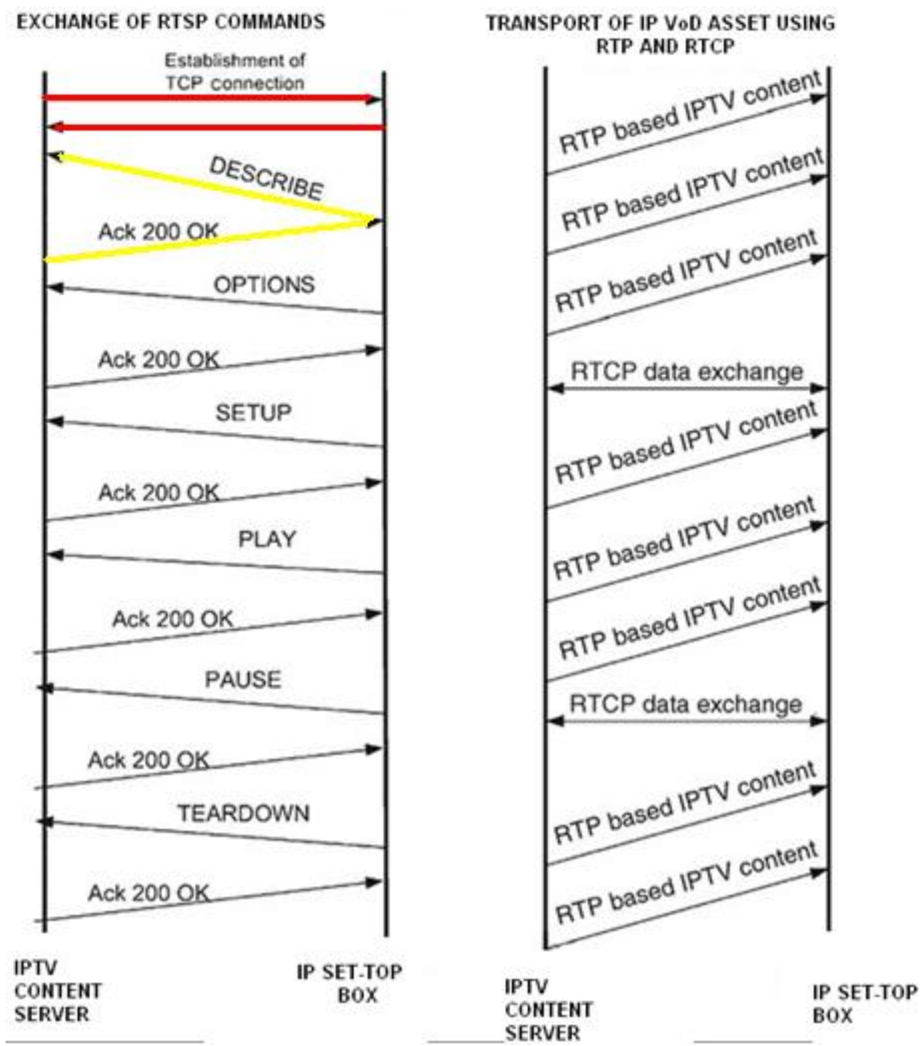
RTSP answers

- Success 2xx
- Redirection 3xx
- Client Error 4xx

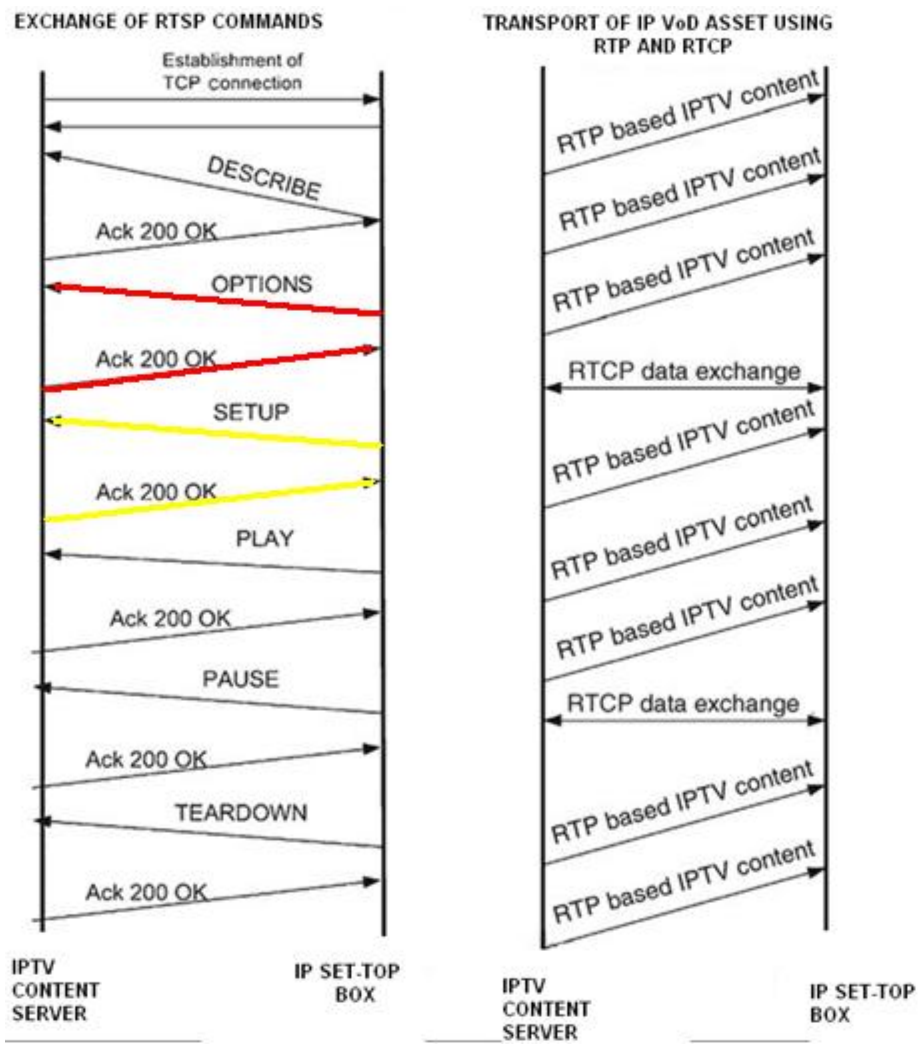
RTSP message format Example (1/5)



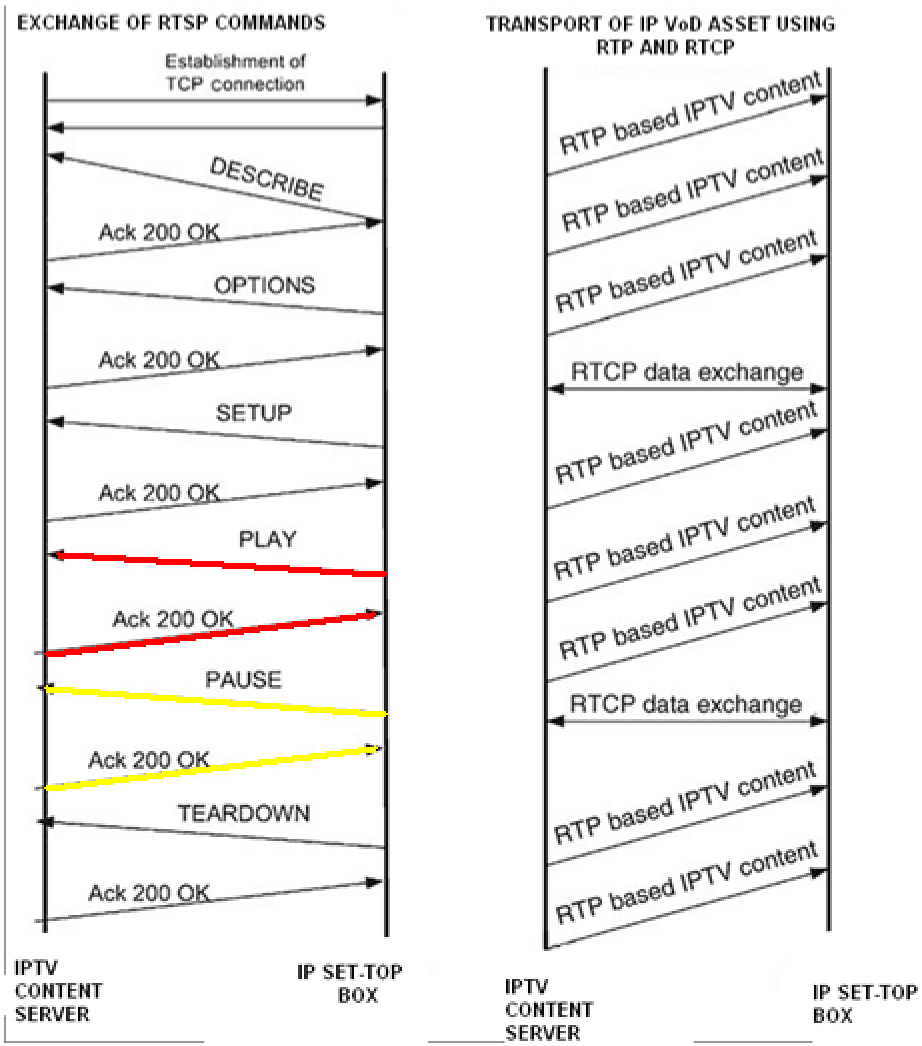
RTSP message format Example (2/5)



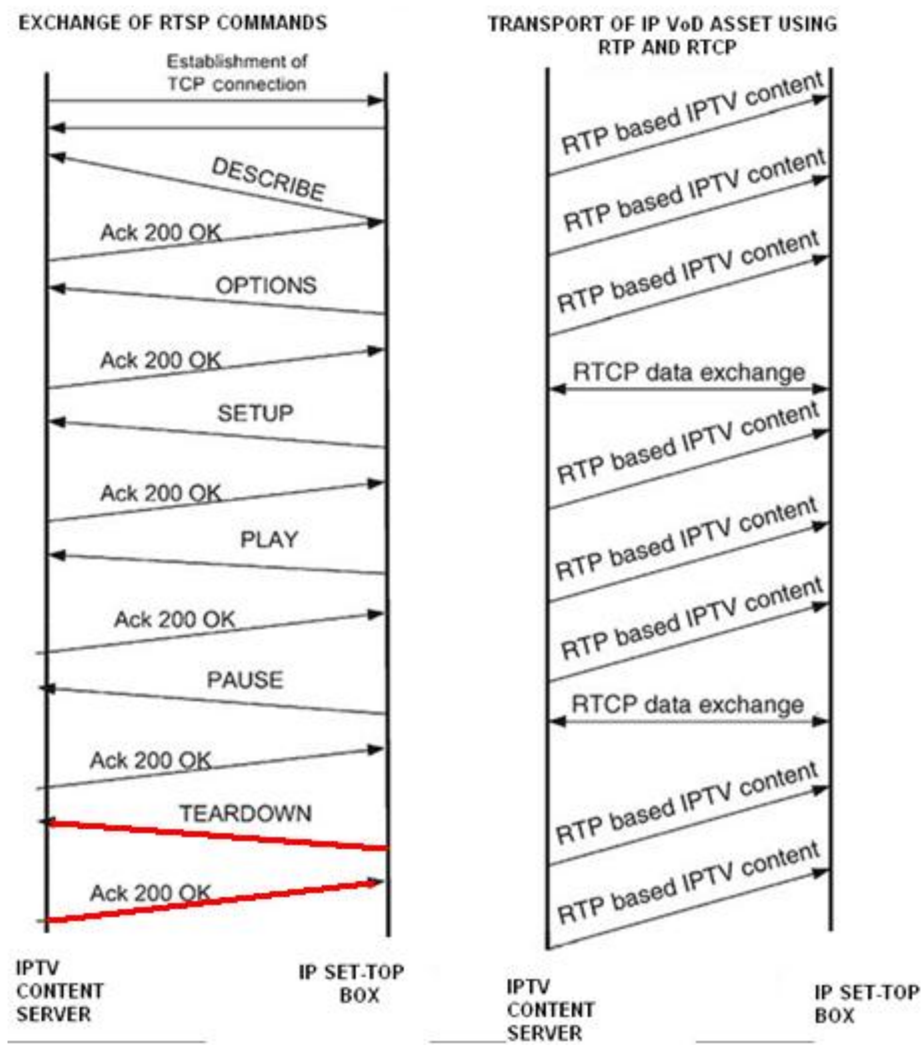
RTSP message format Example (3/5)



RTSP message format Example (4/5)



RTSP message format Example (5/5)



RTSP Properties (1/4)

- **Extendable** (new methods and parametres are easy to add)
- **Easy to parse** (standard HTML or MIME parser can be used)
- **Secure** (HTTP authentication methods, transport and network layer security mechanisms applicable)
- **Transport-independent** (protocols such as UDP, RDP and TCP applicable)

RTSP Properties (2/4)

- **Multi-server capable** (there can be media streams from different servers in one presentation)
- **Control of recording devices** (both playback and recording control possible)
- **Separation of stream control and conference initiation** (The only requirement is that the conference initiation protocol either provides or can be used to create a unique conference identifier)

RTSP Properties (3/4)

- **Suitable for professional applications** (frame-level accuracy through SMPTE time stamps is supported to allow remote digital editing)
- **Presentation description neutral** (no particular format imposed, the presentation description must contain at least one RTSP URI, however)
- **Proxy and firewall friendly** (protocol should be readily handled by both application and transport-layer firewalls)

RTSP properties (4/4)

- **HTTP-friendly** (RTSP reuses HTTP concepts, where sensible)
- **Appropriate server control** (i.e. servers should not start streaming to clients in such a way that clients cannot stop the stream)
- **Transport negotiation** (transport method can be negotiated just before streaming)
- **Capability negotiation** (client must have a way to find out if some of the basic features are disabled in the server)