Lecture 5: Internet Streaming Media

Reji Mathew & Jian Zhang
NICTA & CSE UNSW
COMP9519 Multimedia Systems
S2 2009

IP TV – Video Streaming Example

- **Cisco IP/TV 3400 Series Video Servers**

  Cisco IP/TV Software Version 5.1 Datasheet

  www.cisco.com/en/US/products/hw/contnetw/ps1863/products_data_sheet09186a00801de927.html [a]

Using Industry Standards for Interoperability [a]

An important advantage of the Cisco IP/TV solution is its adherence to industry standards, thus enabling interoperability with other standards-based devices.

Cisco IP/TV technology safely delivers a wide range of video and audio formats, most commonly **MPEG-2** for the highest-quality broadcasts, **MPEG-1** for TV quality, and **MPEG-4**, the new format for high-quality video and audio at low bit rates.

Cisco IP/TV technology is also based on standard **IETF streaming protocols** such as Real-Time Transport Protocol (**RTP**), Real-Time Streaming Protocol (**RTSP**), and IP Multicast. By adopting such standards for media streaming, the Cisco IP/TV solution benefits by interoperating with other devices based on the same standards. ……….

Background

- So now you can code video (and audio)
- But how to transfer this video to your mates ?

- Possibilities
  - File Transfer – TCP
  - Streaming for real-time delivery – TCP, UDP

- Protocols for Enabling Streaming
  - RTP – Transport
  - RTCP – Control
  - SDP - Description
  - RTSP - Signaling

Lecture Outline

- Introduction to Streaming
- Streaming Protocols
  - RTP
  - RTCP
- System Overview
  - System for streaming video and audio on the Internet
  - Overview of important components
  - Sender side issues
  - Receiver side issues

Introduction

- Facts on TCP (Transmission Control Protocol)
  - Data sent on a TCP connection is delivered
    - reliably and
    - in order at the destination.
  - Transmission made reliable via
    - use of sequence numbers and
    - acknowledgments.
  - Little control over the timely arrival of data
    - Arbitrary number of retransmissions
  - Not suitable for real-time streaming
    - Real-time applications have stringent limits on acceptable delay
    - eg video conferencing

Introduction

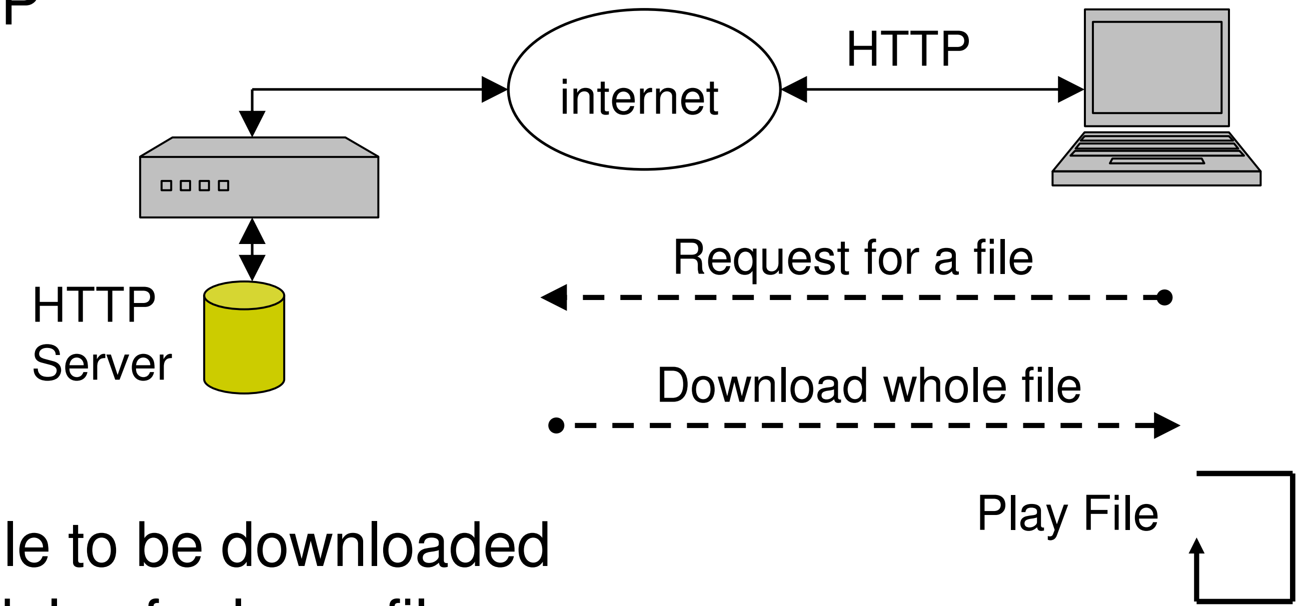
- Facts on UDP (User Datagram Protocol)
  - Datagram mode of communication
  - Send messages with a minimum of protocol mechanism
  - Unordered datagram service
  - delivery and duplicate protection are not guaranteed

  - UDP is unreliable
  - But used when timeliness is more important
    - you have control over when data is sent
    - Preferred for streaming

  - TCP is more reliable
  - Used when reliability is more important (than timeliness)
    - Eg, file download

Introduction

- File Transfer
  - Examples : HTTP, FTP
  - Using TCP

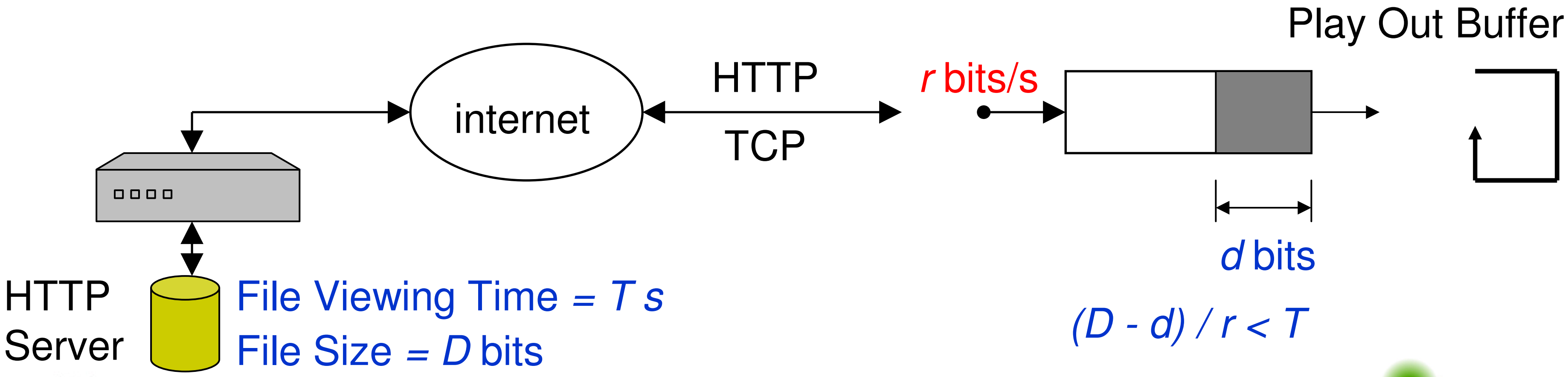


  - Wait for file to be downloaded
  - Lengthy delay for large files
  - Not suitable for real-time applications

Introduction

- Streaming
  - Using TCP (not preferred)

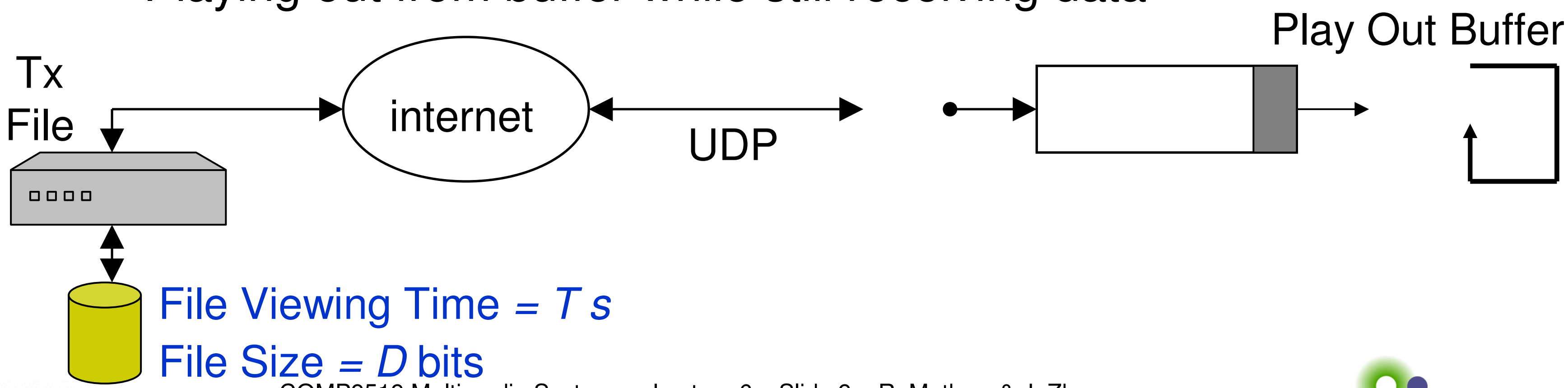
  - Wait until some content is downloaded to a buffer
  - Start playing out from buffer while still receiving data
  - If data rate drops then will need to stop play out
    - Reduction in data rate due to congestion
    - Frequent start and stop of play out



Introduction

- Streaming
  - Using UDP (preferred)

  - Transmit data rate matches coded rate of media
    - Eg Transmit rate = $R = D/T$
  - Start playing out after small amount data is downloaded
    - Small delay (smaller than when using TCP)
    - Play out rate = R
    - Playing out from buffer while still receiving data



Introduction

- So we choose UDP/IP (over TCP/IP) for streaming

- But multimedia data have other requirements
  - Sequencing:
    - to reorder packets at the receiver that are out of order
    - to detect lost packets
  - Intra-media synchronization:
    - timing information for play out of successive packets
    - Eg, period of time to show/hold a single video frame
  - Inter-media synchronization:
    - timing information to synchronize multiple media streams
    - Eg, synchronize audio and video streams (lip sync)

Introduction

- So we choose UDP/IP (over TCP/IP) for streaming
- But multimedia data have other requirements [1] *(Continued)*
  - Sequencing:
  - Intra-media synchronization:
  - Inter-media synchronization:
  - Payload identification:
    - Communicate the codec being currently used
    - Allow codec switching to adjust to bandwidth or user capabilities
  - Frame indication:
    - To indicate start and end of a frame or access unit
    - Helps in delivery of data to higher layers

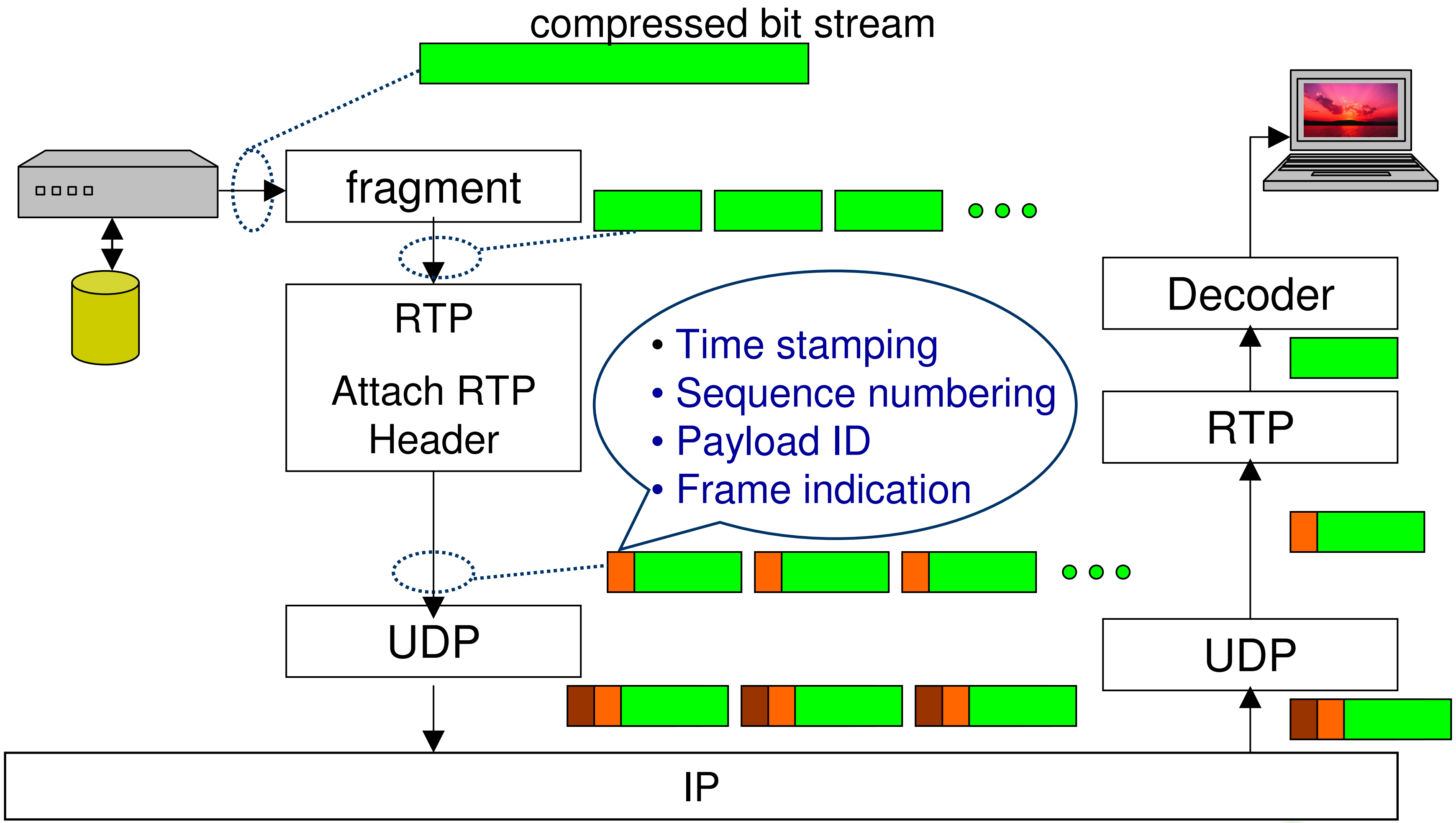
Lecture Outline

- Introduction to Streaming
- Streaming Protocols
  - RTP
  - RTCP
- System Overview
  - System for streaming video and audio on the internet
  - Overview of important components
  - Sender side issues
  - Receiver side issues

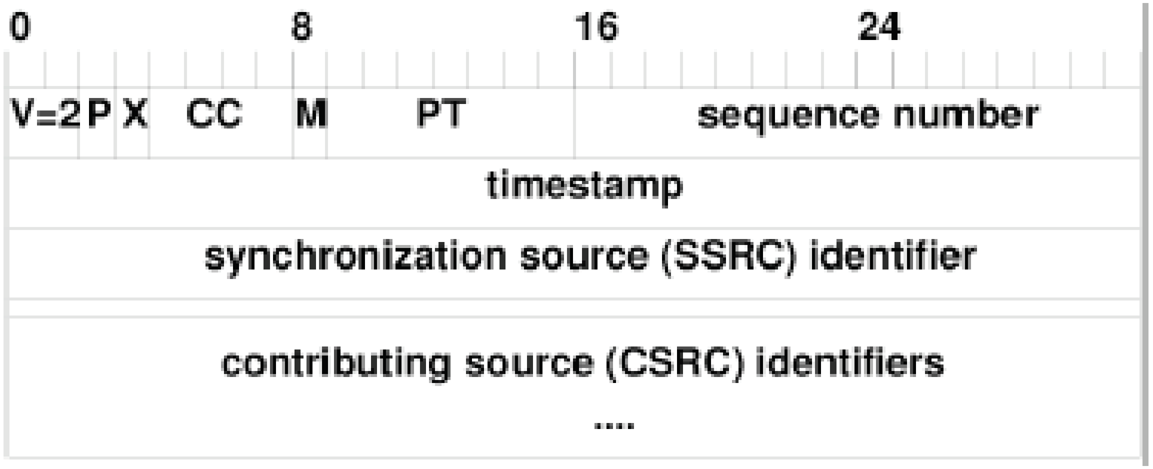
Internet Streaming Protocols

- How to meet requirements for multimedia streaming
  - Solution - Real Time Transport Protocol (RTP)
    - Real Time Control Protocol (RTCP)
- RTP/RTCP
  - Defined in a generic way
    - Developed by IETF
    - RFC 3350 RTP: A Transport Protocol for Real-Time Applications
    - More info ftp://ftp.rfc-editor.org/in-notes/rfc3550.pdf
  - Designed to be multi-cast friendly
  - Generally used with UDP
    - Although designed to be independent of underlying transport
    - And network layers

RTP Overview



RTP Header Fields

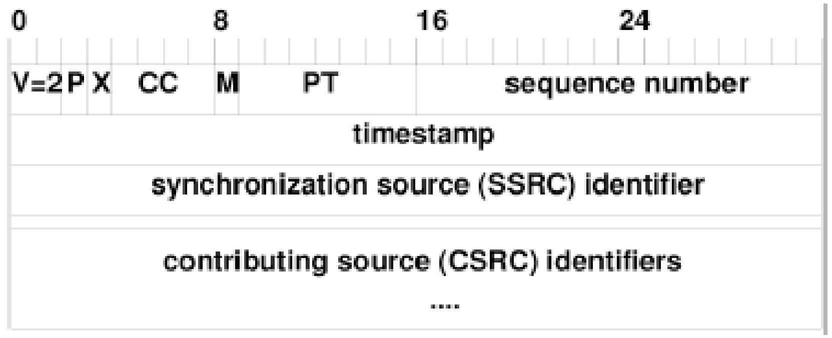


- The first twelve bytes (octets) are present in every RTP packet.

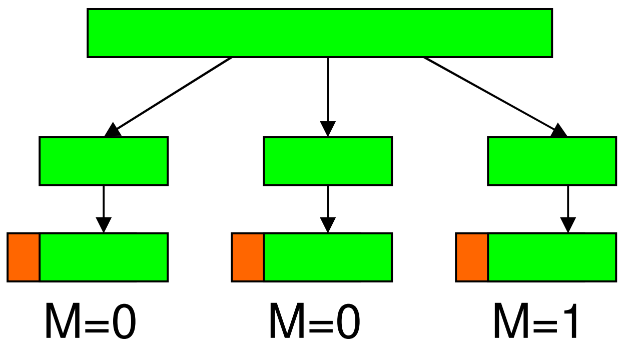
RTP Header Fields

- Marker (M) : 1 bit

  Intended to allow significant events such as frame boundaries to be marked in the packet stream



MPEG-4 Example : set to 1 to indicate last RTP packet of a VOP.

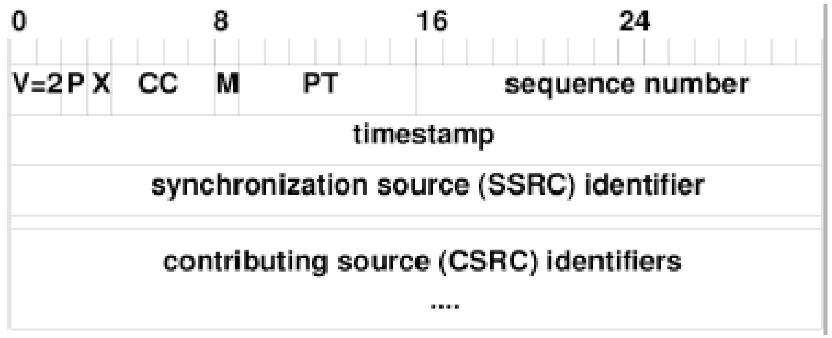


One compressed frame or VOP

Broken up into 3 packets

RTP Header Fields

- Payload Type : 7 bits
- PT



This field identifies the format of the RTP payload (eg payload is MPEG-4 video or u-law audio ….)

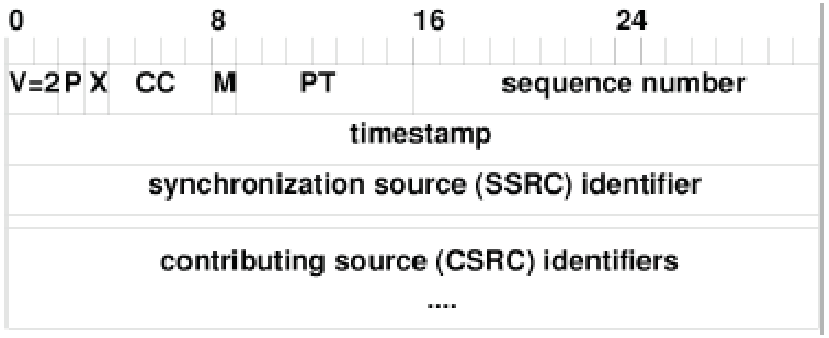
Static mapping

Payload Type value ⟷ codec

Now dynamic Payload types are used, where an out-of-band mechanism (such as SDP) is used to map the Payload field values to codecs.

RTP Header Fields

- Sequence Number : 16 bits

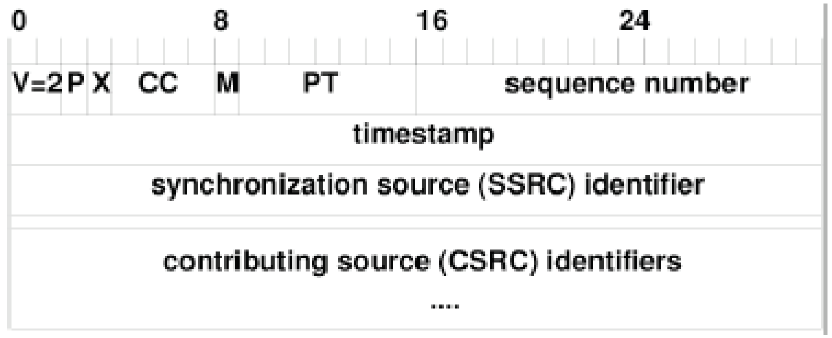


The sequence number is incremented by one for each RTP packet sent. Usually made to start at a random number.

Sequence number can be used by the receiver to reorder out of order packets and detect packet loss.

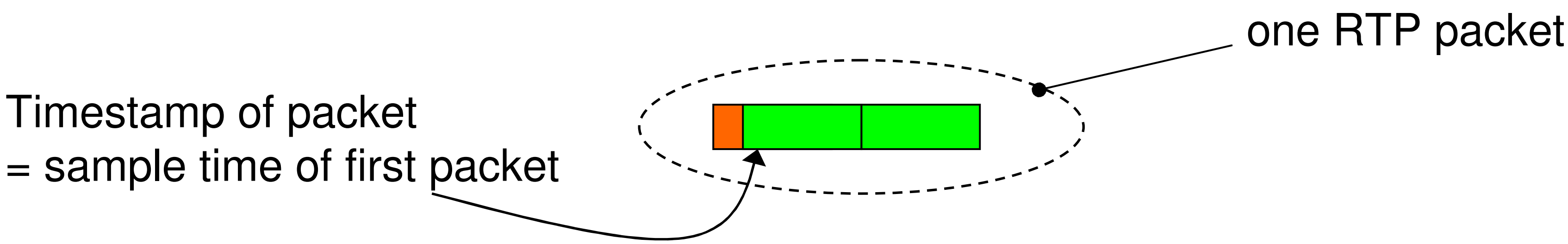
RTP Header Fields

- Time Stamp : 32 bits



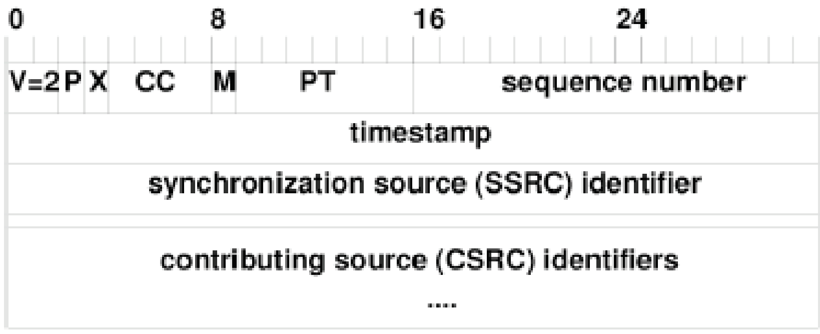
The timestamp specifies the sampling instant of the first octet in the RTP data packet.

Example : two audio samples in one packet, timestamp refers to the sampling instant of the first sample.



RTP Header Fields

- Time Stamp : 32 bits



The timestamp is derived from a clock that increments monotonically and linearly in time.

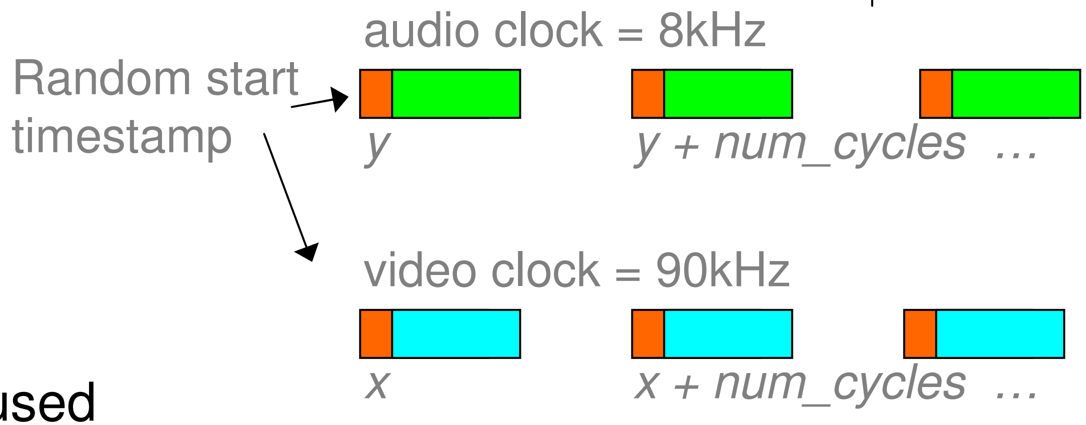
The clock frequency is dependent on the payload format. This means RTP timestamps are in units which are meaningful for the format of media being carried.

Examples :

- 8KHz sampled u-law audio would use an 8KHz sample driven clock for timestamps.
- MPEG video would use the MPEG 90KHz clock.

RTP Header Fields

- Time Stamp : 32 bits



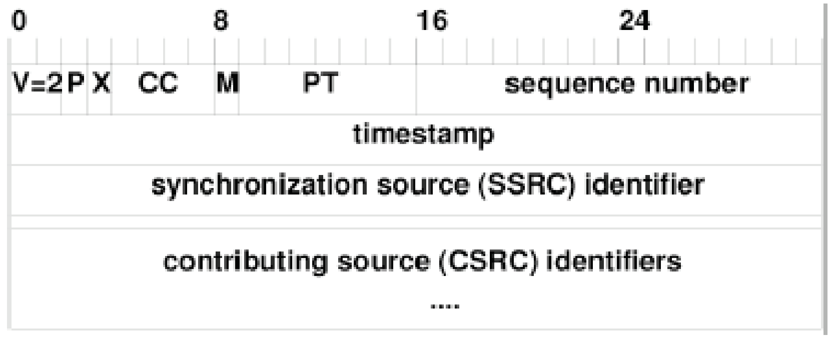
RTP Timestamps can be used
at the receiver for intra media synchronization.

At the receiver inter media synchronization uses RTP Timestamps with the help of RTCP.

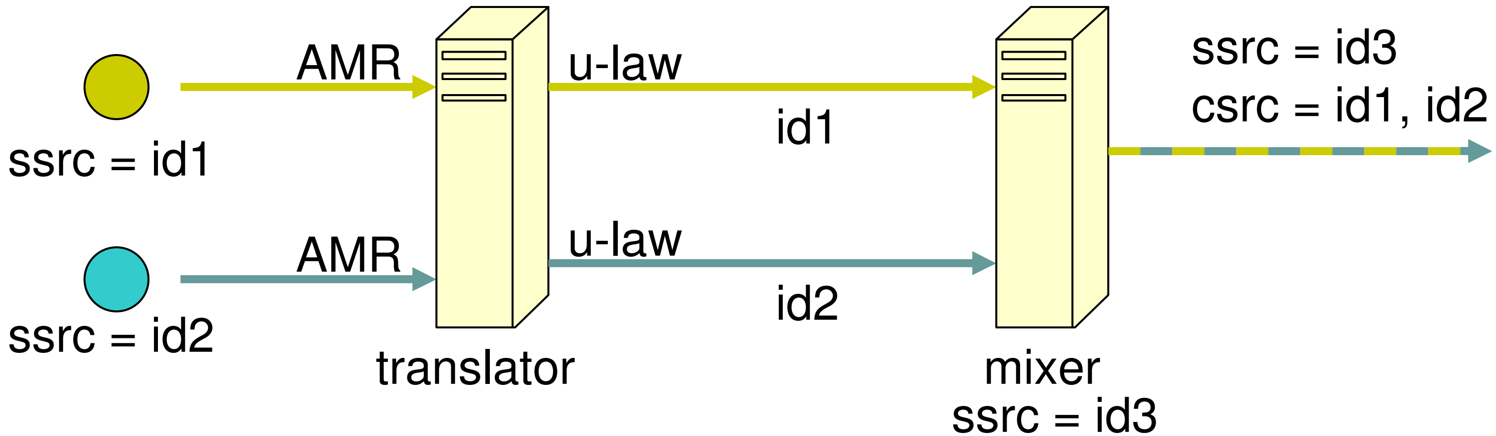
- clocks for audio and video streams from the same source use different timebases.
- RTCP relates these different clocks to a reference clock (wall clock), so audio-video synchronization can be performed.

RTP Header Fields

- SSRC : 32 bits

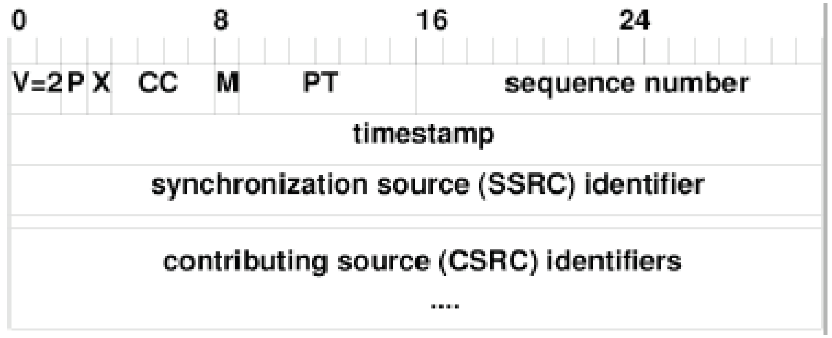


  - Synchronization source
    - identifies an RTP sender
    - value chosen randomly
    - identify sources in multi-sender sessions

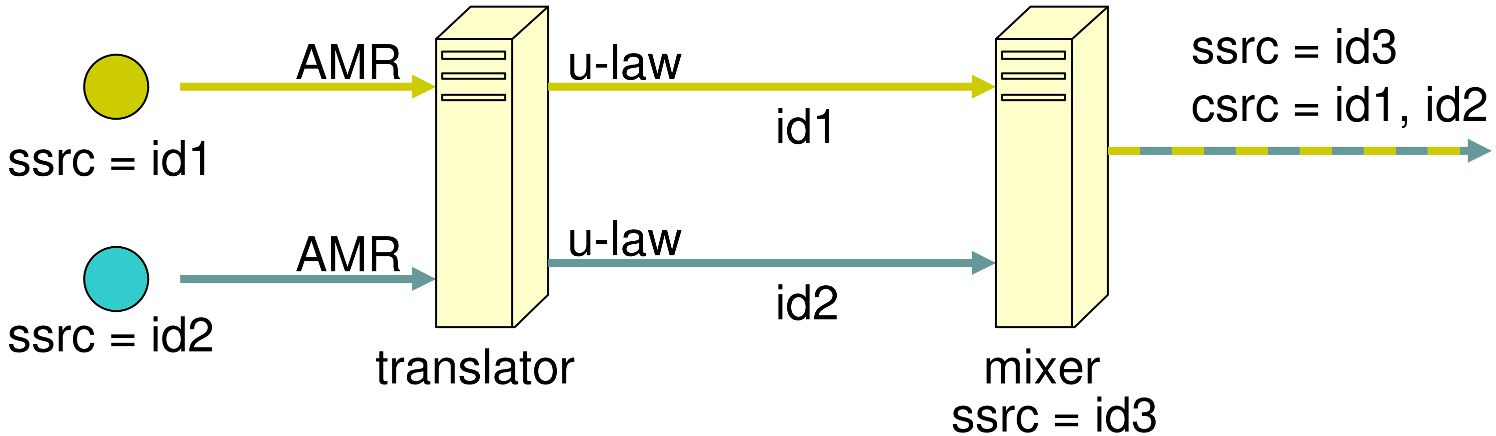


RTP Header Fields

- CSRC : 32 bits each
  - Contributing Sources
  - Identifies contributing sources



  - number of contributing sources specified by CC field
  - CSRC identifiers are inserted by mixers
    - using SSRC of contributing sources



RTP

- Mixers and Translators

- Mixers
  - several media stream to one new stream
    - Possible new encoding
  - reduces bandwidth required
    - Example : mix many audio stream into one, audio conference

- Translator
  - single media stream (input and output)
  - may change encoding
    - Example : transcode to suit low bit rate conditions
  - may perform protocol translation

RTP

- RTP Header Fields

  - CSRC : 32 bits each
  - Contributing Sources

  - Example use of CSRC

    Audio conference where multiple audio streams are mixed together to form an output packet stream.

    The CSRC list identifies all the sources that have been mixed together thereby allowing identification of current participants or talkers in the session.

    Note that mixers will generate its own timing for the combined stream, therefore a mixer is identified by its own SSRC

RTCP

- RTP – allows for transport of media data
- But….
  - How do we know the packets are being received ?
    - packet loss rate ?
    - available bandwidth ?
  - What are the bounds on delay and jitter ?
- Need some feedback on network performance
- Real-Time Control Protocol (RTCP)
  - Used in conjunction with RTP
  - Primary function – feedback on quality of data distribution

RTCP

- Control information exchanged between session members by RTCP packets
- Five RTCP packet types defined
- RTCP packets begin with a fixed part
  - Similar to RTP data packets
- RTCP packets are sent periodically
  - from all participants of a session
  - multicast example
- RTCP packets are stackable – compound packets
  - concatenate multiple packets
- Usually sent over UDP (same as RTP)
  - Using a port number = RTP_port + 1

RTCP

- RTCP packet types to carry control information
  - RTCP SDES : Source description items
    - sent by all session members
    - includes a persistent identifier of sources in the session
  - RTCP SR : Sender Report
    - sent by active senders
    - report of transmission and reception statistics
  - RTCP RR : Receiver Report
    - Sent by receivers (not active senders)
    - report of reception statistics
  - RTCP BYE : Packet to indicate end of participation
  - RTCP APP : Application specific functions

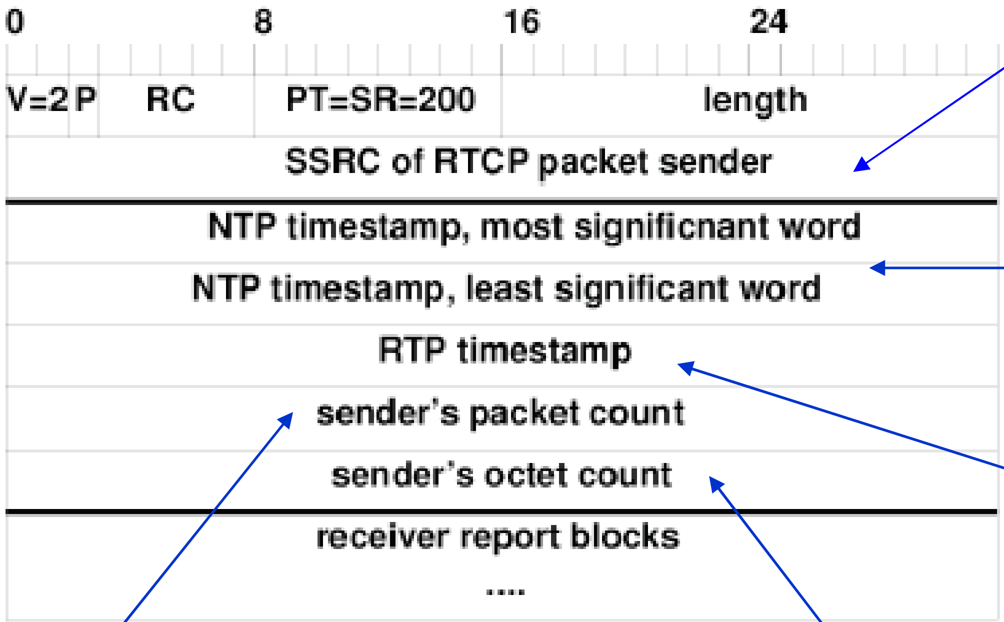
RTCP

- RTCP SDES Packet :
  - Source description items
  - Source identifier item called CNAME is mandatory
  - CNAME – canonical name
  - Receivers map SSRC and CNAME
    - Note : SSRC can change but CNAME is persistent
    - CNAME also required by receivers to associate multiple streams from a single participant (eg synchronize auido + video)

  - Other descriptions items include
    - NAME : of user
    - EMAIL : of user
    - PHONE : number of user
    - LOC : geographic user location, depends on application
    - TOOL : name of application generating the stream

RTCP

- RTCP Sender Report Packet



SSRC ID for the originator of this Sender Report

Network Time Protocol timestamp (wallclock time) when this report was sent

Associated RTP time stamp (same instant as NTP timestamp).

Used for synch at the receiver

Total number of RTP packets sent since start of transmission

Total number of payload octets sent since start of transmission

RTCP

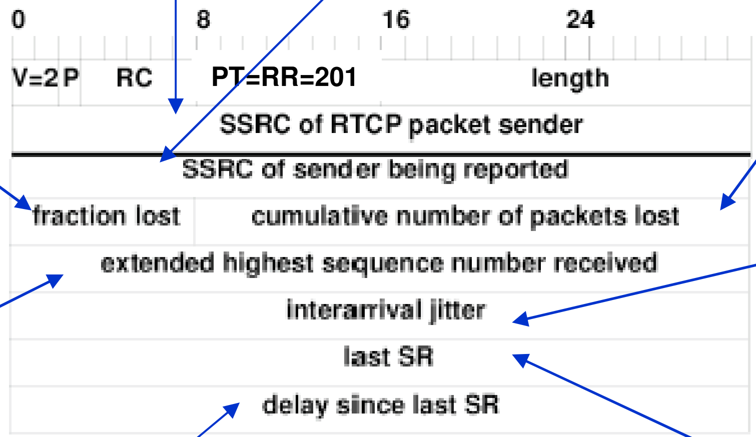
- RTCP Receiver Report Packet

SSRC ID for the originator of this RR

SSRC ID of sender

Fraction of RTP packets lost since last Receiver Report

Total number of RTP packets lost since start of transmission



Highest RTP Sequence number of packets received

RTP packet inter-arrival time

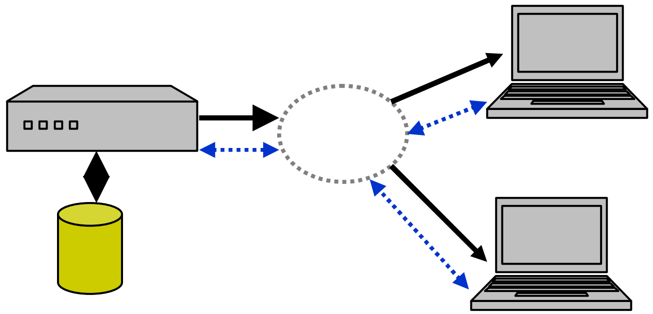
Delay between receiving Sender Report and sending this Receiver Report

RTP timestamp of last Sender Report received from source

RTCP

- RTCP Bandwidth Utilization

If transmission rate of
RTCP packets were fixed
then RTCP traffic would
grow linearly with number of receivers

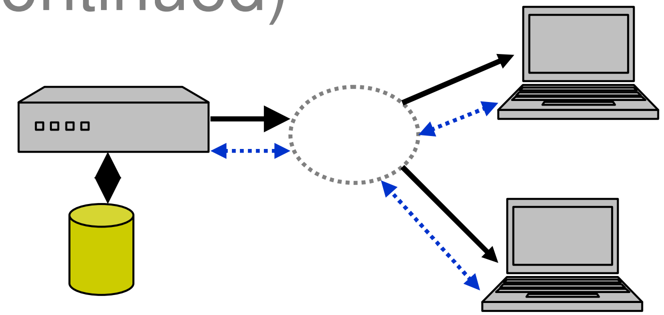


- RTCP rate dependents on the number of participants
  - Increase in participants (eg receivers) then decrease rate
  - Need to keep control traffic to a small fraction at all times
    - So as not to impair the primary function of data transport
  - Need to have common known rules to follow in calculating the bandwidth used for control traffic
    - So that each participant can independently calculate its share of the control bandwidth

RTCP

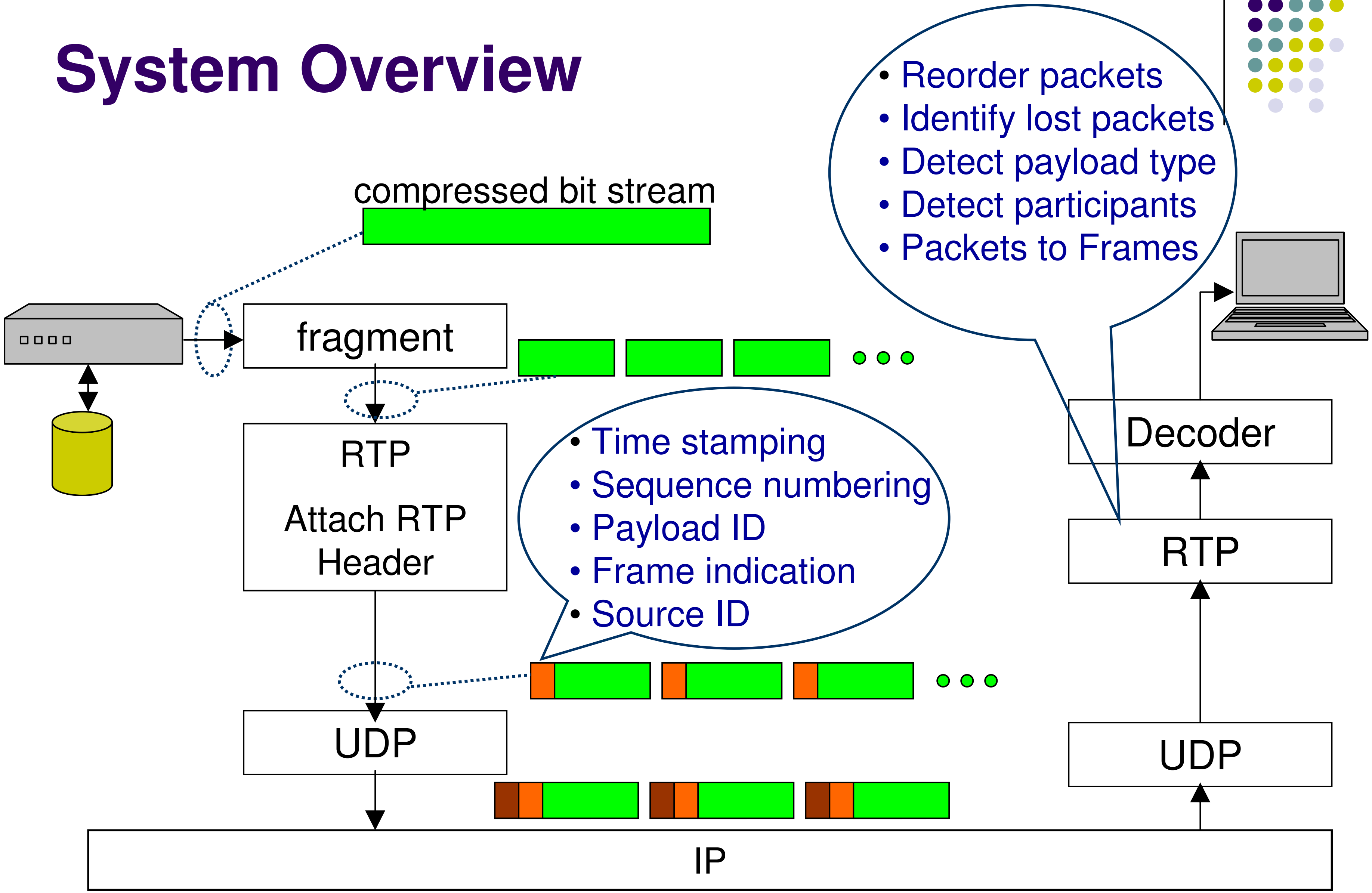
- RTCP Bandwidth Utilization (continued)

  If transmission rate of RTCP packets were fixed then RTCP traffic would grow linearly with number of receivers



- RTCP rate dependents on the number of participants
  - Try to keep control traffic to 5% session bandwidth
  - Senders collectively allocated 25% of control bandwidth
  - Receivers equally share the remaining 75% of control bandwidth
  - Interval between RTCP packets is varied randomly
    - To avoid unintended synchronization of all participants.
    - Varied [0.5,1.5] x interval
  - Interval between RTCP packets is required to be > 5 s

System Overview



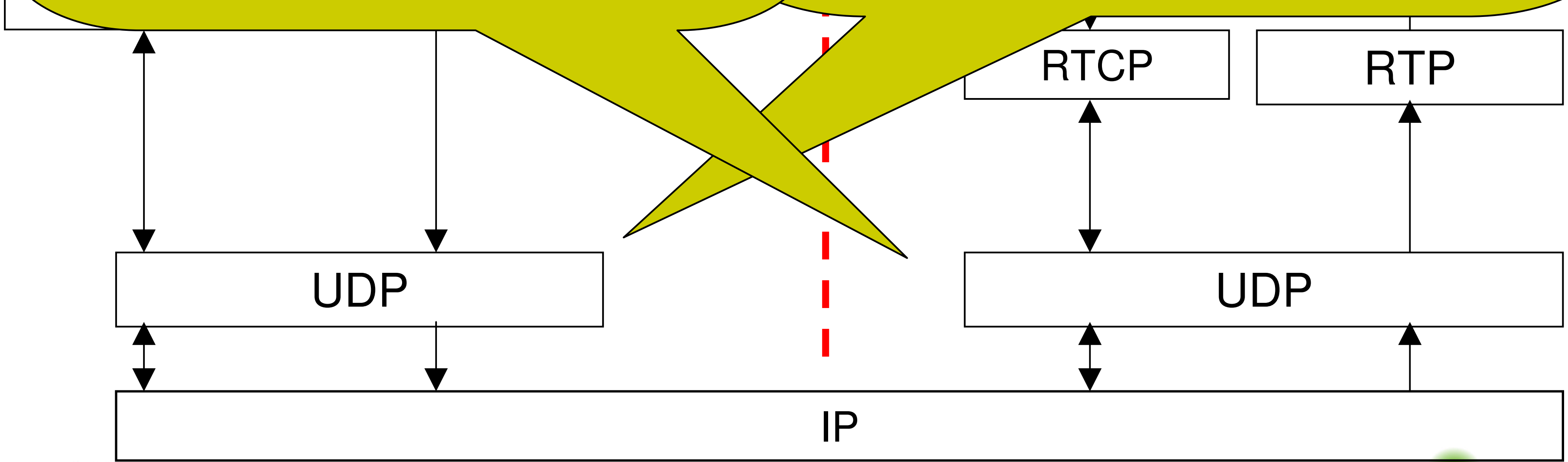
System Overview

Receiver Side Issues

- -Buffering input data
- -Error concealment
- -Inter media synchronization

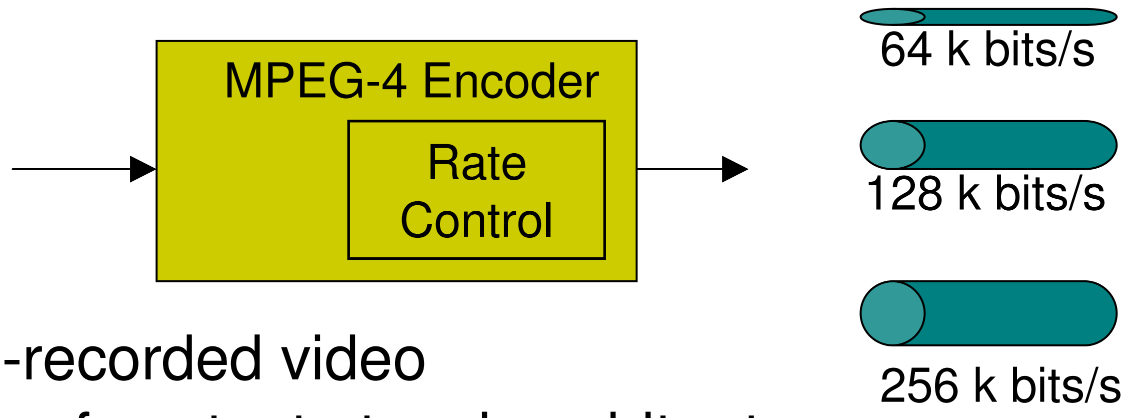
der Side Issues

- -Conforming to Network Bandwidth
- -Adaptive Rate Control
- -Error Resilience
- -Packetization strategy

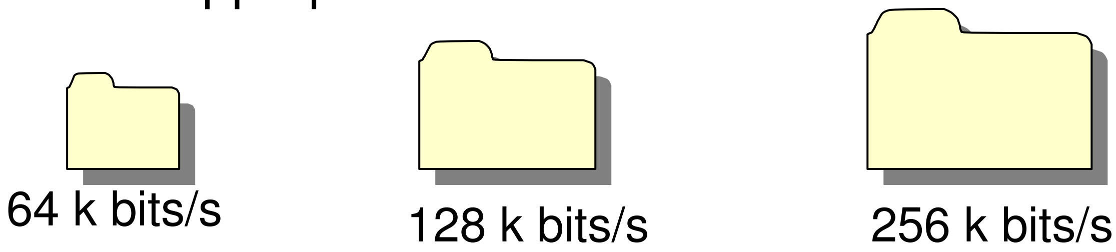


System Overview : Sender Side Issues

- Conforming to Network Bandwidth
  - Example : live video encoding
  - Rate Control to match available bandwidth

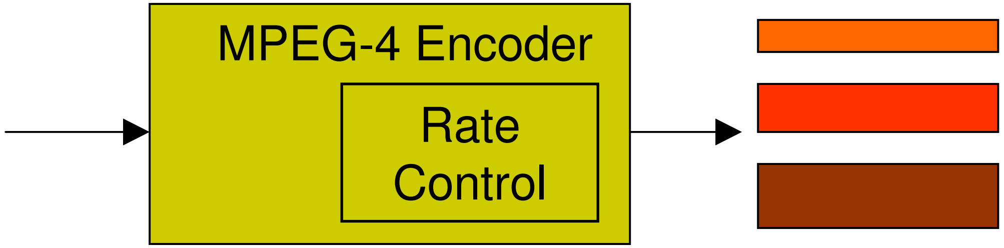


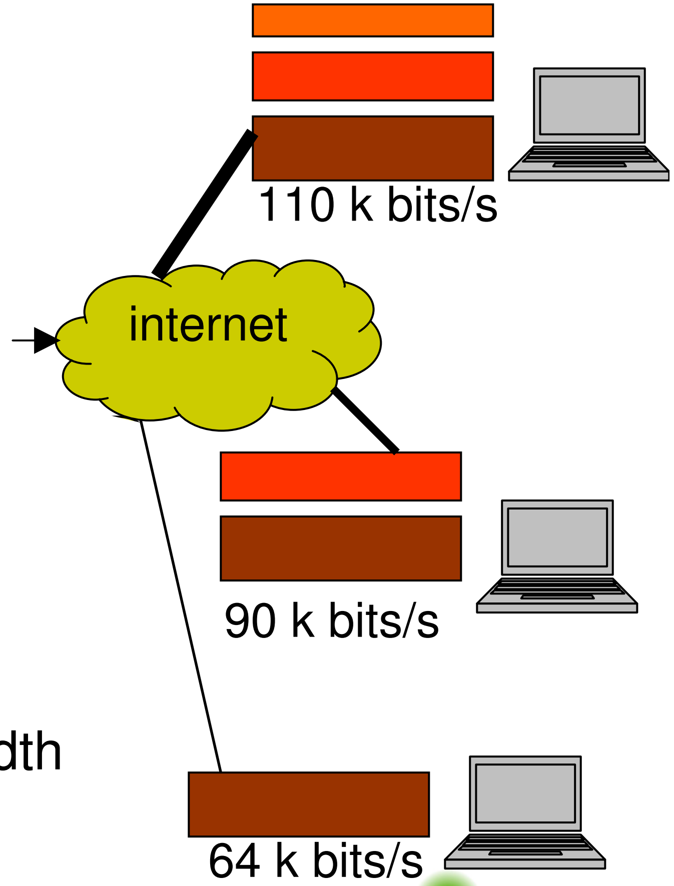
  - Example : Pre-recorded video
  - Multiple copies of content at various bit rates
    - Choose appropriate version to match bandwidth



System Overview : Sender Side Issues

- Conforming to Network Bandwidth
  - For multiple participants with different bandwidth access
  - Scalable coding is an option
  - Example : 3 layer coding
    - Base Layer + 2 Enhancement layers





    - MPEG-4 scalability options
      Temporal, SNR, Spatial
    - Rate control for each layer
      to ensure combined rates meet bandwidth

System Overview : Sender Side Issues

- Adaptive Rate Control
  - Even though rate control meets nominal bandwidth rates
  - Fluctuations of available network bandwidth still occur

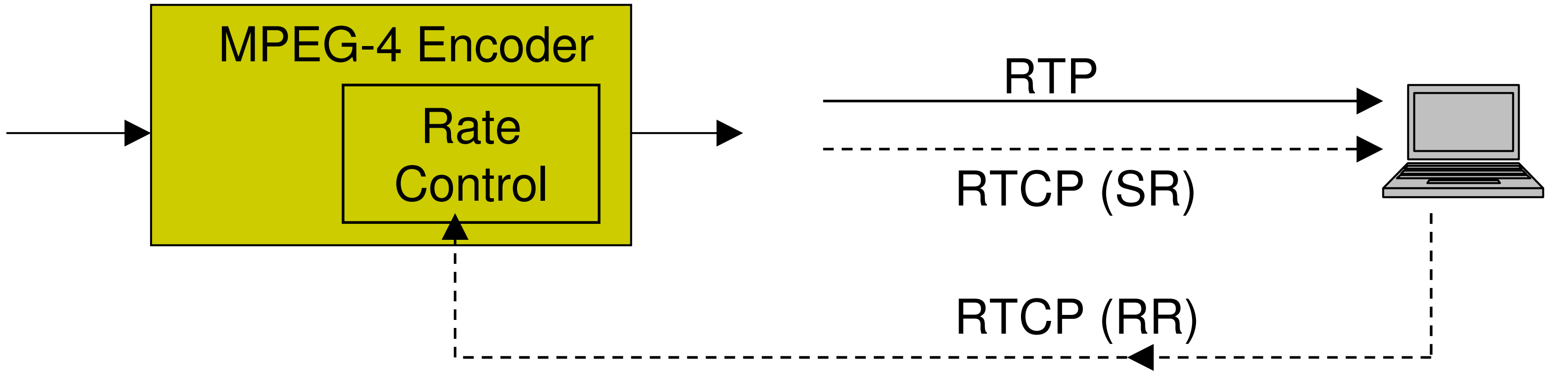
  - Schemes to dynamically control output rate are required
    - Make rate control dynamically adjust to network conditions
    - RTCP provides some feedback about network
    - Therefore can formulate adaptive rate control using RTCP

    - Example :

      [a] " *On End-to-End Transport Architecture for MPEG-4 Video Streaming over the Internet* " , D Wu et al, IEEE CSVT, Vol. 10, No. 6, September 2000

System Overview : Sender Side Issues

- Adaptive Rate Control



  - Use packet loss ratio returned in the RTCP packet
  - This is the packets loss experienced by the receiver since the last RTCP packet

IF small loss ratio
    increase output rate
ELSE
    reduce output rate

IF ($P_{loss} <= P_{threshold}$)
    $r = min\{ (r+\Delta), Peak\_Rate \}$
ELSE
    $r = max\{ (\alpha \times r), Min\_Rate \}$

System Overview : *Sender Side Issues*

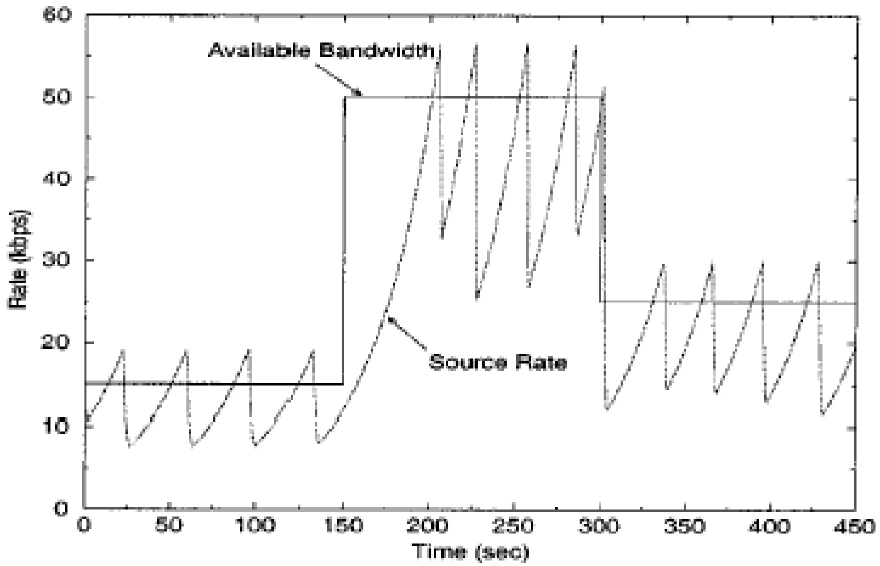
- Adaptive Rate Control

IF ($P_{loss} <= P_{threshold}$)
    $r = min\{ (r+\Delta), Peak\_Rate \}$
ELSE
    $r = max\{ (\alpha \times r), Min\_Rate \}$

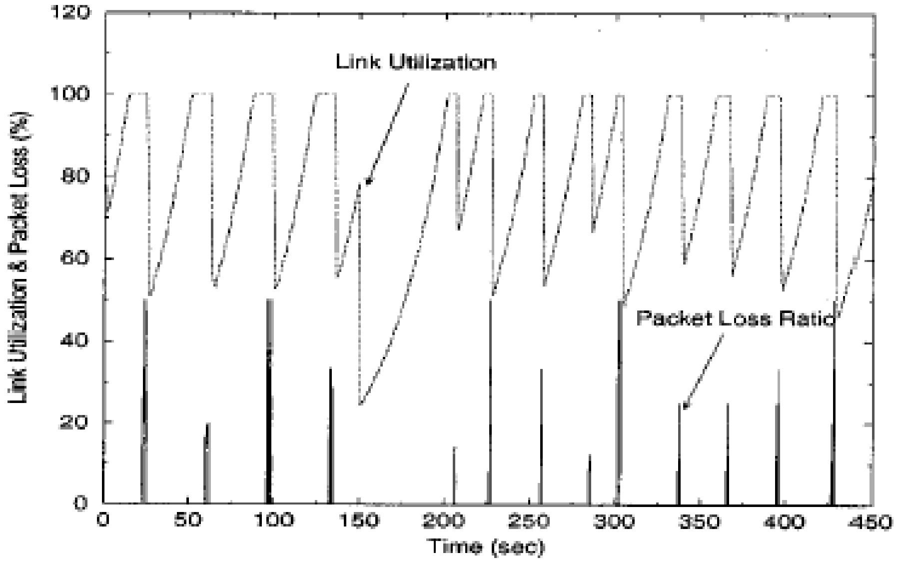
- Trying to adjust output rate (*r*) to maintain packet loss ratio (P_{loss}) below a threshold ($P_{threshold}$)
- Additive increase in rate – conservative increase
  - If there is no congestion and hence low loss ratio
- Multiplicative decrease in rate – fast adaptation
  - Swift reduction necessary to shorten congestion period and reduce packet loss.

System Overview : Sender Side Issues

- Adaptive Rate Control
- Results from [a] for peer-to-peer network.



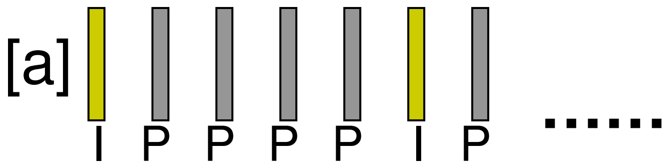
Source rate and link capacity
Rate (kbps) vs Time (s)

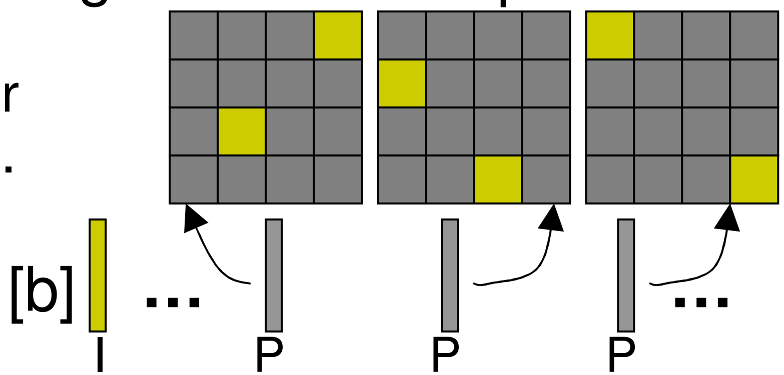


Link utilization and packet loss ratio
Percentage(%) vs Time (s)

System Overview : *Sender Side Issues*

- Error Resilience
  - Packet loss causes degradation of perceptual quality at the receiver
  - Need to ensure source coding includes mechanisms to be resilient to error
  - For MPEG-4 examples of limiting the effects of packet loss include
    - [a] regular intra frame coding or
    - [b] intra block refresh schemes.





    - Intra frame stops the propagation of error
    - Intra block refresh schemes also limit the propagation of error but with a “smoother” bit rate allocation among the frames

System Overview : *Sender Side Issues*

- Packetization Strategy

- Must meet network MTU limits
  - Maximum Transmission Unit
  - Minimum MTU of all links from sender to receiver

- Minimize number of packets required
  - Thereby minimizing overhead
  - Each packet overhead – IP, UDP and RTP header

- Minimize dependency between consecutive packets
  - For better error resilience

System Overview : *Sender Side Issues*

- Packetization Strategy

- Must meet network MTU limits
  - Example : MTU = 1500 bytes
  - Packets larger than MTU will result in IP fragmentation
    - Resulting in overhead for each fragmented packet
    - Loss of one fragment will corrupt the whole packet

  - MPEG-4 : Try to allocate a single VOP (frame) per packet
    - If packet is greater than MTU then break into multiple packets
    - Can break any where - arbitrary byte positions
    - Best to break at GOB or slice boundaries

System Overview : *Sender Side Issues*

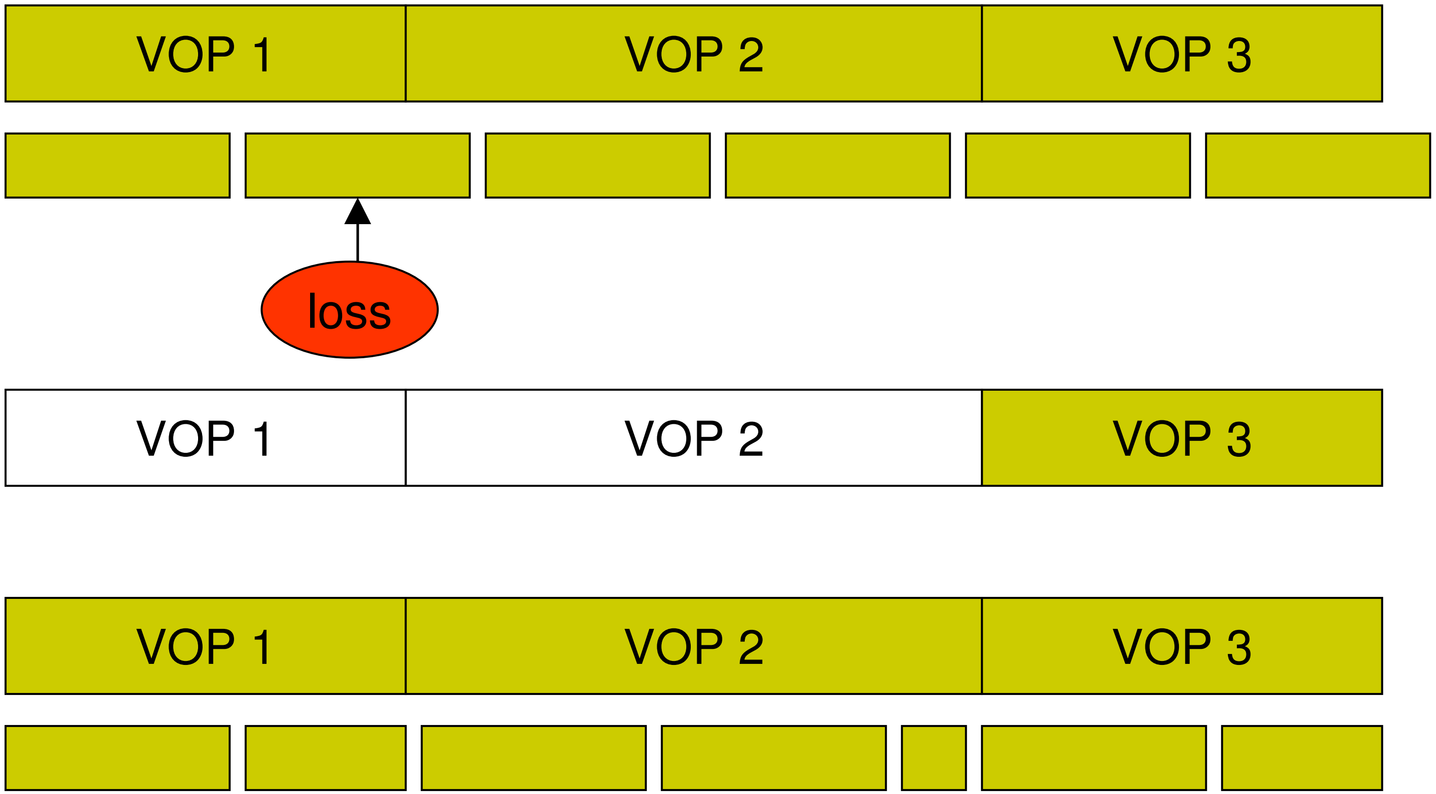
- Packetization Strategy

- Minimize number of packets required
  - Less packets means less bits spend on packet headers
  - Example :
    - 20 byte IP Header
    - 8 byte UDP Header
    - 12 byte RTP Header

- Minimize dependency between packets
  - So that even if one packet is lost, the next packet can still be decoded
  - Example :
    - MPEG-4 – use VOP boundaries to start / stop packets

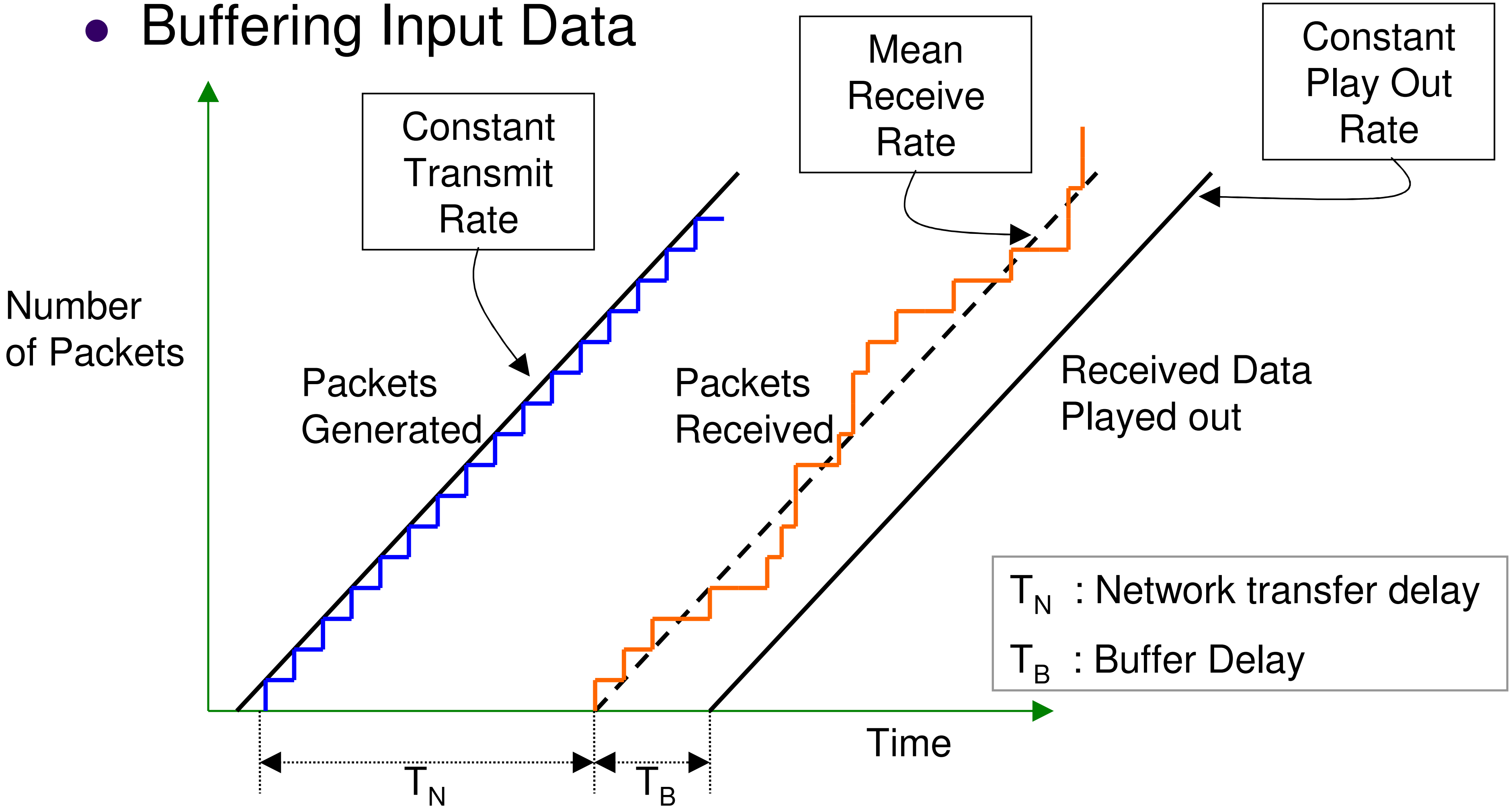
System Overview : Sender Side Issues

- Packetization Strategy
  - Minimize dependency between packets (continued)



System Overview : *Receiver Side Issues*

- Buffering Input Data



System Overview : *Receiver Side Issues*

- Buffering Input Data

- Jitter can be removed by buffering received packets

- Receiver plays out packets $T_N + T_B$ (milliseconds) after generation of the packet at the sender

- There exists a tradeoff between delay (due to buffering) and packet loss
  - Packets received late (> $T_N + T_B$) are ignored (i.e. lost)
  - Live streaming applications have more stringent requirements on delay – so this limits the amount of buffering

System Overview : *Receiver Side Issues*

- Error Concealment

- A streaming service will always be affected by lost packets.

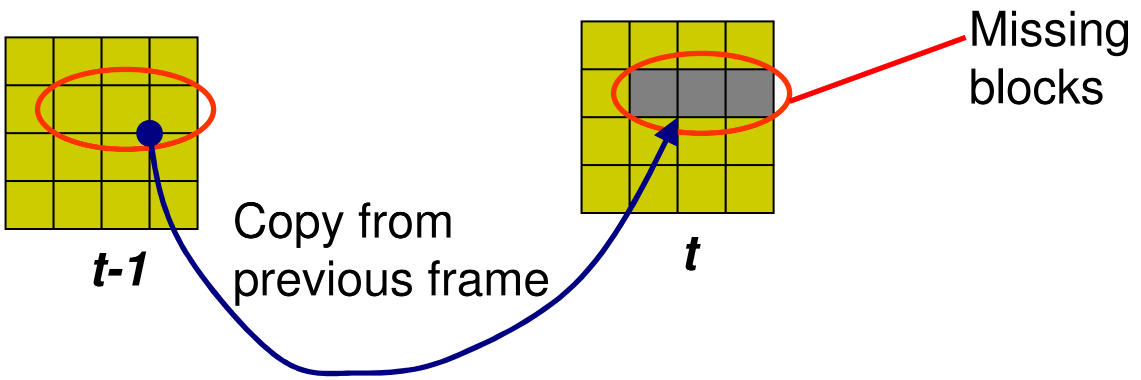
- How to conceal errors due to missing packets ?
  - So as to minimize loss of quality
    - Reduce impact to user perceived quality of image, video or audio service

  - Error Resilience
    - Techniques employed at the receiver to conceal errors
    - These techniques may vary from receiver to receiver

System Overview : Receiver Side Issues

- Error Concealment

- Example : MPEG-4
  - One possibility is to replace a missing group of blocks / slice of the current frame with group of blocks / slice from the previous frame.



  - Not part of the MPEG-4 Standard
    - Various techniques proposed and used by decoder developers

***System Overview** : Receiver Side Issues*

- Inter Media Synchronization

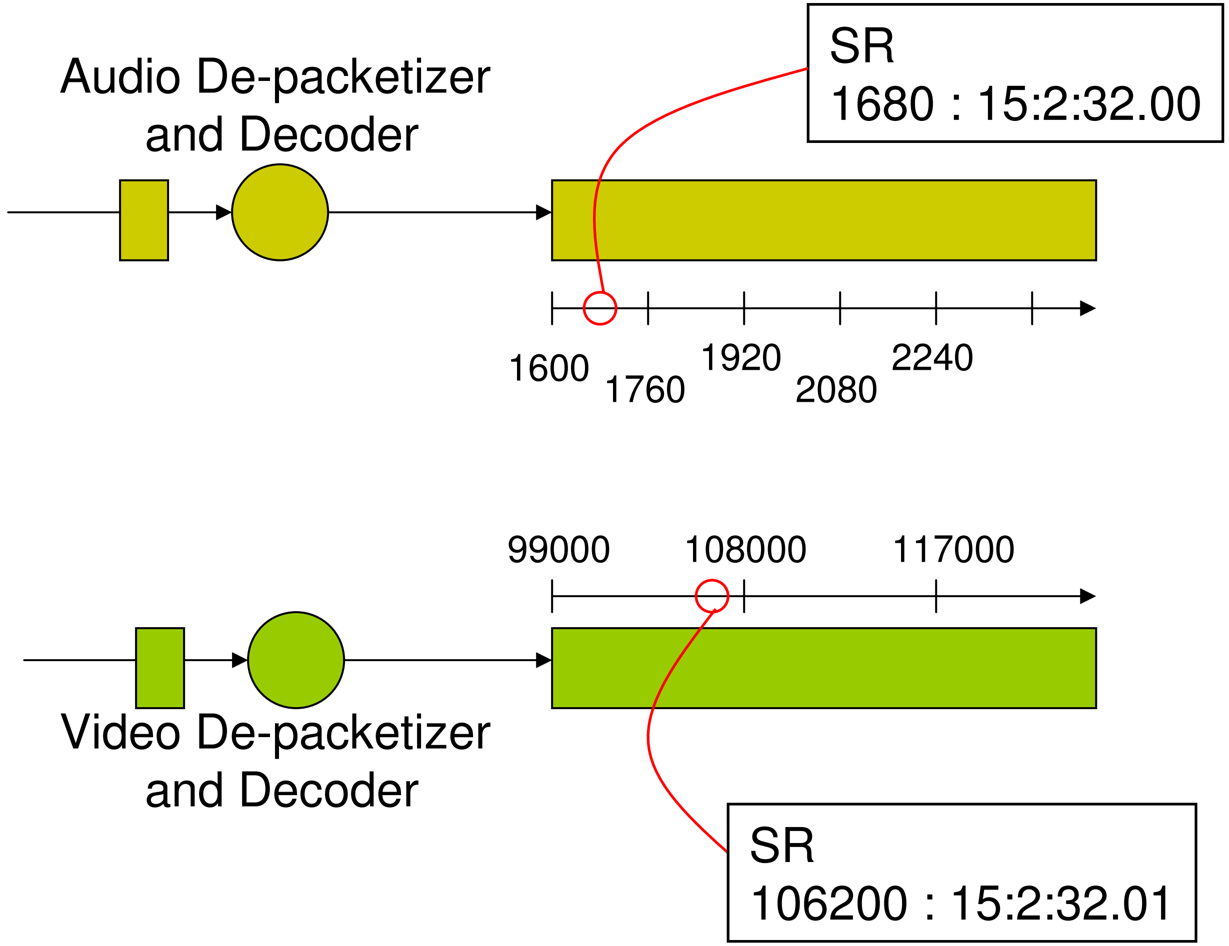
- Need to synchronize multiple media streams
  - Synchronize video and audio streams – lip synch
  - Synchronize video, audio and text

- RTP time stamps
  - Start at a random number
    - random offset for each stream
  - May advance at different rates
    - Audio and video clock rates could be different

- To synchronize streams at the receiver
  - Requires information sent in the RTCP SR (sender reports)

System Overview : *Receiver Side Issues*

- Inter Media Synchronization



Audio clock = 8000Hz
(eg u-law)

1760 = 32.00 + 0.01
1920 = 32.00 + 0.03

2240 = 32.00 + 0.07

Video clock = 90000Hz
(eg MPEG-4)

108000 = 32.01 + 0.02
117000 = 32.01 + 0.12

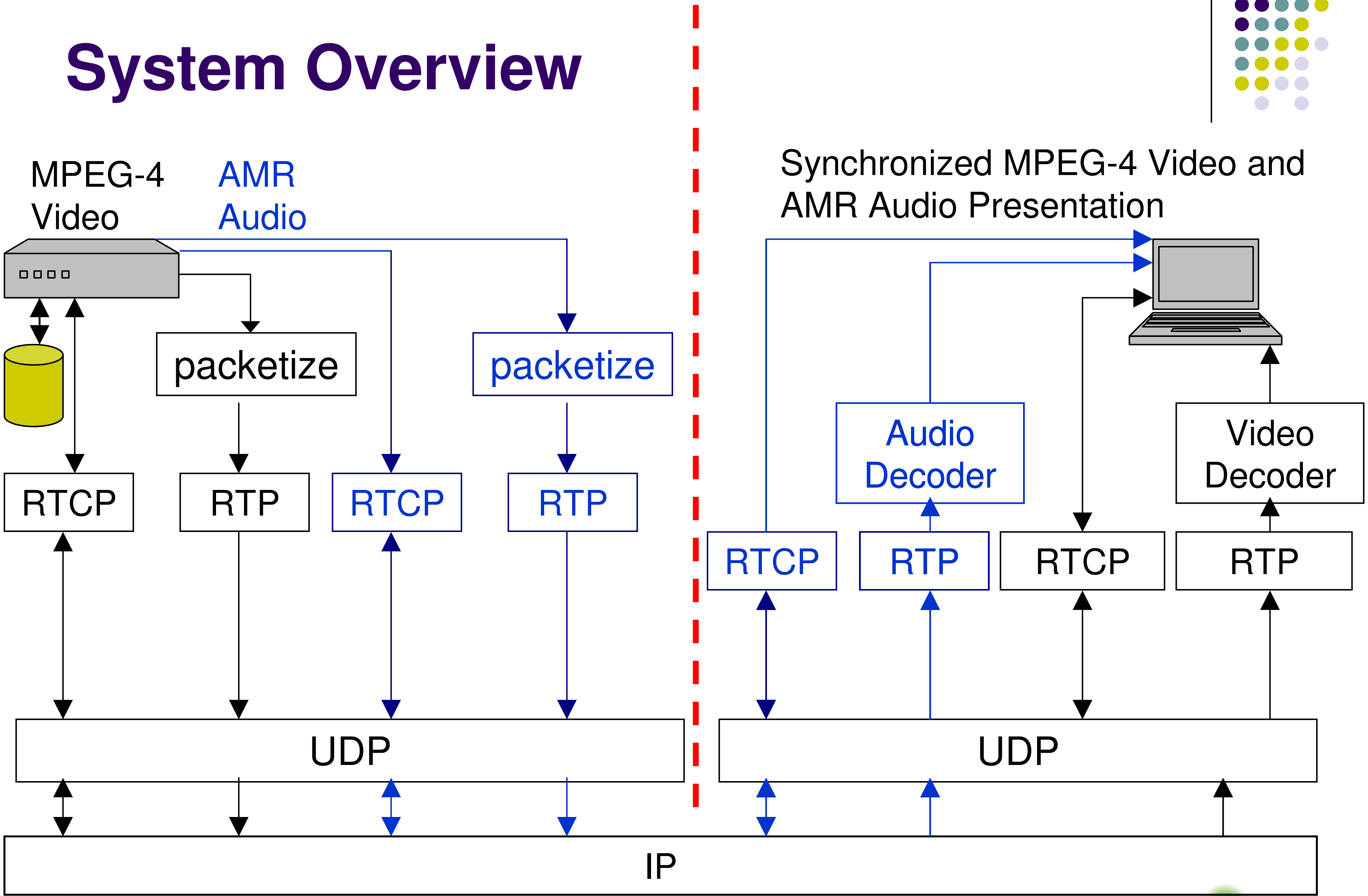
System Overview : *Receiver Side Issues*

- Inter Media Synchronization

- RTCP SR
  - Maps RTP timestamps with the NTP time stamps
  - Relates to the instance when the SR is sent
  - More information on Network Time Protocol

- Synchronizing audio with Video
  - Must meet human perception limits
  - Example :
    - Audio can lead video by 40 ms
    - Video can lead audio by 120 ms

System Overview



System Overview

- Inter Media Synchronization

- Separate RTP & RTCP streams for video and audio
- Synchronized display at receiver
  - Both streams part of one session

- Information on AMR Audio
  - IETF, Request for Comments: 3267
    - *"Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs"*

    - ftp://ftp.rfc-editor.org/in-notes/rfc3267.txt

References and Further Reading

- RTP / RTCP : IETF, RFC 3350 **RTP: A Transport Protocol for Real-Time Applications**
- ***Streaming Video over the Internet: Approaches and Directions****,* Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, MARCH 2001*
- **On End-to-End Transport Architecture for MPEG-4 Video Streaming over the Internet** , D Wu et al, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, Vol. 10, No. 6, September 2000

What's Next ?

- The story's not over !

- How to describe content and session info to client
  - Session Description Protocol

- Internet "VCR Control"
  - How to stop, pause, play ?
  - Real Time Streaming Protocol

  - Extending the streaming system

Acknowledgements

- Thanks to Dr Matthew Leditschke for reviewing slides and for the helpful comments and discussions