Advanced Networks

(EENGM4211)

Rasheed Hussain

rasheed.hussain@bristol.ac.uk

Best contact : Microsoft Team

Slides courtesy: Prof. Reza Nejabati and Slides from Ross's Book

Unit Outline

1. Introduction
2. Internet Routing and Switching
3. IP Multicast
4. Networking for Realtime Applications

University of BRISTOL

Part 4:
Networking for Real-Time Applications

Multimedia Networking Applications

Classes of MM applications:

1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video

Jitter is the variability of packet delays within the same packet stream

Fundamental characteristics:

- Typically **delay sensitive**
  - end-to-end delay
  - delay jitter
- But **loss tolerant**: infrequent losses cause minor glitches
- Antithesis of data, which are *loss intolerant but delay tolerant*.

Streaming Live Multimedia

Examples:

- Internet radio talk show
- Live sporting event

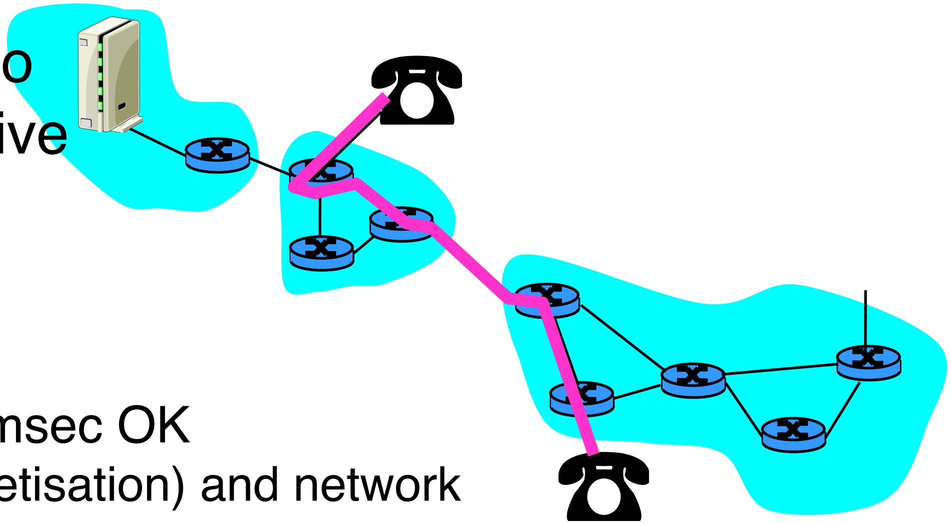
Streaming:

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity:

- fast forward impossible
- rewind, pause possible!

Interactive, Real-Time Multimedia



Applications: IP telephony, video conference, distributed interactive worlds

End-end delay requirements:

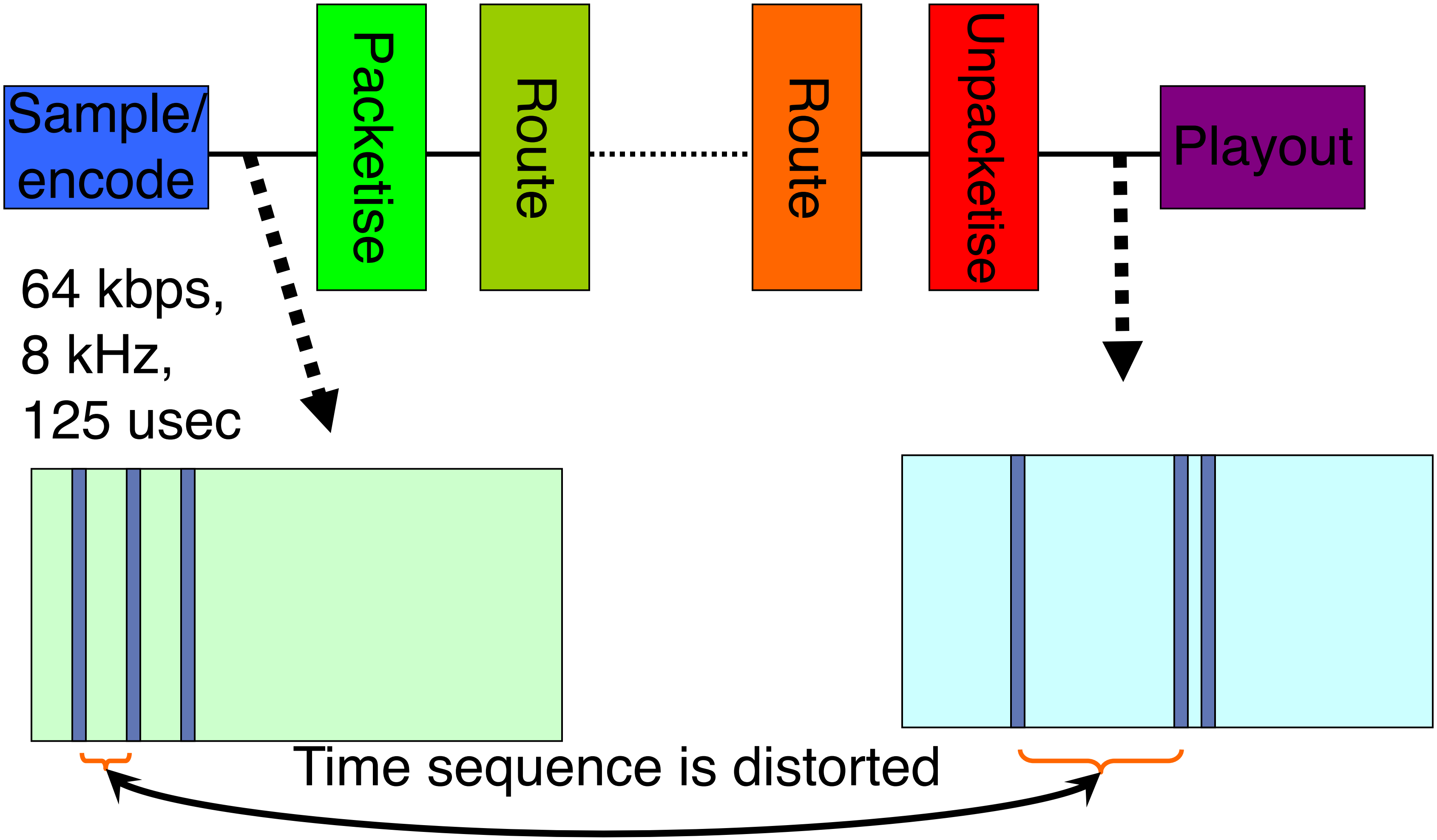
- audio: < 150 msec good, < 400 msec OK
  - includes application-level (packetisation) and network delays
  - higher delays noticeable, impair interactivity

Session initialisation

- how does callee advertise its IP address, port number, encoding algorithms?

University of BRISTOL

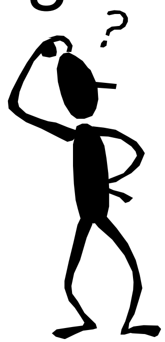
Real Time Delivery over IP



Multimedia Over Today's Internet

TCP/UDP/IP: "best-effort service"

- *no* guarantees on delay, loss



But you said multimedia apps requires QoS and level of performance to be effective!

Today's Internet multimedia applications use application layer techniques to mitigate (as best possible) effects of delay, loss

Real Time Delivery – Synchronicity

- Solutions?
  - Try to share or schedule
    - TDM solution like the telephone system
    - wastes bandwidth
    - too complex
  - Timestamps on everything
    - Receiver reconstructs relationship
    - Uses lots of memory

Reconstruction - Basics

- Average rates must be equal
  - identical at sender and receiver (encode/playout)
  - need estimate of variance
- Delivery delay
  - variable but not unbounded
  - depends on queue sizes of routers
  - most variability < 40ms

User requirements

- Adjust to talkspurt
  - Audio = { talkspurts, silent period }
  - talkspurt is a period of time in which audio data must be acquired and played
- Conferences (high interactivity) require low delay and responsiveness vs. loss and distortion.
- Seminars (low interactivity) require low loss, low distortion but can tolerate delay.

Real Time – Packets and Sessions

- For each packet one needs to know:
  - Format and encoding of content
  - Sequence numbers
  - Timestamps – when should the packet be played?
  - Where (which source) did the packet come from?
    - Q: Why are both timestamps and sequence numbers required?

- For a session one needs to know:
  - Quality of delivery
    - So that one can adapt to network changes, and diagnose faults
  - Who is sending?
  - Who is participating in the session
  - Which sessions are associated
    - So that they can be synchronised.

Real-Time Transport Protocol (RTP) & Application Level Framing (ALF) Principle

(Clark & Tennenhouse 1990)

- Application knows how to process data, therefore

- Data to be sent over the network should be organised into units that fit application requirements best:
  - What is the best network data unit for a stock-update application?
  - What is the most suitable data unit for a video application?
    - (media format, synchronisation requirements, etc.)
  - What is the most suitable data unit for an audio conference application?

- Why do we need another protocol?
  - IP does not provide mechanisms for ensuring reliable and bounded delay delivery; content packets can be lost, re-ordered, delivered too late or too early.
  - Application is closer to content processing than the network.

RTP: Real-Time Transport Protocol

(Schulzrinne et al. RFC 1889 and after)

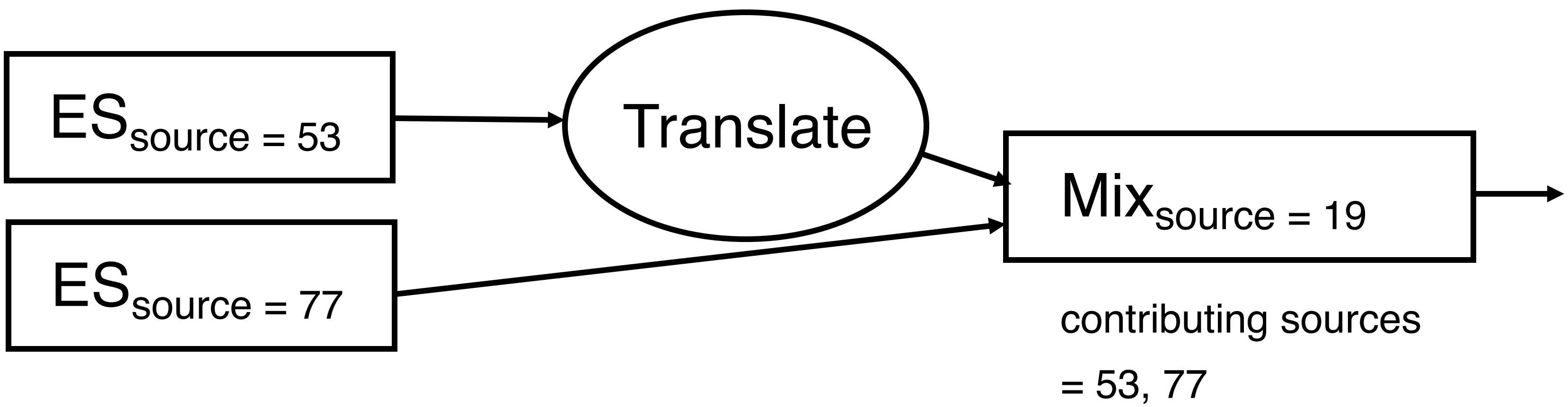
- Product of IETF Audio Video Transport Working Group (AVT WG)
- Functional goals
  - lightweight, interoperable
  - easy integration with application
  - mechanism - not policy
  - scalable - from unicast to multipoint with 2 - 1000s participants
  - separation of control from data
  - adaptable to many types of client media
  - independent of underlying network
    - transported in UDP/IP but not necessarily IP

RTP Sessions Architecture

- Multiple participants
  - May use multicast
- Multiple streams per participant
  - Audio, video, whiteboard, ...
- Dynamic membership
  - Session participants may come and go as they want.
- No central control
  - Participants may not know about each-other.

RTP Entities

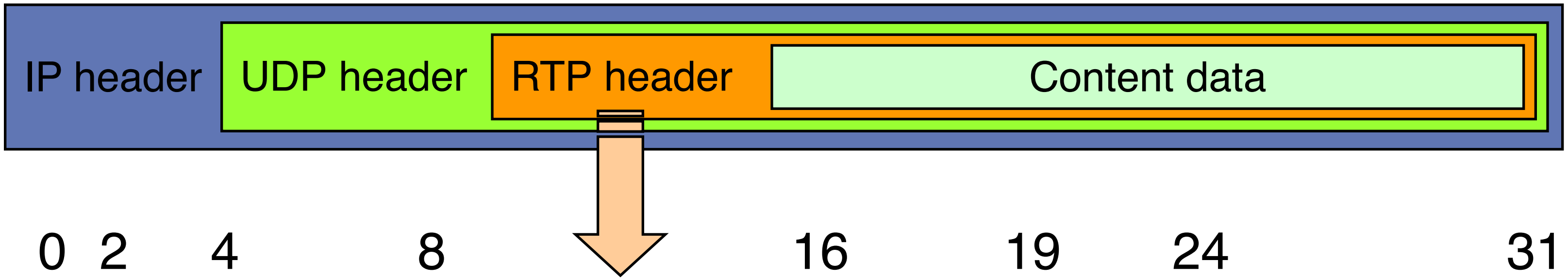
- *End System* (ES) - generates content
- *Mixer* - combines multiple RTP sources into a new RTP packet, re-times and re-formats
- *Translator* - recodes



RTP Functions (1)

- Timing information for playout
- Reordering information
- Loss detection for quality estimation, recovery
- Synchronization
  - network jitter
  - clock drift
  - intermedia (lip sync)
- QoS feedback
- Source identification
- Cannot ensure real-time delivery

RTP Header Format (2)

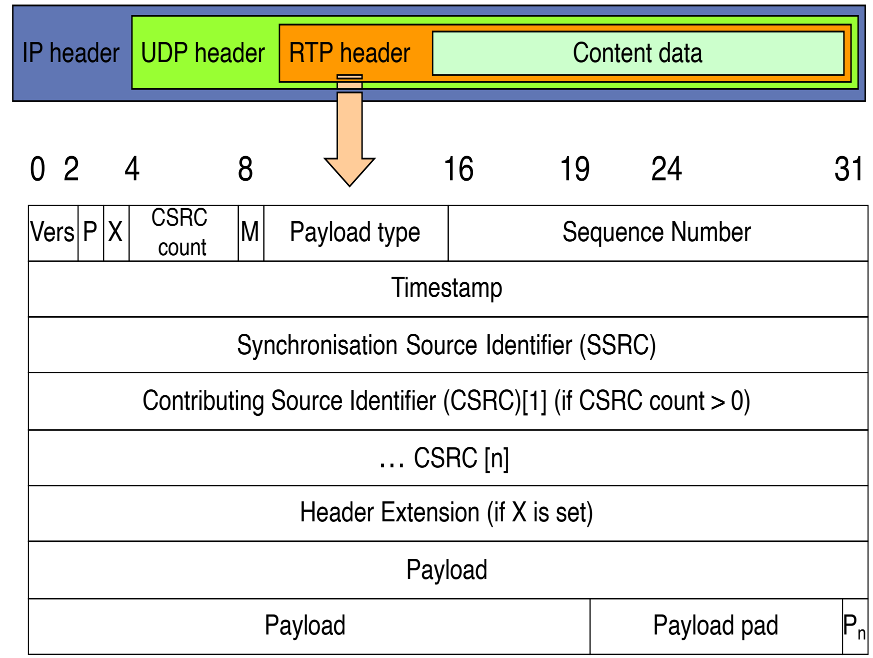


0 2 4 8 16 19 24 31

Vers	P	X	CSRC count	M	Payload type	Sequence Number
Timestamp						
Synchronisation Source Identifier (SSRC)						
Contributing Source Identifier (CSRC)[1] (if CSRC count > 0)						
... CSRC [n]						
Header Extension (if X is set)						
Payload						
Payload					Payload pad	P_n

RTP Header Fields (3)

- Version is always 2 ()
- P: Last field contains padding
  - if lower layer has fixed frame sizes
  - last octet contains size of padding block
- X: Header extension is present
- CSRC count: Number of contributing sources for a mixed stream
- M: Marker used by application to identify significant events, e.g. frame boundaries in a stream



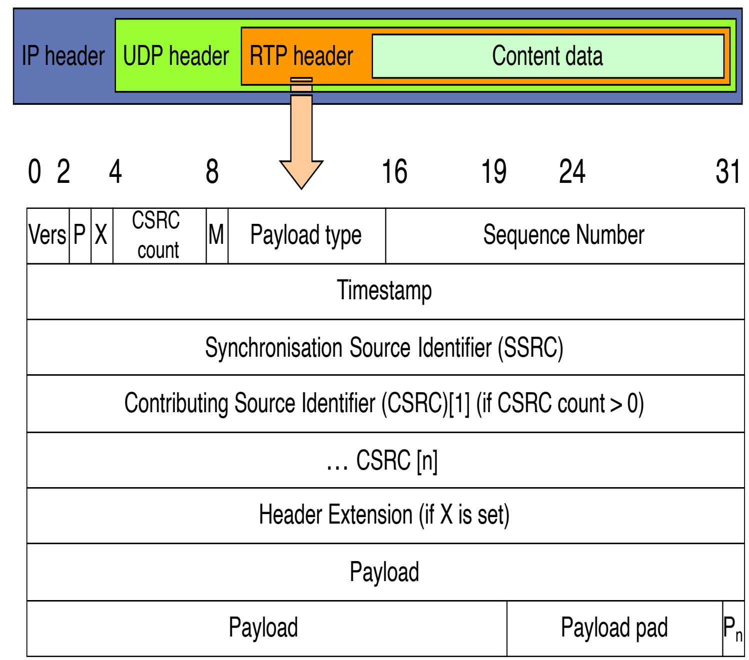
RTP Header Fields (4)

Payload type

- G.721 (2), GSM speech (3), JPEG (26)

Sequence Number

- Increments by 1 for each RTP packet, random initial value, detect loss
- Relate packets to each-other



Timestamp

- Timestamp relates packet to real time
- Timestamp values are sampled from media specific clock!
  - Timestamp = packetisation interval * sampling rate
  - E.g. = 160 for 20ms audio sampled at 8kHz
- fragmented frames have same timestamp
- controls playout rate

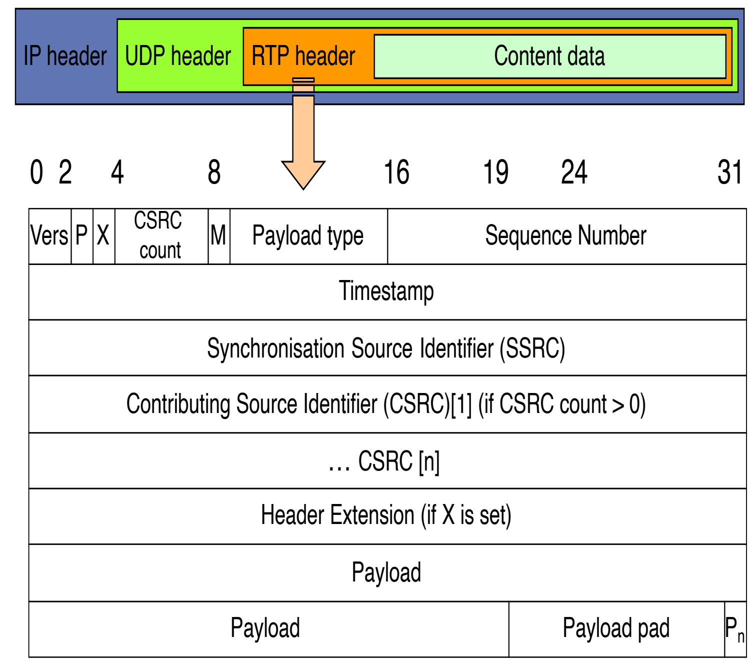
Timestamp vs. Sequence Number

- E.g. video: a frame is send over several packets. All packets will have same timestamp but different sequence numbers.

RTP Header Fields (5)

- Synchronisation source - SSRC
  - Identifies the entity that generated the timestamp and sequence number at the sender
  - It is chosen randomly
  - Identifiers are unique
  - Identifies a stream: all packets with the same SSRC go together.
  - What may be a problem?
    - Collision of source identifiers.

- Contributing source - CSRC
  - It is used by receiver to identify all sources in a stream



RTP Example: Audio / Speech

- Audio samples are packetised every 20msec, timestamped and sent with monotonically increasing sequence numbers.

- What about silence periods?
  - Do not send.
    - Problem: How can the receiving side tell if it is loss of packets or it is a silence?
  - The mechanism (sample, encode, timestamp, packetise, sequence number + 1, send) works in this case as well. How?

RTCP: RTP Control Protocol

- Provides monitoring capabilities
  - quality of routes
  - number/state of participants (talker indication)
  - identification and type of sources
- Feedback to application
  - QoS feedback – sender adjusts rate based on feedback
- Scalable
  - Randomized control traffic : the rate of control messages decreases as number of participants increases

RTCP Messages (1)

- Encoded as RTP messages with pre-defined Payload Type fields
- Sender reports (SR)
  - Real-time of RTP timestamp + sequence number, number of packets and bytes sent
- Receiver reports (RR)
  - proportion and total of packets lost
  - highest Sequence Number
  - jitter (variance in arrival time)
  - time of last Sequence Number and delay since last Sequence Number
- Who uses reports?
  - Applications: sender may modify transmission based on feedback
  - Third-party monitors to locate problems.
- BYE
  - Sent by sources that leave a session.

RTCP Messages (2)

- Source description (SDES)
  - maps SSRC/CSRC values to real objects
  - source types include
    - [user*]domain_name (CNAME)
    - name
    - e-mail address
    - phone number
    - location
    - application name

Synchronization of Streams

- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - timestamp of the RTP packet
  - wall-clock time for when packet was created.
- Receivers can use this association to synchronize the playout of audio and video.

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
- Not tied to the wall-clock time

RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.

Example

- Suppose one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
- RTCP gives 75% of this rate to the receivers; remaining 25% to the sender

- The 75 kbps is equally shared among receivers:
  - With R receivers, each receiver gets to send RTCP traffic at 75/R kbps.
- Sender gets to send RTCP traffic at 25 kbps.
- Participant determines RTCP packet transmission period by calculating avg RTCP packet size (across the entire session) and dividing by allocated rate.

University of BRISTOL

References

- RTP: A Transport Protocol for Real-Time Applications.
  Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. January 1996.
  RFC 1889 (http://www.ietf.org/rfc.html)
- Some Frequently Asked Questions about RTP
  http://www.cs.columbia.edu/~hgs/rtp/faq.html