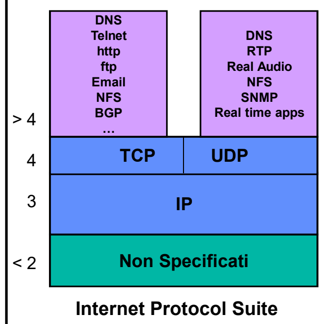
Protocols for multimedia in the Internet

Andrea Bianco
Telecommunication Network Group
firstname.lastname@polito.it
http://www.telematica.polito.it/

Applications and protocol stack

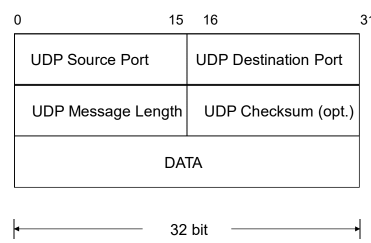


- Which transport protocol?
- UDP is suited for:
  - Request-response (LAN)
  - Multimedia applications
  - Multicast
- TCP (reliability) is suited for:
  - File transfer
  - Terminal emulation
  - Request-response (WAN)
  - Unicast

UDP protocol

- UDP (User Datagram Protocol) permits application to application (host to host) communication through datagram transmission
- UDP provides a layer 4 service:
  - Connectionless (out of sequence packets)
  - Unreliable (packet lost)
  - Low overhead (slim header)
    - Optional checksum
- Application identification through :
  - Source IP address, destination IP address, source UDP port, destination UDP port
  - No rate control
    - No flow control (possible receiver saturation)
    - No congestion control

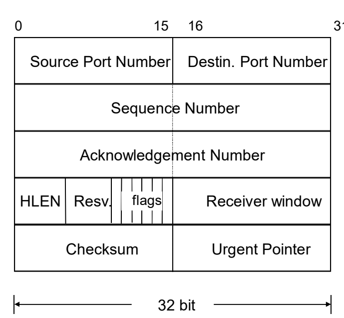
UDP packet format



TCP protocol

- TCP (Transmission Control Protocol) is the other Internet layer 4 protocol
- Main characteristics
  - Connection-oriented
    - full-duplex
  - Reliable and in sequence delivery
    - Retransmission
  - Rate control
    - Flow controlled by receiver
    - Congestion control to avoid network saturation

TCP packet header



Problems for multimedia

- Packetization
  - Voice sample of few bit
  - Single image has very large size
- How to distinguish at the receiver among different coding techniques?
- How to compensate for IP limitations?
  - Packet losses
  - Out of order delivery
  - Packet duplication
- How to notify to the source the correct reception of data?
- How to deal with multicast?

Using TCP for multimedia?

- TCP is reliable, but
  - Retransmissions cause delays
- TCP is rate controlled to avoid receiver and network congestion, but
  - The available bit rate for the multimedia application is highly variable
- TCP does not support multicast
- TCP cannot be used for real-time multimedia
  - Non real time multimedia can be treated as file transfer

UDP for multimedia

- UDP supports multicast, is not rate controlled and does not use retransmissions, but
  - Does not guarantees in-order delivery
  - Does not detect and deals with packet losses
  - Does not compensate for delay fluctuations
  - Does not recognize multimedia contents
- IETF proposal: add RTP protocol over UDP

Example of multimedia application

- IP telephony: three different problems
  - H.323 SIP: Establish multimedia connection, find IP addresses (possibly multicast), negotiate the type of coding and/or compression scheme, possibly inter-operate with the telephone network
  - RTP: Once the connection has been established, transfer audio packets
  - RTCP: Periodically send feedback information to the transmitter (and to receivers) to indicate the quality of the multimedia connection

RTP: Real-time Transport Protocol

- Defined in RFC 1889
  - a framework to build multimedia applications
- Provides some basic mechanisms for multimedia data transfer
- It is not an independent protocol, but must be included in the application (no API defined)
- Composed by two different (although related) protocols:
  - RTP: deals with multimedia data transport (even UDP ports)
  - RTCP: provides control and monitoring services (odd UDP ports)
    - Feedback on packet delay and losses
    - Request/response to coding modifications
    - Helps in the management of the list of participants

RTP + RTCP

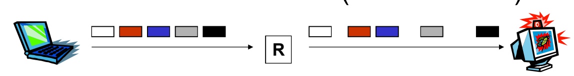
- Functions available
  - Payload identification (coding)
  - Sequence number management
  - Timestamp management
  - Monitoring and performance analysis
  - Participants identification
- Functions NOT available
  - No QoS support
    - No reliability, bit rate or delay guarantees
  - No guarantee on correct and in order delivery
    - Exploits UDP checksum to detect errors

RTP: packet losses

- UDP/IP do not guarantee zero losses
  - Packets may be lost or delivered not in order
- RTP provides sequence numbers in the RTP header to detect out of order delivery

RTP: delay jitter

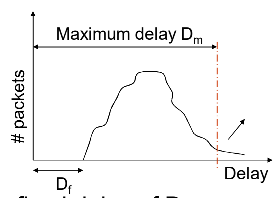
- When the multimedia source sends synchronous signals (e.g., voice), one packet is sent every T seconds (source clock)
- The network introduces variable delays even when there are no losses (router buffers)



- How to recover the synchronous signal at the receiver?

RTP: delay jitter

- Possible solution: introduce a delay at the receiver
- Use a playback buffer
  - Received packets are stored in the buffer
  - One packet every T seconds is played out
  - The buffer size emulates a fixed delay of D_m seconds
  - D_m is a compromise between low delays and low losses



RTP: delay jitter

- If the source uses silence suppression?
  - During talk spurt: one packet every T seconds
  - During silence: no packets
- … a packet may be delayed because
  - the network has delayed it
  - it was preceded by a silence suppressed
- The sequence number is not enough
- A 'timestamp' field (conventional number which increases at the transmitter according to the source clock) is needed in the RTP header to recover information on the generation instant of a packet
  - Timestamp cannot be used as sequence number, since many packets may have the same timestamp (e.g., a video frame generates several packets with the same timestamp)

RTP: applications

- Applications using RTP may be multicast
- If the network does not support multicast addressing and routing
  - A unicast connection among each pair of participants to the multicast session should be opened
  - The number of connections increases quadratically with respect to the number of users
- An application must specify two RTP parameters:
  - RTP profile
    - A table that associates to each type of payload (coding) a unique code
  - How RTP should use the payload
    - Information such as RTP packet size, number of samples contained in each packet, etc

RTP: example of voice transmission

- Consider a multi-user voice conference over RTP
- Basic elements:
  - One IP multicast address
    - Users must register the IP multicast address to access the service
  - Two UDP ports
    - One even port for RTP, the next one (odd) for RTCP
  - RTP does not specify how to choose these numbers
- Voice is produced according to a proper coding technique
- Voice samples are grouped in packets
  - Sent to the multicast address
  - The number of samples in each packet depends on the RTP configuration

RTP: example of voice transmission

- Packets size should be small to keep under control the packetization delay (should be smaller than few tens of ms)
  - Few samples in each packet
- Samples are encapsulated within an IP + UDP + RTP headers

IP header	UDP header	RTP header	Samples

RTP: example of voice transmission

- Through RTCP, each participants sends (in multicast) statistical data
  - It is possible to analyze service performance
    - Code rate adaptation may be envisioned to adapt the transmission to the measured quality
- Since the RTCP traffic is sent in multicast to all participants (and is generated by all participants) the required bit rate may be significant and should be kept under control

RTP: example of audio-video conference

- The standard suggests to use two independent RTP flows
  - Advantage: a user may access only one of the two services
- It is mandatory to synchronize the two flows:
  - The RTP timestamps of the two flows may be used with this goal

RTP: terms and definitions

- An RTP **session**
  - Allows the communication among a set of users through RTP
  - Is identified by a pair of transport addresses
    - A transport address is a IP address-UDP port pair
    - One transport address is used by RTP, the other one for RTCP
    - The IP multicast address (if used) is the same for both pairs
- The **host** or **end-system** (ES) is the user system where applications based on RTP are running. Hosts generate and receive multimedia flows

RTP: terms and definitions

- An RTP **translator** is a coding translator
  - Modify the data coding and retransmit the modified data flow over the session
  - May also operate as gateways
  - Permit the service provisioning also over non IP islands
- An RTP **mixer** is an RTP flow aggregator
  - More RTP flows in input are mixed and generate a new single RTP flow at output
    - Typically, the bit rate of the new flow is smaller that the sum of the bit rate of original flows
  - Devices more complex than translator
    - Correlated flow synchronization must be dealt with

RTP: terms and definitions

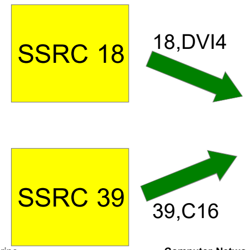
- RTP **monitors** observe externally the control packet flow (RTCP packets)
  - Do not participate in the RTP transmission/reception process
  - Collect information on the QoS of the RTP session
  - Useful for network providers to control the network service quality

RTP: terms and definitions

- The **SSRC** (Synchronization SouRCe) is the unique identifier of the data flow generator
  - Is a 32 bit number contained in the RTP header
  - Warning: mixers are characterized by an own SSRC
    - The output data flow from a mixer is a new flow, with its proper timestamp
  - Within an RTP session, each SSRC must be unique
- Since it may be useful or needed to recover the original source of a mixed RTP flow
  - The **CSRC** (Contributing SouRCe) are fields optionally contained in the RTP header that contain the SSRCs of the original sources of the RTP mixed flow

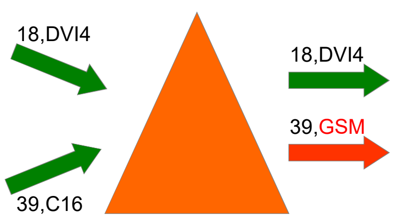
RTP: Host

- The first host creates an RTP flow with SSRC 18 and DVI4 code, the second host sends RTP data with SSRC 39 and C16 code



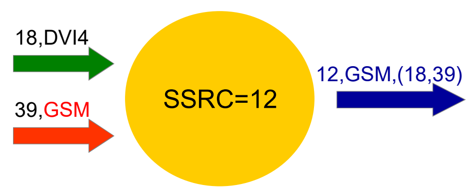
RTP: Translator

- The translator modifies the coding of the flow with SSRC 39 from C16 to GSM

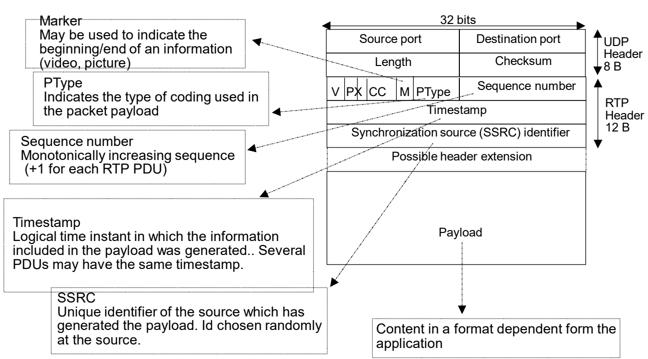


RTP: Mixer

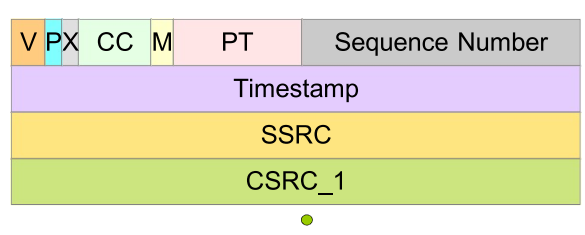
- The RTP flow generated by the mixer has SSRC 12 and CSRC 18 e 39



RTP packet format



RTP header



RTP header

- V (Version) (2bit)
  - Version of the RTP protocol
- P (padding) (1bit)
  - Padding contained in the payload
- X (eXtension header) (1bit)
  - Proprietary
- CC (CSRC count) (4 bit)
  - Number of CSRC field contained in the header
- Marker (1 bit)
  - Use dependent on the RTP session profile
- PT (Payload Type) (7 bit)
  - Type of coding used in the packet payload

RTP header

- Sequence number (16 bit)
  - The initial sequence number X is chosen randomly at session startup
  - X is inserted in the sequence number field of the first generated RTP packet
  - The second packet will have sequence number X+1, the third X+2...
    - The random extraction minimizes the probability of choosing the same number previously selected in an older RTP session (which may have some packets still in the network)

RTP header

- Timestamp (32 bit)
  - Represents the “logical” time instant in which the packet payload was generated. More packets my have the same timestamp value
  - The first timestamp value is randomly selected at RTP session startup
  - The timestamp refers to the creation time instant of the first sample contained in the packet
  - Example: if each packet of an RTP phone session contains 160 samples:
    - If packet I has timestamp X, then packet I+1 will have timestamp X+160

RTP header

- SSRC (32 bit)
  - Identifier of the source that has created the RTP payload. Randomly chosen at the source at session startup. Algorithm defined to solve contentions.
- CSRC (32 bit)
  - Optional field
  - Identifier of the sources that have originally created the payload from which the current payload was derived through a mixing operation

RTCP: objectives

- Quality of service and congestion control
  - RTCP packets are used as “low frequency” ACKs to signal the reception quality
  - On the basis of RTCP “report”, the server may adapt the coding to the communication status
- Identification
  - Provide identification information on RTP session participants (signalling)
- Estimate the number of participants to the multicast session
  - Needed to control the transmission bit rate required by RTCP control signals which would increase too much if the number of participants increases too much

RTP Control Protocol (RTCP)

- Control protocol for RTP data flow
- Defines the information exchange among the source and the destinations
- Several types of RTCP packets:
  - SR (Sender Report): sent by all active sources to all participants; includes TX and RX statistics
  - RR (Receiver Report): sent by all receivers to all participants; includes RX statistics
  - SDES (Source DEScriptor): source description through a unique identifier
  - BYE: session ends or one participant leaves the session
  - APP: application-specific

RTCP Sender Report

- SR is used to provide information on data recently sent
- A SR contains:
  - An absolute timestamp (NTP timestamp) of the data sending time
  - Relative timestamp referring to the current RT flow
  - Amount of data sent from RTP session start-up
    - Total number of RTP packets sent
    - Total number of byte sent

RTCP Receiver Report

- RR are sent to inform senders on the quality of the RTP session as seen by receivers
- A RR is sent to each source from which a SR was received
- A RR contains:
  - Identification of the received source
  - Timestamp of the last received SR
  - Delay from the reception of the last SR
  - Highest sequence number received from the source
  - Number of lost RTP packets
  - Fraction of lost RTP packets
  - Estimate of RTP packets jitter

RTCP SDES

- Used by sources and destinations to identify themselves
- An SDES may contain:
  - CNAME: user identifier (user@host.domain)
  - NAME: name of the person using the application
  - EMAIL
  - PHONE
  - LOC: geographical location of the user
  - TOOL: application using RTP
  - NOTES

RTCP: report transmission speed

- Four rules must be followed to generate RTCP packets
  - The RTCP traffic should be limited to a given percentage of the data traffic (5%)
  - 25% of the session bit rate is devoted to SR packets, the remaining part to other packets
  - An RTCP packet cannot be sent earlier than 5s after the previous RTCP packet transmission
  - A variable time P should be added to the waiting time

RTCP: report transmission speed

- P is computed as a random number uniformly chosen between 0.5 e 15, multiplied by T_{sr}
- The period for transmitting RTCP packets for the transmitter

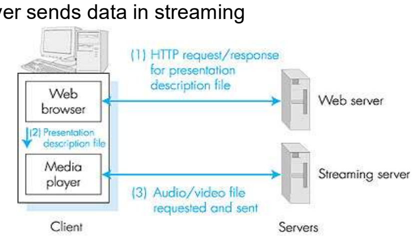
$$T_{SR} = \frac{\text{Num\_senders}}{0.25 \times 0.05 \times \text{Session\_rate}} \times \text{Avr\_RTCP\_pkt\_size}$$

Multimedia traffic

- Interactive
  - IP telephony
    - Delay management
- Streaming
  - RTSP
    - Playback control

Multimedia: streaming approach

- The client browser receives the metafile containing the multimedia streaming file description
- The browser passes the metafile to the player
- The player contacts the streaming server
- The server sends data in streaming



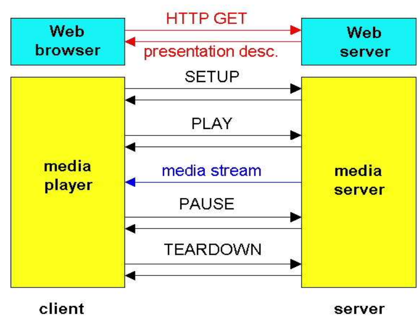
User control of streaming multimedia: RTSP protocol

- Real Time Streaming Protocol (RTSP): RFC 2326
  - Supports user control: rewind, FF, pause, resume
  - Out-of-band protocol:
    - Exploits port 544 for control and signalling messages
    - Another port for multimedia stream
  - Exploits either TCP or UDP for control connections
- Operations
  - Metafile sent to the browser
  - The browser activates the proper player
  - The player activates a control connections and an RTSP data connection with the server

Metafile example

<title>Twister</title>
<session>
<group language=en lipsync>
<switch>
<track type=audio
e="PCMU/8000/1"
src =
"rtsp://audio.example.com/twister/audio.en/lofi">
<track type=audio
e="DVI4/16000/2" pt="90 DVI4/8000/1"
src =
"rtsp://audio.example.com/twister/audio.en/hifi">
</switch>
<track type="video/jpeg"
src =
"rtsp://video.example.com/twister/video">
</group>
</session>

RTSP operation



RTSP messages: example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231

S: 200 3 OK