RTP

Real-Time Transport Protocol
RFC 1889

What is RTP?

- 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?

What RTP is not

- RTP does not guarantee reliable, on-time delivery of the packets (the network is expected to do that).
- RTP does not address resource reservation (use RSVP or any other resource reservation protocol).
- RTP does not address QoS in any way (the underlying network needs to take care of that).

RTP Assumptions on the Network

- The network may drop or re-order the packets
  - RTP will detect this but has no mechanisms to recover
- The network must provide the following services:
  - Application multiplexing and demultiplexing (port numbers)
  - Payload framing
  - Payload size indication
  - Multicast delivery
  - Data integrity check (checksums)
- RTP typically runs on top of UDP, but the use of other protocols is not precluded.

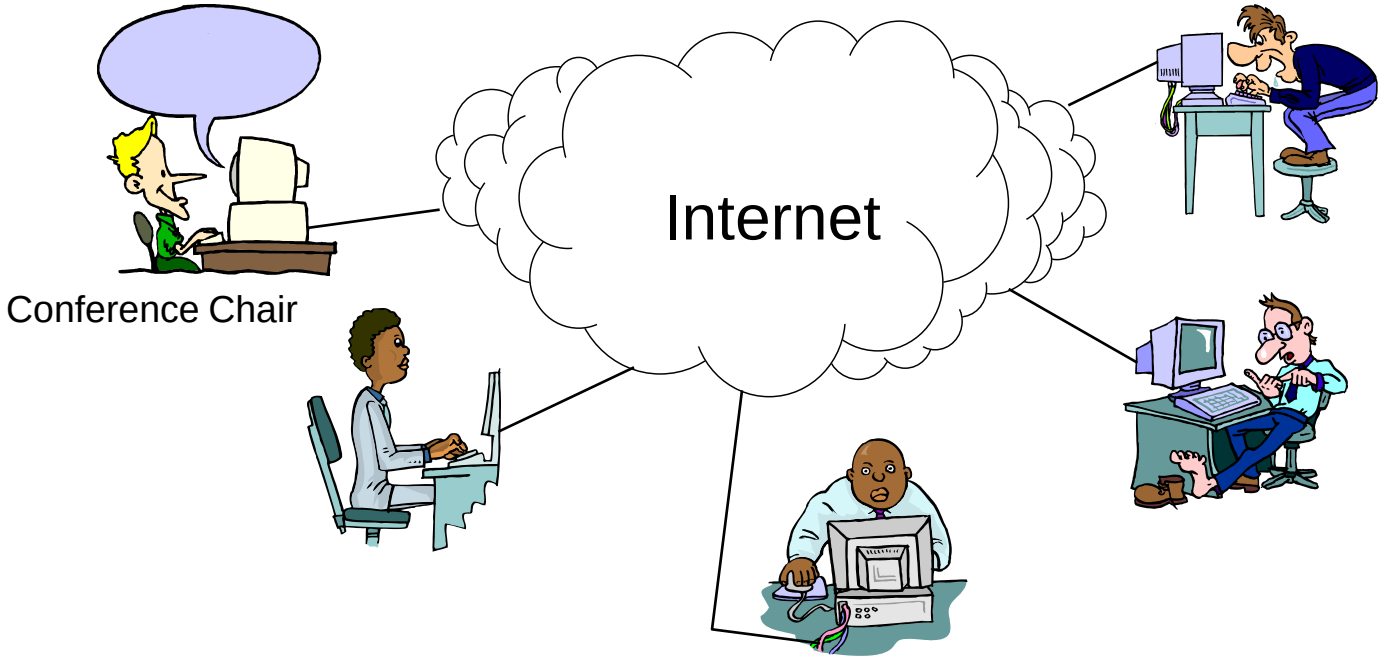
RTP Parts

- 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

What is missing on RTP

- RTP (as defined in RFC 1889) is a protocol framework and is deliberately incomplete
- RTP defines a basic set of functionality that is common to most multimedia applications.
- RTP defines the concept of a ***profile***, which completes the specification for a particular application:
  - Media encoding specifications
  - Payload format specifications

RTP Usage Example

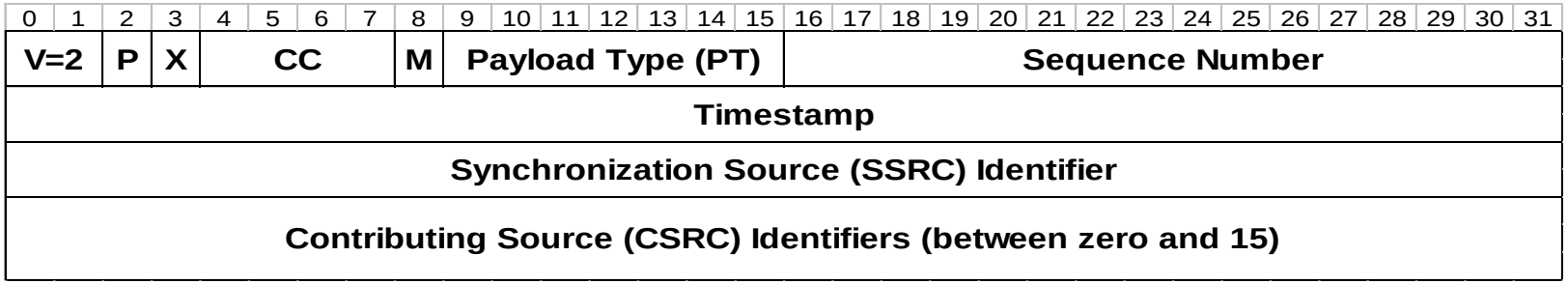


Example: Audio Conferencing over the Internet

Usage Example (cont.)

- The conference chair acquires a multicast address and two UDP ports, one for RTP, another for RTCP.
- Addressing information is distributed to all the participants.
- Conferencing application captures audio in 20 ms chunks. To this data, RTP/UDP/IP headers are prepended and the packet is sent.
- The RTP header indicates what kind of audio encoding is being used by each participant (PCM, ADPCM, etc.)
- Timing information in the RTP packet header allows each receiver to play the audio samples each 20 ms.
- Sequence numbers in the RTP header allow receivers to detect packet loss
- RTCP allows participants to be notified when someone joins or leaves the conference, and how well the participants are being received.

RTP Header



- **V**: Protocol version, currently equals 2.
- **P**: Padding. If this is set, the RTP packet contains padding bytes at the end. The last byte of the padding is a count of how many padding bytes were added. Used with encryption systems that require a fixed block length.
- **X**: Extension: If this is set, the RTP header is followed by exactly one extension header.

RTP Header (cont.)

- **CC**: CSRC count. Indicates the number of CSRC fields at the end of the header (between zero and 15).
- **M**: Marker bit. Defined by the profile. Intended to flag significant events in the payload, such as frame boundaries.
- **PT**: Payload Type. Identifies the kind of media and type of compression being carried by the stream. This is an enumerated type with actual values being defined by the IETF.
- **Sequence Number**: Incremented by one at each transmitted packet. Initial value chosen at random.
- **Timestamp**: Reflects the sampling time corresponding to the first byte in the RTP payload. Timestamp frequency is defined by the profile and initial value is random. It is legal to have multiple packets with the same timestamp.

RTP Header (cont.)

- **SSRC**: Synchronization Source identifier. This is a randomly generated identifier for this particular stream. It is intended to be independent of the network address. In a session, all SSRCs are supposed to be unique. A mechanism exists to guarantee that.
- **CSRC**: Contributing Source identifiers. List of original SSRCs that contributed to this stream. Example application: Audio-Conferencing mixer.
  - All participants send their streams to the mixer.
  - The mixer combines them into a single stream, which is sent back to the participants.
  - The stream from the mixer will have a list of SSRCs from each participant.

RTCP

- Composed of periodic transmissions to the participants of the session, on a different UDP port number, but using the same distribution mechanism.
- Functions:
  - Feedback on reception quality, to be used in:
    - Adaptive media encoding
    - Fault isolation
  - Distribution of a persistent transport level identifier (CNAME) for each session participant
    - Used to group multiple data streams from the same source, for example, audio and video.

RTCP Functions (cont.)

- Functions:
  - RTCP rate must be controlled to allow the protocol to scale
    - Every participant sends RTCP packets to all other participants
    - Participants can measure volume of RTCP traffic and throttle down as needed
    - Target: keep rate to 5% of the total session bandwidth
  - Convey minimum session control information (optional)
    - Useful in loosely-controlled sessions where participants can join and leave at any time.
    - Not a general session control mechanism

RTCP Packet Types

- RTCP defines several types of packets
- Multiple RTCP packets can be combined into a single UDP packet for efficiency
- Defined packet types:
  - SR: Sender Report. Contains sender statistics and reception statistics.
  - RR: Receive Report. Subset of the sender report containing only receive statistics.
  - SDES: Source Description. Contains information about the source (CNAME, e-mail, geographical location, username, tool information)
  - BYE: Goodbye packet. Indicates that a source is leaving the session.
  - APP: Application data. Used to extend the protocol with application-specific extensions.

RTCP Source Reports

RTCP Source Reports contain:

- Sender SSRC
- NTP and RTP timestamps corresponding to the time this report was transmitted.
- Cumulative count of RTP packets and bytes sent by this source since it started sending this stream.
- Zero or more reception reports containing:
  - SSRC of the stream being reported on.
  - Fraction of the packets lost from this source since last report
  - Cumulative number of packets lost from this source
  - Highest sequence number received
  - Measured interarrival jitter, in timestamp units
  - Last SR timestamp received from this source (LSR)
  - Delay between the reception of the last SR and the transmission of this SR

Receiver Reports(RR)

- Receiver Reports (RR) are a subset of the Source Reports
- They only contain receive blocks (no sender information)
- RTP profiles can define extensions to SR and RR messages

Analyzing 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

Security Considerations

- Application-level encryption of full packet recommended
  - RTP / RTCP header contains padding bit to recover original packet length
  - random start values of sequence number prevent known plaintext attacks
  - decryption verified by sanity checking fields in header
  - algorithm choice left to application; default is DES in CBC mode
  - key distribution must be accomplished by external means
  - network layer encryption can be used where it is available
- No explicit RTP support at this time for authentication
  - authentication could be accomplished via encapsulation
- RTP can also be used in the clear carrying encrypted payloads
  - good for hardware which both decrypts and processes payload data
- Preferred way of encrypting: use IPSec instead.