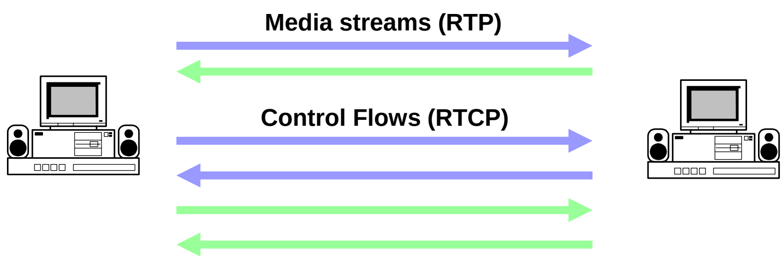
Real-Time Transport Protocol (RTP)

Real-time Transport Protocol (1)

- RTP Functionality (RFC 3550)
  - framing for audio/video information streams
  - preserve intra- and inter-stream timing
  - mechanisms for awareness of others in a conference
  - ➲ RTP sessions



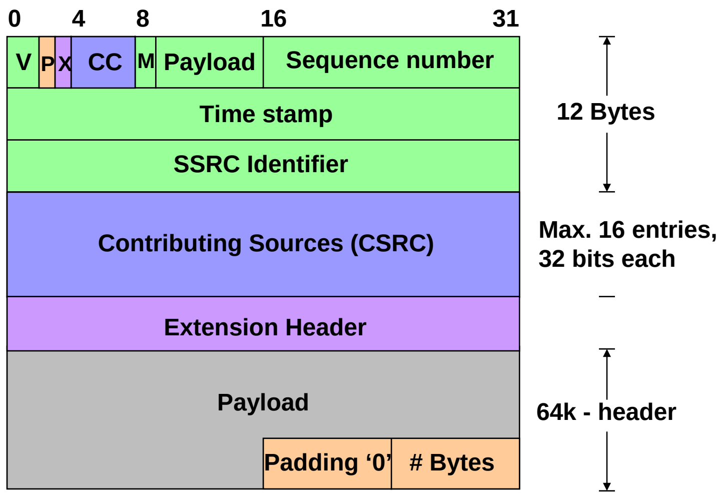
Real-time Transport Protocol (2)

- Standard RTP packet header
  - Independent of *payload type*
  - Possibly seconded by *payload header*

- Mechanisms
  - Detect packet loss, cope with reordering
    - sequence number per media stream
  - Determine variations in transmission delays
    - media specific time stamp (e.g., 8 kHz for PCM audio)
    - allows receiver to adapt playout point for continuous replay
  - Source identification
    - possibly mixed from several sources
  - Payload type identifier

HELSINKI UNIVERSITY OF TECHNOLOGY
NETWORKING LABORATORY

RTP Header



RTP Header Fields (1)

V: Version — version 2 defined in RFC 1889

P: Padding — indicates padding
bytes indicated in last byte

X: eXtension bit — extension header is present

Extension header — single additional header (TLV coded)

CC: CSRC count — # of contributing sources

CSRC: contributing sources — which sources have been "mixed" to produce this packet's contents

RTP Header Fields (2)

M: Marker bit — marks semantical boundaries in media stream (e.g. talk spurt)

Payload type — indicates packet content type

Sequence # — of the packet in the media stream (strictly monotonically increasing)

Timestamp — indicates the instant when the packet contents was sampled (measured to media-specific clock)

SSRC: synchronization source — identification of packet originator

Real-time Transport Control Protocol

Mechanisms:

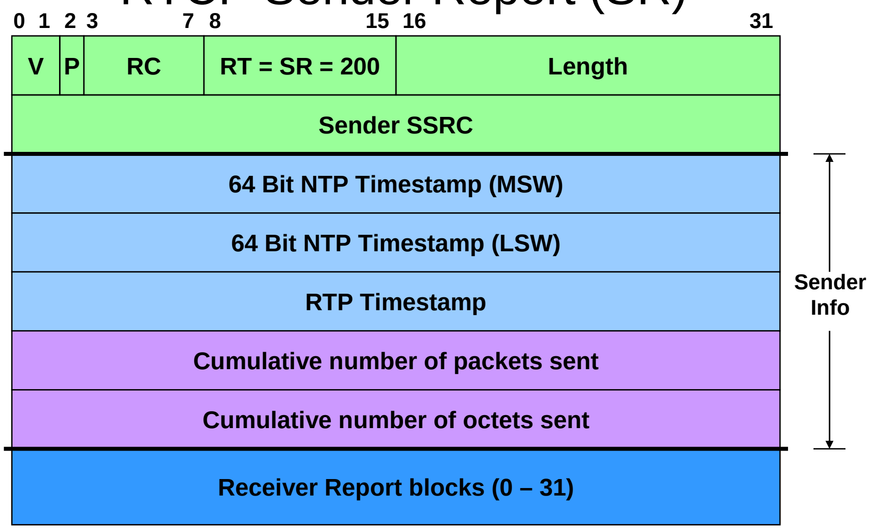
- Receivers constantly measure transmission quality
  - delay, jitter, packet loss
- Regular control information exchange between senders and receivers
  - feedback to sender (receiver report)
  - feed forward to recipients (sender report)
- Allows applications to adapt to current QoS
- Overhead limited to a small fraction (default: 5% max.) of total bandwidth per RTP session
  - members estimate number of participants
  - adapt their own transmission rate

Obtaining sufficient capacity: outside of RTP

RTCP Sender Report

- Enable cross-media stream synchronization
  - Relate stream-specific RTP time stamp to wall clock time
  - NTP timestamp + RTP timestamp
  - Playout adjustment to be performed by the receivers
- Provide data point for RTT measurement
  - NTP timestamp
- Provide feed forward about data transmitted
  - Transmit sender's packet and byte count
  - Enable receiver to do proper loss calculation
- Include Receiver Reports for the sender as well

RTCP Sender Report (SR)



RTCP Receiver Report

- Feedback timing for RTT estimation
  - SR Timestamp
    - Middle 32 bits taken from the last SR's NTP timestamp
  - Delay since last SR
    - Local delay at receiver between receiver SR and sending the RR block
    - Measured in units of 1 / 65556 seconds
- Provide per-sender reception statistics
  - Total number of packets lost
  - Fraction of packets lost (in units of 1 / 256)
  - Highest sequence number received so far
  - Jitter of received packets
- Enable adaptive sender behavior
  - Adjust codecs, codec parameters, transmission rate, etc.

RTCP Receiver Report (RR)

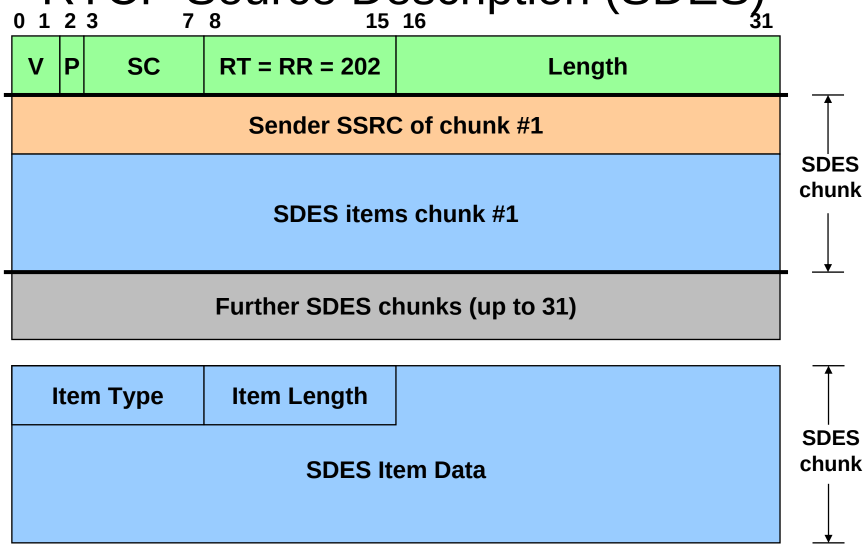
0 1	2	3 … 7	8 … 15	16 … 31
V	P	RC	RT = RR = 201	Length
Sender SSRC				
SSRC of sender reported in this block				
Fraction lost			Cumulative number of packets lost	
Extended highest sequence number received				
Interarrival jitter				
Last SR				
Delay since last SR				
Further receiver report blocks (up to 31 in total)				

Receiver Report block

RTCP Source Description (SDES)

- Persistent Identification of an endpoint: Canonical Name
  - CNAME — globally unique identifier (id@host)
  - *Mandatory!*
  - Binding across RTP sessions
  - Identification across changes in the SSRC in an RTP session
- Providing additional information about an endpoint
  - NAME — Name of user (or system)
  - EMAIL — mailto: address
  - PHONE — phone number
  - LOC — location (no format defined)
  - TOOL — (software) client in use
  - NOTE — brief to other participants (e.g. “on the phone”)
  - PRIV — private extensions

RTCP Source Description (SDES)

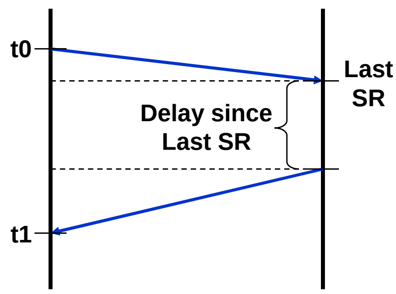


RTCP Statistics Collection (Sender)

- Round-Trip Time (sender only)
  - Derived from time stamps in RR
  - Simple formula:

    RTT = t1 – t0 – DSL_SR

  - RTT may be asymmetric!



- Byte count
- Packet count

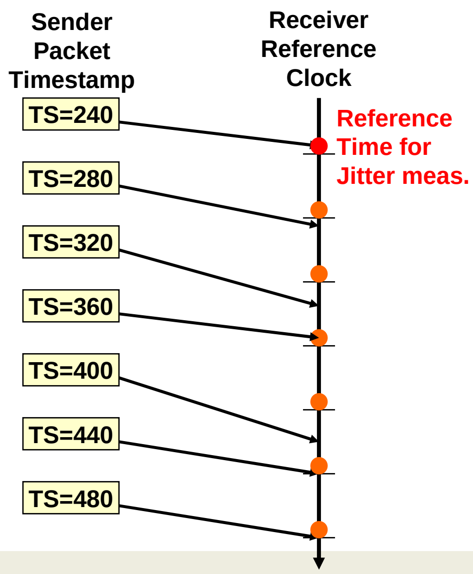
RTCP Statistics Collection (Receiver)

- Packet Loss
  - Calculated from gaps in sequence number space
    - First (lowest packet sequence number) received
  - Expected number of packets = current – lowest
  - Received number of packets
    - Count duplicates, out-of-order, and late packets as received!
  - Absolute # of lost packets = expected – received
    - May be negative!
  - Fraction of lost packet
    - Loss since last SR or RR packet was sent
  - Loss of all packets not detected!
- Extended highest sequence number received (32 bits)
- Time of last SR reception
- Jitter

RTCP Interarrival Jitter Estimation

- Receiver measures in time units of the media clock
- Relates it to local real-time clock
- Initialized through first packet received
- Derives expected reception time
- Calculates deviation D upon packet reception
- Sampled for each packet
- Jitter derived for each peer of successively received packets
  - Ordering is not relevant
- Weighing function:

  J = J' + (D – J') / 16



Other RTCP Packets

- BYE
  - Announce that an entity will be leaving a session
  - Provide a reason phrase
- APP
  - Application-specific extensions
- Further extensions being defined
  - Timely feedback extensions
  - Single source multicast extensions
  - More detailed reception statistics

RTP / RTCP Bandwidth Calculation

- Senders and receivers estimate group size (independently!)
  - # senders from SRs
  - # receivers from RRs
  - Consider BYE packets
- RTCP bandwidth: 5% of RTP session bandwidth
  - 5% for both senders and receivers   if >25% senders
  - 1.25% for senders and 3.75% for receivers   otherwise
- Sample of RTCP packets sent and received
  - Average number of bytes per time unit currently transmitted
  - Consider whether you are sender or receiver
- Last time own packet was sent
  - Calculate expected next time to send based upon above data rate
  - Use a minimum of 5 seconds (initially, 2.5 seconds)
  - Dither with random function: 0.5 .. 1.5 of calculated interval
  - Timer reconsideration: double check values when timer expires

RTCP Transmission Interval

- Must scale with the number of group members
  - Must not take up too much network capacity (rate-limited!)
- Overall "RTP session bandwidth"
  - Includes UDP and IP header overhead
  - Provided by the application (i.e. not measured dynamically)
- Default: 5% of the session bandwidth for RTCP
  - Takes role (sender or receiver) into account
  - Up to 25% of session members are senders
    - 3.75% for receivers, 1.25% for senders
  - More than 25% of session members are senders
    - Share data rate proportionally
- May be modified by profiles
  - Parameters S and R to indicate relative share for senders/receivers
- Scalable RTCP transmission interval
  - Based upon the group size, RTCP data rate, average RTCP packet size

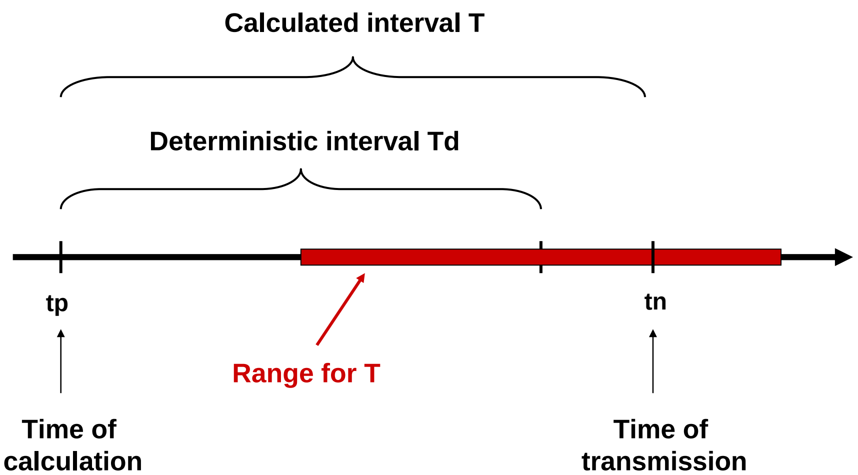
RTCP Variables for Bandwidth Calculation

- Data rate
  - Session bandwidth
  - R, S: Receiver, sender bandwidth share
  - Average RTCP packet size (moving average)
- Time
  - Tp — last time an RTCP packet was sent
  - tc — current time
  - tn — next scheduled transmission of an RTCP packet
- Membership
  - pmembers — # members when tn was last computed
  - members — current # members
  - senders — # senders in the session
  - n — relevant # of members (depending on role, etc.)
- Intervals
  - Td — Deterministic calculated interval
  - T — Calculated interval
  - Tmin — minimal interval between RTCP packets

Basic Operation

- Determine role (sender or receiver)
  - Derive n as # of relevant members for calculation
  - Derive relevant bandwidth share
- C = average RTCP size / relevant bandwidth share
- Td = max (Tmin, n*C)
- T = Random [0.5 – 1.5] * Td

Basic RTCP Interval Calculation



Timer Reconsideration

- The group size may change between tp and tn
- Particularly during startup and shutdown phase
  - Many users may join / leave during a short period of time
- Many joining parties: risk of RTCP implosion
- Algorithm for joining members
  - Validate the group size at time tn before transmission
  - Recalculate T as above
  - If tp + T <= tc   transmit RTCP packet and update variables
  - If tp + T > tc   set tn = tp + T and set timer to expire at tn
- Algorithm for leaving members
  - Adjust tp, tn according to the observed membership change
    - Factor: members / pmembers
  - Run every time a member leaves or times out

Extended Operation

- Determine role (sender or receiver)
  - Derive n as # of relevant members for calculation
  - Derive relevant bandwidth share
- C = average RTCP size / relevant bandwidth share
- Td = max (Tmin, n*C)
- T = Random [0.5 – 1.5] * Td
- T = T / e^-1.5   (T = T / 1.21828)
  - Correction factor for timer reconsideration

RTP Payloads

RTP Payload Types

- 7-bit payload type identifier
  - Some numbers statically assigned
  - Dynamic payload types identifiers for extensions – mapping to be defined outside of RTP (control protocol, e.g. SDP “a=rtpmap:”)

<u>Payload formats defined for many audio/video encodings</u>

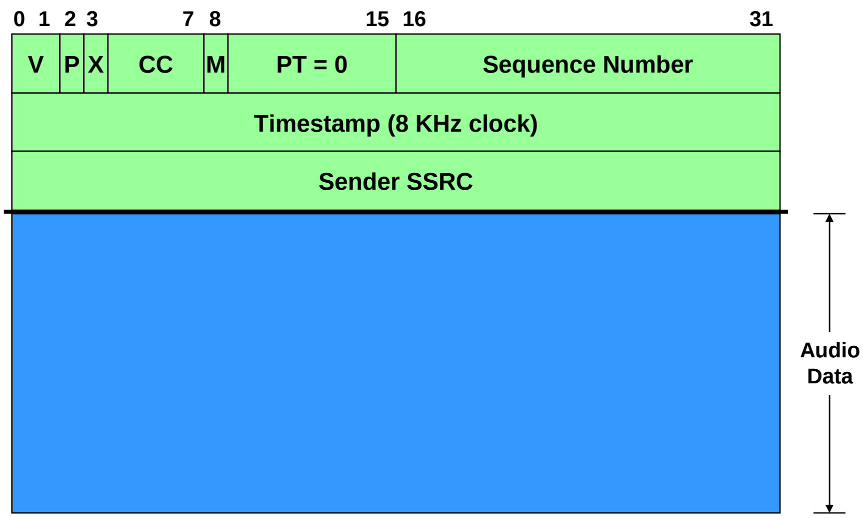
- Conferencing profile document RFC 3551
  - Audio: G.711, G.722, G.723.1, G.728, GSM, CD, DVI, …
- In codec-specific RFCs
  - Audio: Redundant Audio, MP-3, ...
  - Video: JPEG, H.261, MPEG-1, MPEG-2, H.263, H.263+, BT.656
  - Others: DTMF, text, SONET, ...
- Generic formats
  - Generic FEC, (multiplexing)

Media Packetization Schemes (1)

General principle:

- Payload specific additional header (if needed)
- Followed by media data
  - Packetized and formatted in a well-defined way
  - Trivial ones specified in RFC 3551
  - RFC 2029, 2032, 2035, 2038, 2190, 2198, 2250, 2343, 2429, 2431, RFC 2435, 2658, 2733, 2793, 2833, 2862, and many further ones
  - Guidelines for writing packet formats: RFC 2736
- Functionality
  - Enable transmission across a packet network
  - Allow for semantics-based fragmentation
  - Provide additional information to simplify processing and decoding at the recipient
  - Maximize possibility of independent decoding of individual packets

Audio over RTP: PCM



Video over RTP: H.261

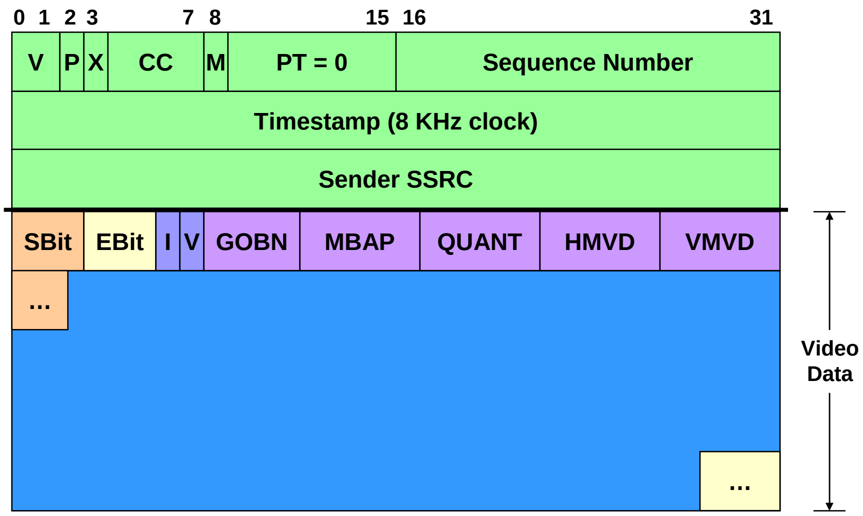
Additional payload-specific header preceeds payload

- To avoid expensive bit shifting operations
  - Indicate # invalid bits in first (SBit) and last (EBit) octet of payload
- Indicate Intra encoding (I bit)
- Indicate the presence of motion vector data (V bit)
- Carry further H.261 header information to enable decoding in the presence of packet losses

Further mechanisms for video conferencing

- FIR: Full Intra Request
  - Ask sender to send a full intra encoded picture
- NACK: Negative Acknowledgement
  - Indicate specific packet loss to sender

Video over RTP: H.261 (2)



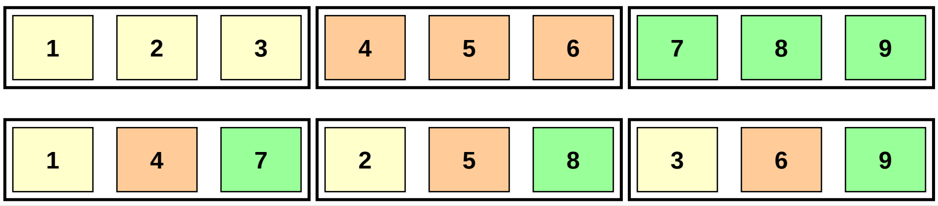
Media Packetization Schemes (2)

Error-resilience for real-time media

- Input: Observation on packet loss characteristics
- Generic mechanisms (RFC 2354)
  - Retransmissions
    - in special cases only (e.g. with no interactivity!)
  - Interleaving
  - Forward Error Correction (FEC)
    - media-dependent vs. media-independent
    - Generic FEC: RFC 2733
- Feedback loops for senders
  - based upon generic and specific RTCP messages
  - adapt transmission rate, coding scheme, error control, ...

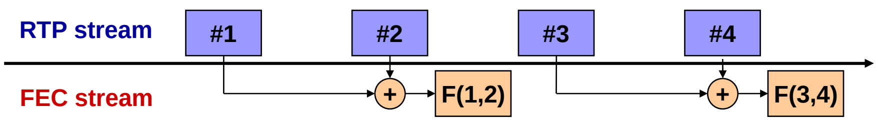
RTP Interleaving

- Distribute packets or packet contents for transmission
  - Avoid consecutive packet erasures in case of (burst) losses
  - Avoid loss of large consecutive data portions in case of single packet losses
- Motivations
  - Human perception tolerates individual losses better
  - Make simple FEC schemes work better with burst losses (e.g. XOR)
- Drawback
  - Re-ordering causes additional delay at the receiver

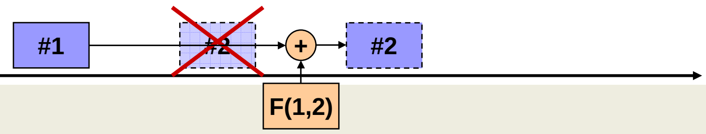


RTP FEC (RFC 2733)

- Forward Error Correction scheme for RTP packets
  - Media-independent, flexible FEC (that can be enhanced)
- Simple XOR-based (parity) FEC
  - P_fec = P1 XOR P2 XOR P3 XOR … XOR Pn
    - Allows reconstruction of any single missing packets of P1, …, Pn, P_fec
- RTP FEC stream transmitted independently of RTP stream
  - Separate transport address (IP address, port number)
  - Different SSRC

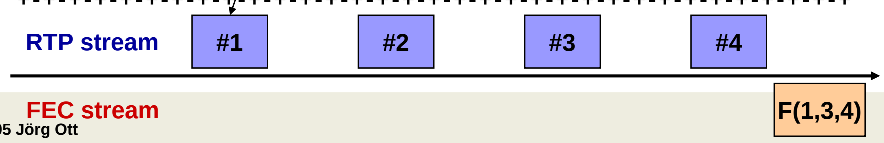


- Recovery



RTP Parity FEC Packet format

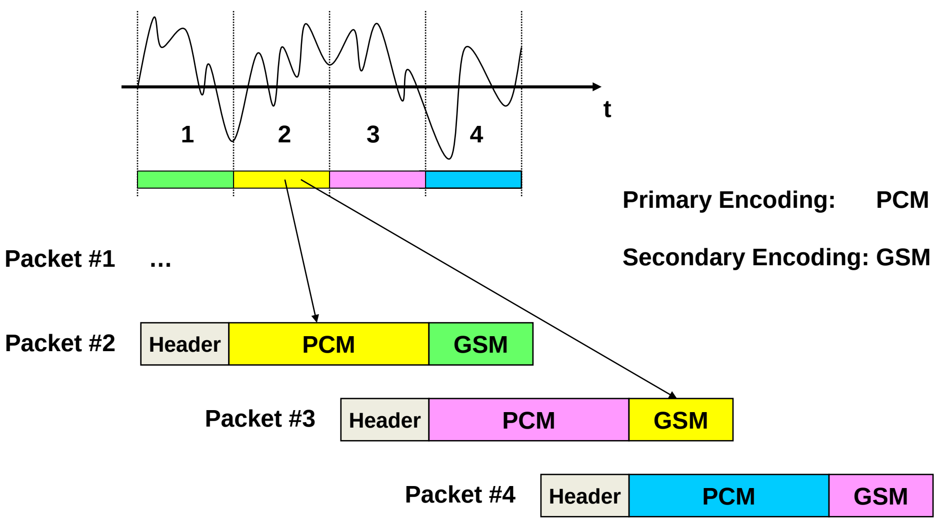
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT   |       PT      |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            SN base      = 1   |        length recovery        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| PT recovery |       mask  = 10110000 00000000 00000000      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         TS recovery                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:        XOR of Payloads indicated by SN Base and mask          :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



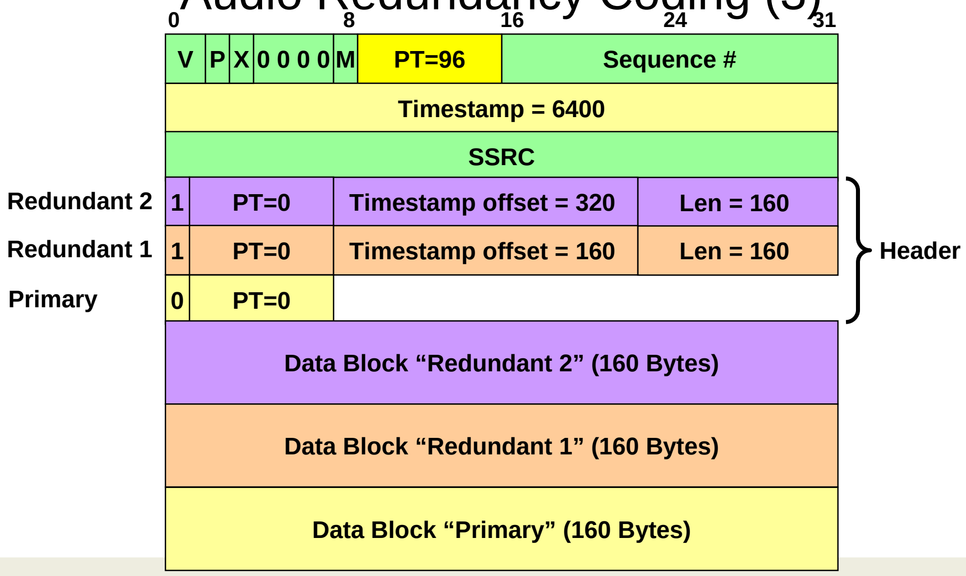
Audio Redundancy Coding (1)

- Audio Packets are small!
  - have to be because of interactivity
    - avoid large packetization delay
  - packet loss primarily depends on packet rate
    - rather than packet size
- Payloads for multiple time slots in one packet
  - send redundant information in packet n
    to reconstruct packets k, ..., n-1 in packet n
  - redundant information typically sent at lower quality
  - details defined in RFC 2198
  - uses dynamic payload type
- Format specification, e.g. using SDP
  - m=audio 20002 RTP/AVP 96 0 0 0
  - a =rtpmap:96 red/8000/1

Audio Redundancy Coding (2)



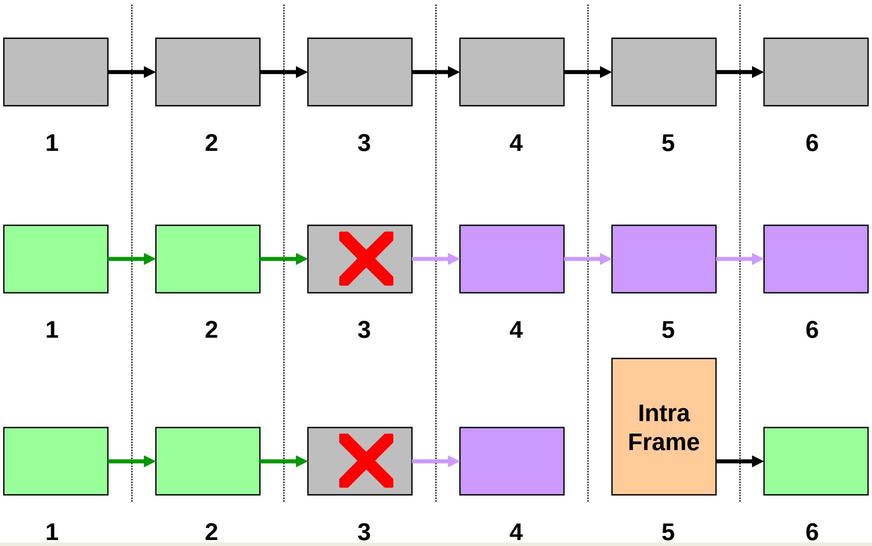
Audio Redundancy Coding (3)



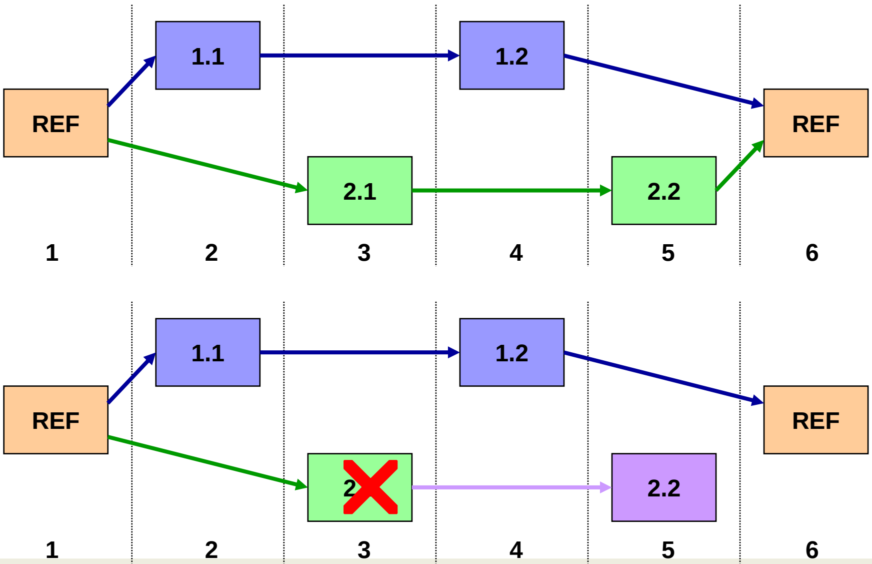
Video Redundancy Coding (1)

- Video redundancy coding
  - For H.263+ video streams
  - Transmit several interleaved sequences of predicted frames (threads) instead of one
    - improves error resilience against packet loss
- Principle
  - create several (n) independently decodable streams
  - achieved by choosing different reference pictures
  - decode only streams with no packet losses
    - reduces temporal resolution by 1/n-th per affected stream
  - bit rate penalty due to larger deltas between frames
  - RFC 2429, revised version in progress

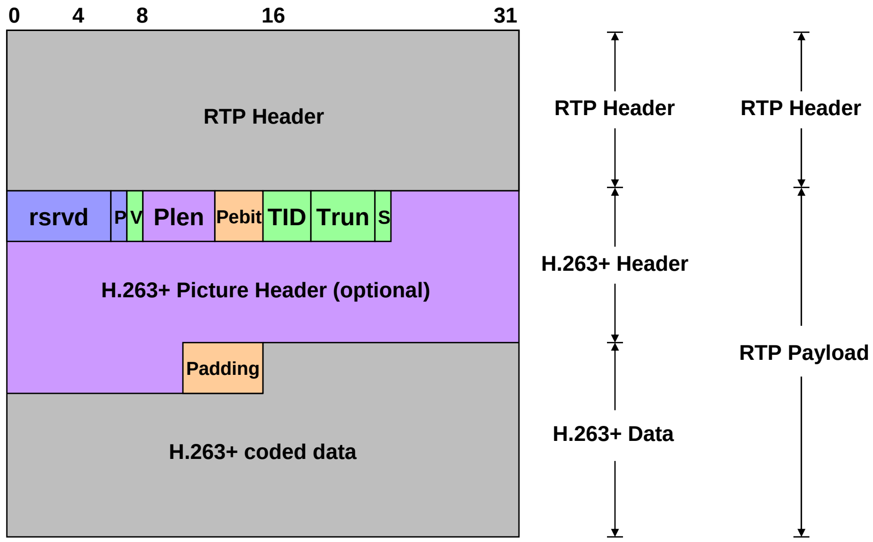
Video Redundancy Coding (2)



Video Redundancy Coding (3)



Video Redundancy Coding (4)



DTMF over RTP (1)

- DTMF digits, telephony tones, and telephony signals
  - two payload formats
  - 8 kHz clock by default
  - audio redundancy coding for reliability
- Format 1: reference pre-defined events
  - 0 - 9 * # A - D (Hook)Flash [17]
  - modem and fax tones [18]
  - telephony signals and line events [43]
    - dial tones, busy, ringing, congestion, on/off hook, …
  - trunk events [44]
  - specified through identifier (8-bit value), volume, duration
- Format 2: specify tones by frequency
  - one, two, or three frequencies
  - addition, modulation
  - on/off periods, duration
  - specified through modulation, n x frequency, volume

DTMF over RTP (2)

Packet Format 1: Events

0		8	16	24	31
RTP Header					
Event	E	0	Volume	Duration	

Packet Format 2: Tones

0	8	16	24	31
RTP Header				
Modulation	T	Volume	Duration	
0 0 0 0	Frequency	0 0 0 0	Frequency	
. . .				
0 0 0 0	Frequency	0 0 0 0	Frequency	

RTCP Payload Type Overview (1)

- RFC 3551 Collection of simple packetization formats (formerly RFC 1890)
- RFC 2029 Sun CellB Video encoding
- RFC 2032 H.261 video
- RFC 2435 JPEG video (was RFC 2035)
- RFC 2250 MPEG-1/MPEG-2 video (was RFC 2038)
- RFC 2190 H.263 video (historic)
- RFC 2343 Bundled MPEG
- RFC 2429 H.263+ video & video redundancy support
- RFC 2431 BT.656 video
- RFC 2658 PureVoice audio
- RFC 2793,4103 Text conversation
- RFC 2833 DTMF, telephony tones, and telephony signals
- RFC 2862 Real-time Pointers
- RFC 3016 MPEG-4 Audio/visual streams
- RFC 3047 G.722.1 audio
- RFC 3119 Loss-tolerant format for MP3

RTCP Payload Type Overview (2)

- RFC 3189 DV video
- RFC 3190 12-bit DAT and 20-/24-bit linear audio
- RFC 3267 Adaptive Multirate (AMR) audio
- RFC 3389 Comfort noise
- RFC 3497 SMPTE 292M video
- RFC 3557 ETSI Distributed speech recognition (ES 201 108)
- RFC 3558 Enhanced variable rate codecs and selectable mode vocoders
- RFC 3640 MPEG-4 elementary streams
- RFC 3952 Low Bit Rate Codec (iLBC) Speech
- RFC 3984 H.264 Video
- RFC 4040 64 kbit/s Transparent Call
- RFC 4060 Distributed speech recognition encoding (ES 202 050/211/212)
- RFC 4175 Uncompressed Video
- RFC 4184 AC-3 Audio

Many more to come…

- Revisions to many of the above, no-op payload for connectivity detection, further codecs…

RTP Extensions

- Timely feedback from receivers to senders
- Support for Source-specific Multicast (SSM)
- Extended RTCP Report (detailed statistics)

RTCP Feedback Issues

- Senders provide regular information about media stream
  - Seems ok
- Receivers transmit RTCP at somewhat regular intervals
- RTCP RRs provide long-term statistics on reception quality
- Senders can adapt transmission strategy to receiver observations
  - Different codecs, data rate, etc.
- BUT: No short-term feedback possible
  - Error repair or mitigation impossible
- Problem: Value of receiver feedback descreases over time
  - Repair more expensive at later times
  - Artefacts become noticeable to the user

Approach: RTCP-based Feedback

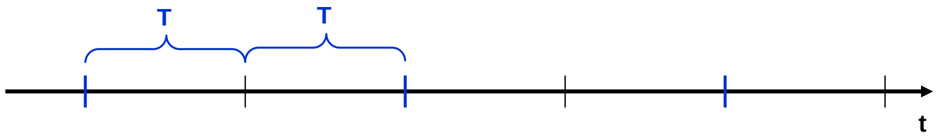
- New Profile for RTP: AVPF

Idea:

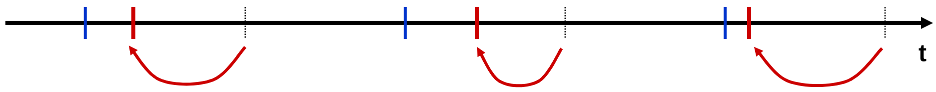
- Packet losses are usually rare
- Provide statistical chance of virtually immediate feedback from receiver(s) to sender
- Keep the basic RTCP properties
- Eliminate Tmin
- Work most efficiently with unicast
- Also scale to moderate group sizes

Overview

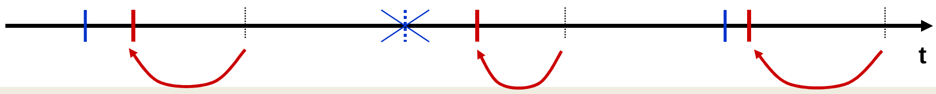
Regular RTCP operation (depicted w/o randomization, i.e. T = Td)



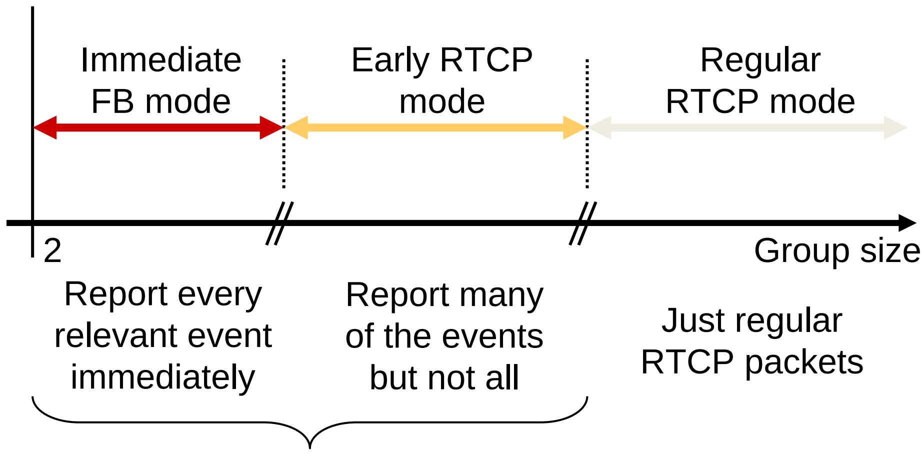
Allow (at most every other) RTCP packet to be sent earlier



Allow to reduce the number of regular RTCP packets (w/o affecting RTCP rate)

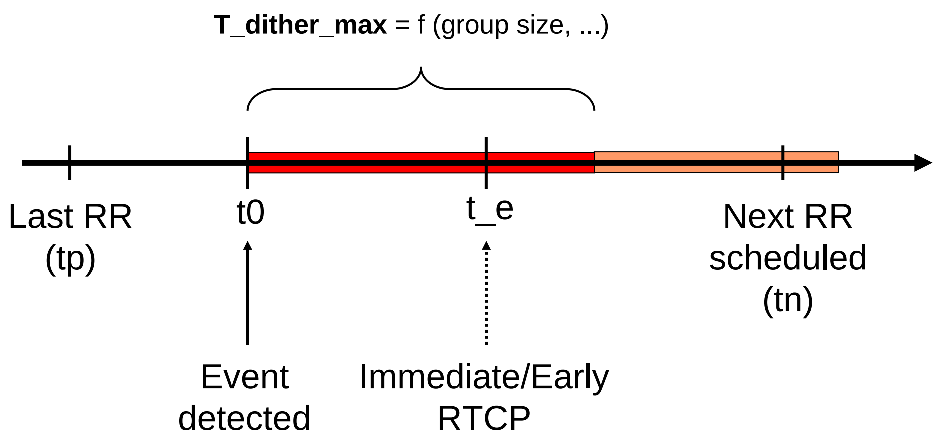


Modes of Operation



Send feedback + regular RTCP packets

RTCP Feedback Timing



Delay calculation

$$\text{T\_dither\_max} = \begin{cases} 0 & \text{if grp size} = 2 \\ l * T & \text{otherwise} \end{cases}$$

Simulated guess: **l = 0.5**

Better approach: use RTT measurements!
But those are only available to senders…
Mixed operation (using Td and RTT) will not work.

RTCP Types of Feedback

- ACK Mode
  - Positive acknowledgements for received packets
  - Restricted to point-to-point operation
- NACK Mode
  - Negative acknowledgments e.g. for missing packets or other events
  - Scalable with suppression technique
- Other types of feedback conceivable
- Transport layer feedback packets (Generic NACK)
  - Identifies missing or received packets
- Payload-specific feedback packets
  - Specific to certain codecs (e.g. video)
  - Picture / frame loss indication, reference picture selection
- Application feedback packets

RTCP Feedback Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT   |       PT      |          length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            Feedback Control Information (FCI)                 :
:                                                               :
```

```
Example: Generic NACK Packet

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            PID                |             BLP               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RTCP Feedback Packet Format (2)

Example: Slice lost indication

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            First        |   Number                | PictureID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example: Reference Picture Selection

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PB       |0| Payload Type|  Native RPSI bit string       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   defined per codec          ...                |  Padding (0)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


Example for Statistical Feedback

- Applicability of feedback depends on many parameters
  - Group size, RTP & RTCP bandwidth, application requirements

256 kbit/s video stream, 30 frames per second, 1500 bytes MTU
Single sender, > 3 receivers (i.e. 3.75% RTP bandwidth for receivers)
H.263+ with approximately 1 packet per frame
5% packet loss, equally distributed, receiver independence

Statistically yields 3 losses every two seconds per receiver
3.75% * 256 kbit/s = 9.6 kbit/s for all receivers
Assuming 120 bytes (= 960 bits) per RTCP packet: 10 packets / s

If every receiver reports every loss event: 6 – 7 receivers on average

If reporting every other loss event is sufficient: ~14 receivers

Increases further if losses are correlated in some fashion

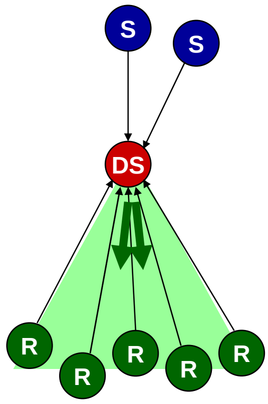
RTP Retransmissions

- Explicit repair mechanism for RTP streams
- Works for applications with acceptable higher latency
  - E.g. media streaming
- Applicable to point-to-point and small group scenarios
- Used with RTCP feedback extensions
- Approach
  - Original RTP stream
  - Augmented by retransmission RTP stream
  - Mapped to different RTP sessions or sender SSRCs
    - Use always different sessions for multicasting
  - Keeps the retransmission scheme backward compatible
  - Does not confuse RTCP statistics
  - Works with all payload types
  - Allows for multiple payload types in a session

RTCP Retransmission Packet Format

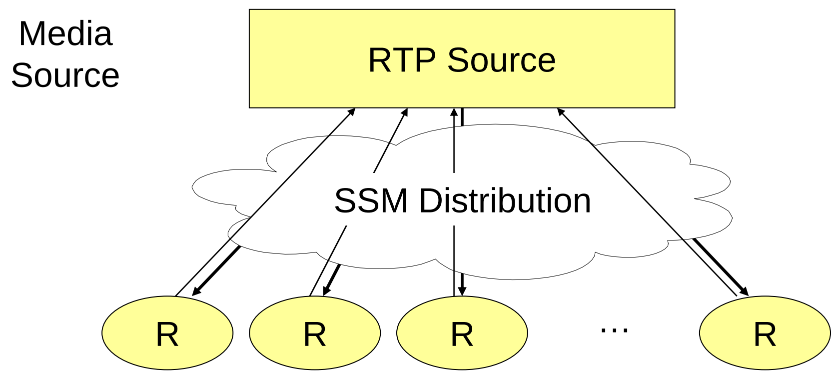
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT   |       PT      |          length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            OSN                |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                  Original RTP Packet Payload                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RTCP for SSM

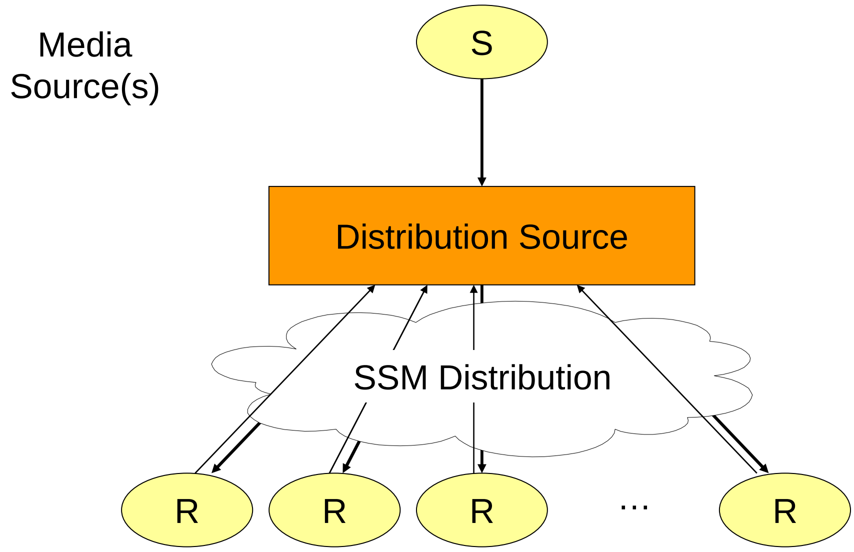


- Multicast connectivity unidirectional
  - From Distribution Source to receivers
  - Opposite direction needs to use unicasting
    - May follow different network path
- Result: no direct communication between receivers
- Adaptations required to make RTCP work
  - Estimate group size
  - Adjust timing of RTCP transmission (adhere to bandwidth limit)
  - Resolve SSRC collisions
- Two basic modes of operation
  - Make distribution source reflect RTCP traffic back to receiver
  - Provide summaries of relevant information along with sender reports

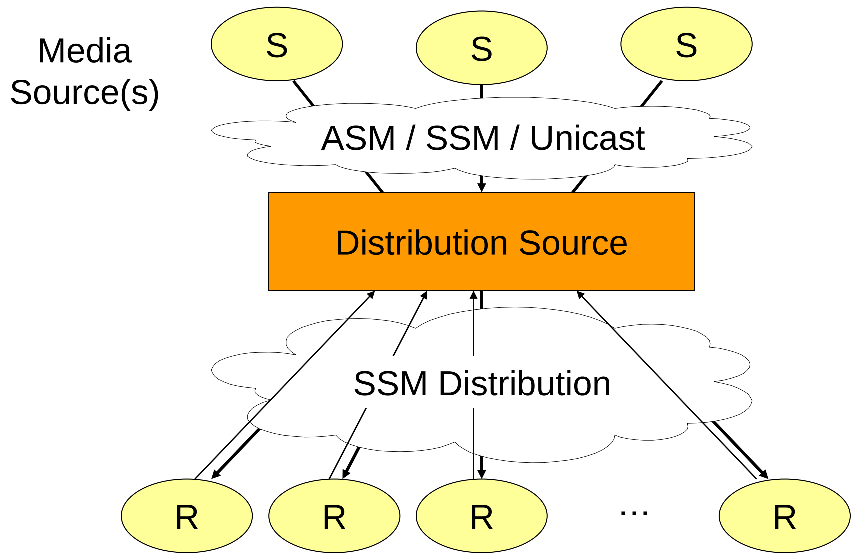
RTCP SSM Big Picture



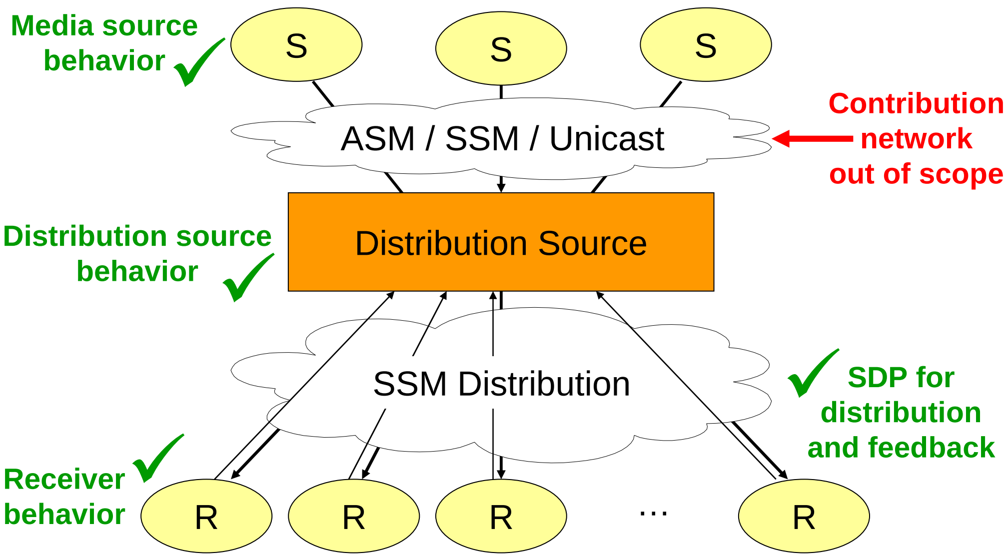
RTCP SSM Big Picture



RTCP SSM Big Picture



Scope of RTCP over SSM Spec



Simple Feedback Model

- Distribution source reflects packets back to receivers
  - Simple mirroring at the transport / application layer
- Uses the bandwidth share for receivers for distribution
  - Not an issue: non-overlapping paths
- Increases delay for inter-receiver communication
  - Particularly with asymmetric networks
  - May impact e.g. feedback suppression
- Required for all RTCP packets that cannot be summarized
  - Unknown extensions
  - Packets that require knowledge of the originator
- Particularly applies to RTCP APP packets

Feedback Summary Model

- Distribution source collects information from receivers
- Aggregates the information over time
- Distributes representative summaries back to receivers
  - In somewhat regular intervals
  - Saves bandwidth compared to simple reflection
  - Uses (part of) receiver rate in addition to sender rate
  - Acts as another receiver from an RTP/RTCP perspective (own SSRC)
- New RTCP packet: Receiver Summary Information (RSI)
  - Contains distributions for RTCP receiver statistics
    - Relative loss, cumulative loss, RTT, jitter
    - Allows receivers to relate themselves to group reception quality
  - Simple form: general statistics report on loss and jitter
  - Feedback target address
    - Where to unicast feedback packets to
  - SSRC collision reports
  - RTCP bandwidth indication

HELSINKI UNIVERSITY OF TECHNOLOGY
NETWORKING LABORATORY

RTCP RSI Packet Format

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2| reserved  |   PT=RSI=208  |             length            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           SSRC/CSRC                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +                           Timestamp                           +
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 :                     optional report blocks                    :
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Report Block:

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      RBT      |    Length     |                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      RBT-specific data        +
 |                                                               |
 :                                                               :
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Detailed Statistics Sub-Report Blocks

```
    0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 |     SRBT      |    Length     |        NDB            |   MF  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   Minimum Distribution Value                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   Maximum Distribution Value                  |
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 |                      Distribution Buckets                     |
 |                             ...                               |
 |                             ...                               |
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

- Used for
  - Loss, Jitter, RTT, Cumulative Loss
- Reflects information collected from RTCP RRs

Other Report Blocks

- Feedback target address
  - In-band signaling for distribution source address
  - Security!
- SSRC Collision
  - Initiate selection of new SSRCs
- General statistics
  - Average loss, average jitter, highest cumulative loss
  - Calculated from received RTCP RRs
- RTCP Bandwidth indication
- Group size and average RTCP packet size

Pace RTCP RRs

Further RTP Extensions in Progress

- RTP over TCP
  - Delineate RTP packets in a TCP connection
  - Linked to setup/teardown of TCP connections for media
- RTP over DCCP
  - Make use of congestion control characteristics of underlying transport
- TCP-friendly RTP profile
  - Adaptive transmission behavior compliant to the TCP-friendly rate control
  - Based upon Padhye equation for TCP throughput (RFC 3448)
- Secure Profile for AVPF
  - AVPF with SRTP

RTCP Report Extensions

- Defines new eXtended Report (XR) packet
- Supports RTT measurements for receivers
- Purpose
  - Provide more detailed information
  - Allow precise monitoring of network characteristics
    - Also by third parties
- Incorporate many statistics in RTCP packets
  - Exact packet receipt times
  - Lost and duplicate packets
  - Receiver reference time
    - for RTT measurements
  - VoIP metrics
    - Burst, delay, ...

RTP Specs (Summary)

- RFC 3550 Base specification (formerly RFC 1889)
- RFC 3551 RTP Profile for Audio and Video Conference with minimal control (was RFC 1890)
- RFC 2198 Redundant (Audio) coding
- RFC 2508 RTP header compression for low-speed links
- RFC 2733 Generic FEC
- RFC 2736 Guidelines for writers of RTP payload specifications
- RFC 2762 Group membership sampling (“timer reconsideration”)
- RFC 3095 Robust header compression for RTP (among others)
- RFC 3096 Requirements for robust IP/UDP/RTP header compression
- RFC 3158 RTP testing strategies
- RFC 3242 Link-layer assisted profile for IP/UDP/RTP header compression
- RFC 3243 Requirements & assumptions for 0-byte IP/UDP/RTP header compression
- RFC 3409 Lower-layer guidelines for robust IP/UDP/RTP header Compression
- RFC 3545 Enhanced compressed RTP (CRTP) for high-delay links
- RFC 3555 MIME registrations of RTP payloads
- RFC 3611 RTCP XR extension
- RFC 3711 Secure RTP (SRTP)