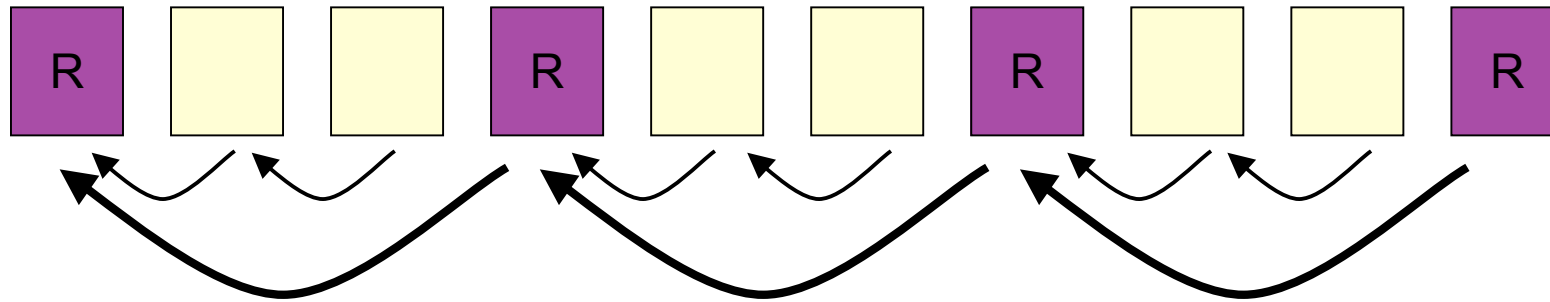# Marking and Selectively Retransmitting High-Priority Packets

Jonathan Lennox

Layered Media

"Note Well":

- Layered Media has potential IPR on this proposal.
- If it's accepted as a standards-track document, we'll license on the basis of reciprocity.
- See https://datatracker.ietf.org/public/ipr_detail_show.cgi?&ipr_id=726

Motivation

- It's possible to construct a packet stream so only a subset of the packets are needed for decoder state.
  - E.g. only need I frames and long-term reference frames.
- If an intermediate packet (yellow) is lost, a decoder can recover.
- The subset packets ("R Packets", purple) are far enough apart (>1 RTT) to be recoverable.
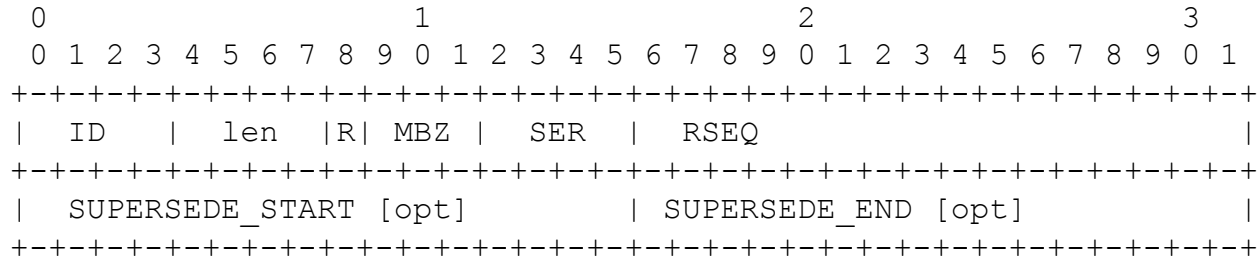- Want to detect and recover missing R packets as soon as possible.

Requirements

- Mark a subset of the packets as R packets.
- Detect a missing R packet as soon as any stream packet is received.
- Receiver can know whether any R packets are lost (and if not, take no action) no matter how many non-R packets are lost.
- R packets can supersede previous R packets, marking them no longer needed.
- Independent of codec structure.
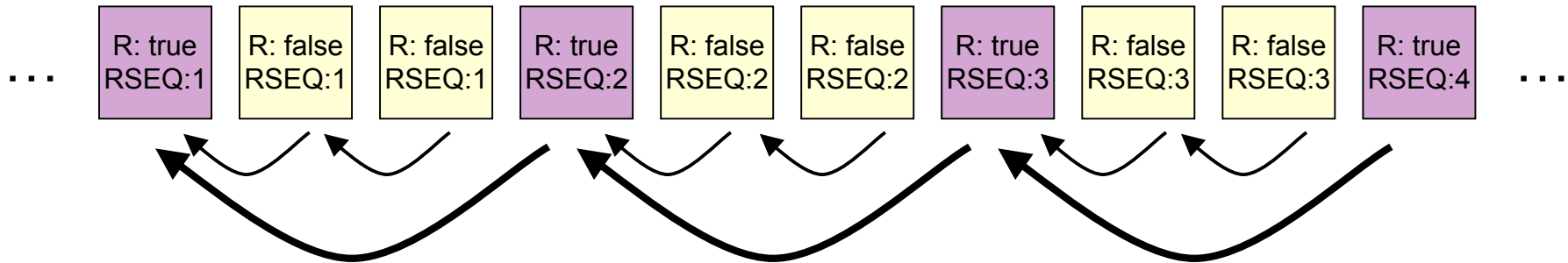- Multiple series of R packets.

Named extension header

- Mark packets with a named extension header (ietf-avt-rtp-hdrext):

  "org.ietf.avt.r-packet/200606".

Header extension

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ID    |  len  |R| MBZ |  SER  |  RSEQ                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  SUPERSEDE_START [opt]        | SUPERSEDE_END [opt]          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- ID, len – common header of named extension
- R – is this an R packet
- SER – series of the R packet
- RSEQ – index of this R packet (R = true) / of most recent R packet in series (R = false)
- SUPERSEDE_* – whether this R packet supersedes earlier ones
  - Optional; presence indicated by length field

Example

| R: true RSEQ:1 | R: false RSEQ:1 | R: false RSEQ:1 | R: true RSEQ:2 | R: false RSEQ:2 | R: false RSEQ:2 | R: true RSEQ:3 | R: false RSEQ:3 | R: false RSEQ:3 | R: true RSEQ:4 |

- R packets have R = true, increment RSEQ by 1.
- Non-R packets have R = false, RSEQ of previous R packet.
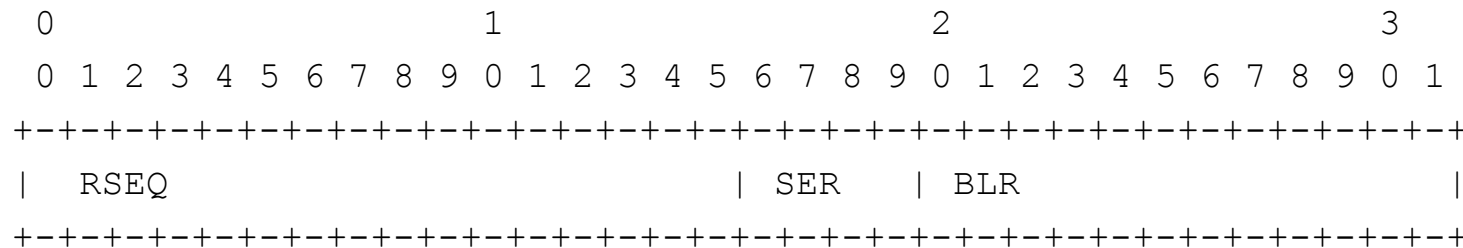
Example 2: Superseding
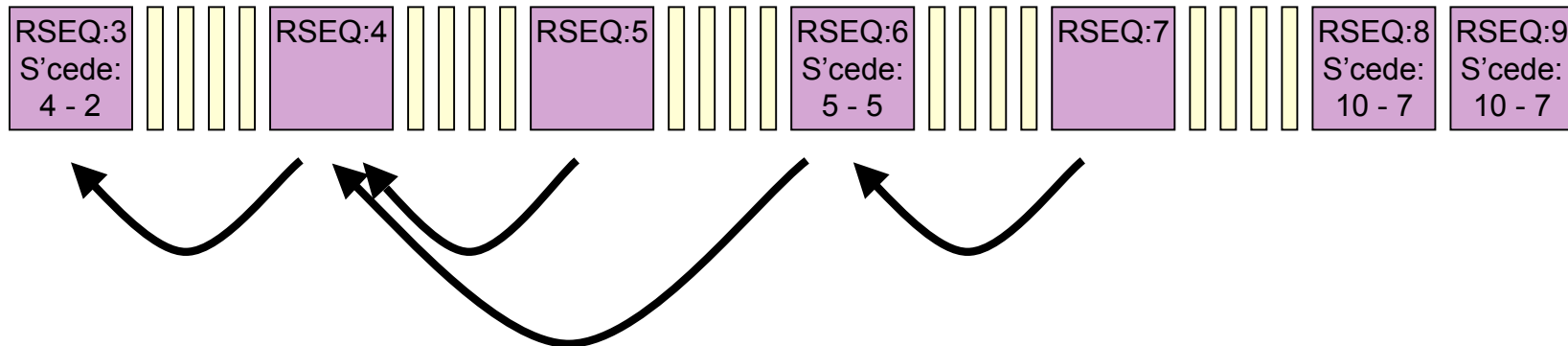
- R packets can supersede previous R packets: superseded ones no longer need to be received.
- R packets can either supersede a few previous R packets (explicit range) or all (start = seq + 1).
- If several R packets together supersede previous packets, they carry the same supersede range.

AVPF feedback packet: RNACK

- When an R packet is lost, send AVPF feedback message RNACK (R-packet negative acknowledgement): RTPFB, FMT=4.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RSEQ                         | SER   | BLR                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Similar to Generic NACK
  - ID of lost packet, bitmap of following lost packets.
  - Uses R sequence numbers.
  - Mentions series of R packets lost.

Open questions

- Is this a useful problem to solve?
- Is this the best way of solving this problem?
- Is the working group interested in seeing this developed further?