Reliable Internet Stream Transport

Kieran Kunhya - kierank@obe.tv

@kierank_

Introduction

- Goals
  - Low-latency video between professional devices (not necessarily end-users)
  - Packet recovery on a lossy network
  - Backwards compatible with existing MPEG-TS in UDP streams

RIST
RELIABLE INTERNET STREAM TRANSPORT

OPEN
BROADCAST SYSTEMS

Background

- Video Services Forum
  - (Closed) Industry forum of broadcast manufacturers
- Interest in replacing proprietary protocols (building on IETF)
- Published in October 2018
  - Unusually RIST published under Creative Commons Licence

VIDEO SERVICES FORUM

Why not TCP?

- A whole research field in itself!

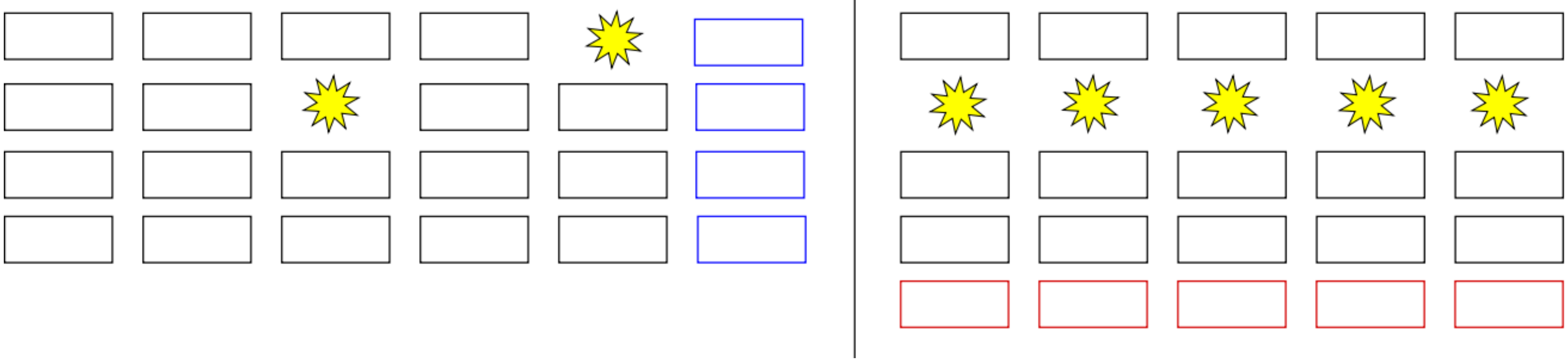
- Traditional TCP drops transmission rate with packet loss, quite aggressively in many cases
  - Not usually throughput drops in our use-cases
- UDP has native support for multicast
  - Many receivers requesting retransmits
- Backwards compatibility
  - Send RIST stream to existing devices
- Easier to do multipathing
  - Lets application handle the congestion-control decisions

Forward Error Correction (FEC)

- SMPTE 2022-1 FEC

Row FEC

Column FEC



- XOR based, very basic
- Can’t handle loss > matrix

Similar protocols/software

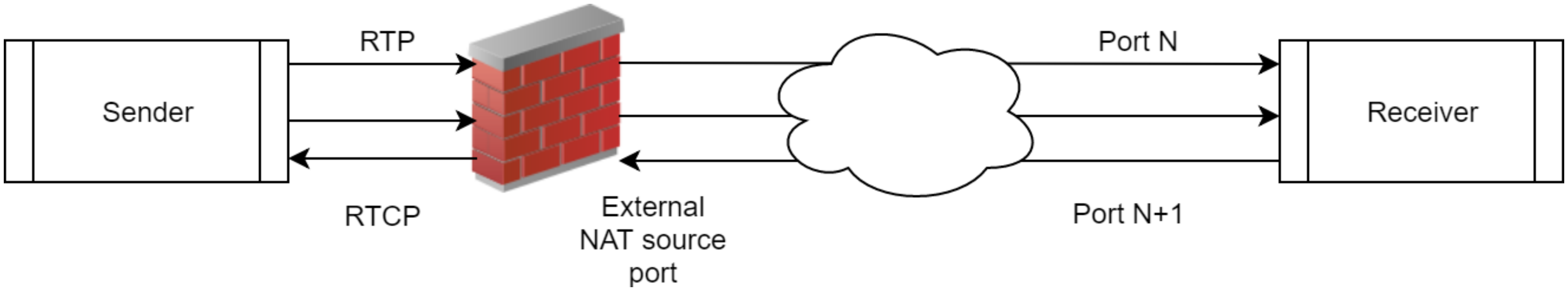
- Aggregatrtp (VideoLAN)
  - Allows aggregation of links with retransmission
  - From 2007
- SRT (Haivision)
  - Monolithic application
  - Very complex codebase around UDT (file transfer!)
  - Supports single-links only
  - Has encryption built-in
  - 89 page document!
- Many other proprietary solutions (mixing FEC and retransmissions)

SRT
SECURE
RELIABLE
TRANSPORT

RTP/RTCP

- Uses port N (RTP – main) and N+1 (RTCP – control)
- Sender periodically sends RTCP packets to keep state alive
  - Receiver sends Receiver Report RTCP packets
  - Allows for NAT "traversal"
- This can be over multiple links

RTP/RTCP

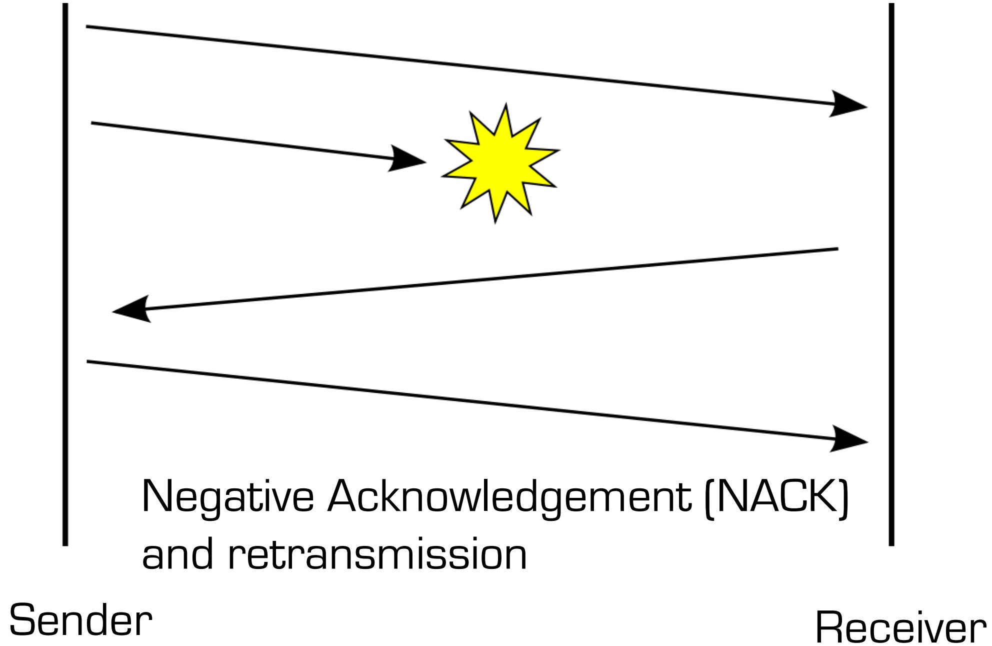


- Receiver sends to external (possibly NATed) source port, stateful NAT passes through

- Exactly like DNS

Acknowledgements

- Every RTP packet has a sequence number (16-bit) for identification



Bitmask NACK

```
 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|0|   FMT=1   |      PT=205   |          length=4             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         SSRC of packet sender (ignored by RIST sender2)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0xAA     |      0xBB     |      0xCC     |  0x00 or 0x01 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          PID=100              |1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          PID=117              |0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Uses base RTP sequence number + bitmask
- Useful for “random” packet loss

Range NACK

```
 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|0|Subtype=0|   PT=APP=204  |             length=4          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0xAA     |      0xBB     |      0xCC     |  0x00 or 0x01 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    0x52  (R)  |    0x49  (I)  |    0x53  (S)  |    0x54  (T)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Start=100              |         Additional=0          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Start=103              |         Additional=19         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Custom RTCP packet (RIST)
- Allows for range of loss

Retransmissions

- Signalling retransmission by setting LSB of SSRC to 1
  - Really strange way of doing it, should have used RFC 4588
- Decoder can identify retransmissions by SSRC
- Rate of retransmit requests implementation defined within latency bounds
  - Also how to handle “thundering herd” of retransmits

Implementations

- Open Source
  - Upipe (in examples)
  - VLC rist://
- Proprietary
  - VideoFlow
  - Zixi
  - Couple of other hardware ones

OPEN BROADCAST SYSTEMS

Future work

- Encryption (probably DTLS)
- Null packet removal
- Encoder bitrate changes
- Pull mode (?), otherwise end user can implement

- Scalable video (hard in practice)
- RIST on uncompressed video (completely crazy!)

OPEN
BROADCAST SYSTEMS

Live demo!

Credits

- Thanks to Adi Rozenberg (VideoFlow) and Rafaël Carré for their comments

OPEN
BROADCAST SYSTEMS