SCReAMv2

IETF-119 CCWG

Ingemar Johansson, Ericsson AB
ingemar.s.johansson@ericsson.com
Topics

- 5G properties
- Video coders
- SCReAMv2
- Source and experiments
5G properties

- Resource allocation in frequency and time (average)
  - End user applications may be bitrate limited
  - Resource allocation can drop in a few RTTs, or in an instant when other users enter in the same cell

- Actual throughput (average)
  - Modulation and Coding Scheme (MCS) varies with channel quality, power limitation in uplink
  - Result, varying throughput
5G properties

- Cellular transport is subject to fast fading $\rightarrow$ throughput varies on short time scale
  - Trade off between large network buffer, high link utilization and small network buffer, reduced link utilization, pick one
  - Throughput can drop quickly
  - Large dynamic range in throughput
- (Dynamic) uplink scheduling of intermittent data
  - Increased delay
  - Reduced link utilization
- Delay can occur due to
  - Congestion, scheduling
  - Hand over, battery saving (DRX)
  - Retransmission on lower radio protocol stack layers
Video coder aspects

- Frame sizes typically vary
  - I-Frames can be large → GDR (Gradual Decoding Refresh) highly recommended
  - Also P-frames vary in size
- Additional headroom required to cope with varying frame sizes → avoid excessive queue build-up for large frames
- Video has a max bitrate → congestion control becomes application limited
- Odd features: Systematic error in output rate, slow rate change.

Example, video frames transmission over a constrained bottleneck

One frame should ideally be transmitted before a new frame is generated
SCReAMv2

- Extensive update to RFC8298
  - Simplified, more stable
  - L4S designed in, instead of patched in
  - Goal, obsolete RFC8298
- Bonus: Provide some BCP on congestion control for multimedia

- Congestion control based on
  - Estimated one way delay (similar to LEDBAT CC)
  - Packet loss detection
  - Classic ECN
  - L4S (main focus)
- Sender transmission control soft limits bytes in flight
  - Max bytes in flight = 1.5x CWND
  - Media bitrate is (mainly) based on CWND and RTT
  - Packet pacing rate >= 1.5x media bitrate
  - Congestion window validation
  - RTCP: RFC8888
Congestion window update

On congestion event, max once per RTT

▪ Packet loss or classic ECN-CE marking
  ▪ Reduce CWND by fixed value

▪ Delay based congestion control
  ▪ Average of estimated queue delay to avoid over-reacting to non-congestion events like hand over
  ▪ Virtual L4S marking based back CWND decrease when average queue delay > target delay/2

▪ L4S based congestion control
  ▪ Reduce CWND proportional to fraction of packets marked (like Prague)
  ▪ Very small CWND → limit reduction

Once per RTT

▪ Increase CWND similar to Prague (Reno)
  ▪ One MSS / RTT
  ▪ In addition, multiplicative increase if uncongested → faster convergence to increased capacity
  ▪ Very small CWND → limit increase

▪ Non-L4S. Stabilize CWND with inflexion point, similar to Cubic.

▪ Validate CWND for the case that max media bitrate is reached.
Target bitrate update

- Executed when congestion window is updated
- \( \text{targetRate} = \text{CWND} \times 8 / \text{RTT} \)
  
  \( \text{CWND} \) [byte], \( \text{RTT} \) [s]

- Additional down-scaling based on 75%-ile of normalized frame size

- For not-L4S: scale down additionally when bytes in flight > CWND

- For very small CWND: Scale down target rate slightly
Delay based and L4S based congestion control

- Congestion control is delay based when L4S is not enabled or marking does not occur.
- Delay based CC should be sufficiently responsive but should not over-react on non-congestion events.
- Note: Bytes in flight can exceed CWND.
Multiplicative increase

- Multiplicative increase gives faster convergence when link throughput increases
- $0.05 = \text{up to 5\% CWND increase per RTT}$
Packet pacing

- Default pacing overhead is 50%, i.e. pacing rate is 1.5x media bitrate
- In addition: Large frames → pace even faster
- Objective: Avoid that large frames are held unnecessarily in RTP queue
Congestion window validation

- SCReAM can often be application limited
  - Max video bitrate reached
- Congestion window growth becomes restricted when SCReAM becomes application limited
- Example, max target rate is 10Mbps
- Note, relax the restriction when uncongested and max target bitrate not reached
Sources


- Code: [https://github.com/EricssonResearch/scream](https://github.com/EricssonResearch/scream)
  - Continuously developed since March 2015
  - SCReAM code
  - BW test application, the plumber’s aid
  - Multicamera gstreamer and C++ wrapper (multicam)
  - Complete gstreamer with multicam support (gstscream)
Experimentation, so far pretty much L4S-ish

- Small RC cars
  - [https://www.youtube.com/watch?v=RZmS10djDEg](https://www.youtube.com/watch?v=RZmS10djDEg)
- Large RC Cars, demo at MWC 2023
  - [https://www.youtube.com/watch?v=H8CBOKgHTOQ](https://www.youtube.com/watch?v=H8CBOKgHTOQ)
- Boat Attack cloud rendered gaming

- On the wish list, integration into WebRTC
The impact of L4S, log-data from car MWC 2023 demo

- 4 video streams with varying priority
- Varying background load
- L4S reduces latency spikes considerably
Questions, comments?

65°30'37.9"N 22°22'27.1"E