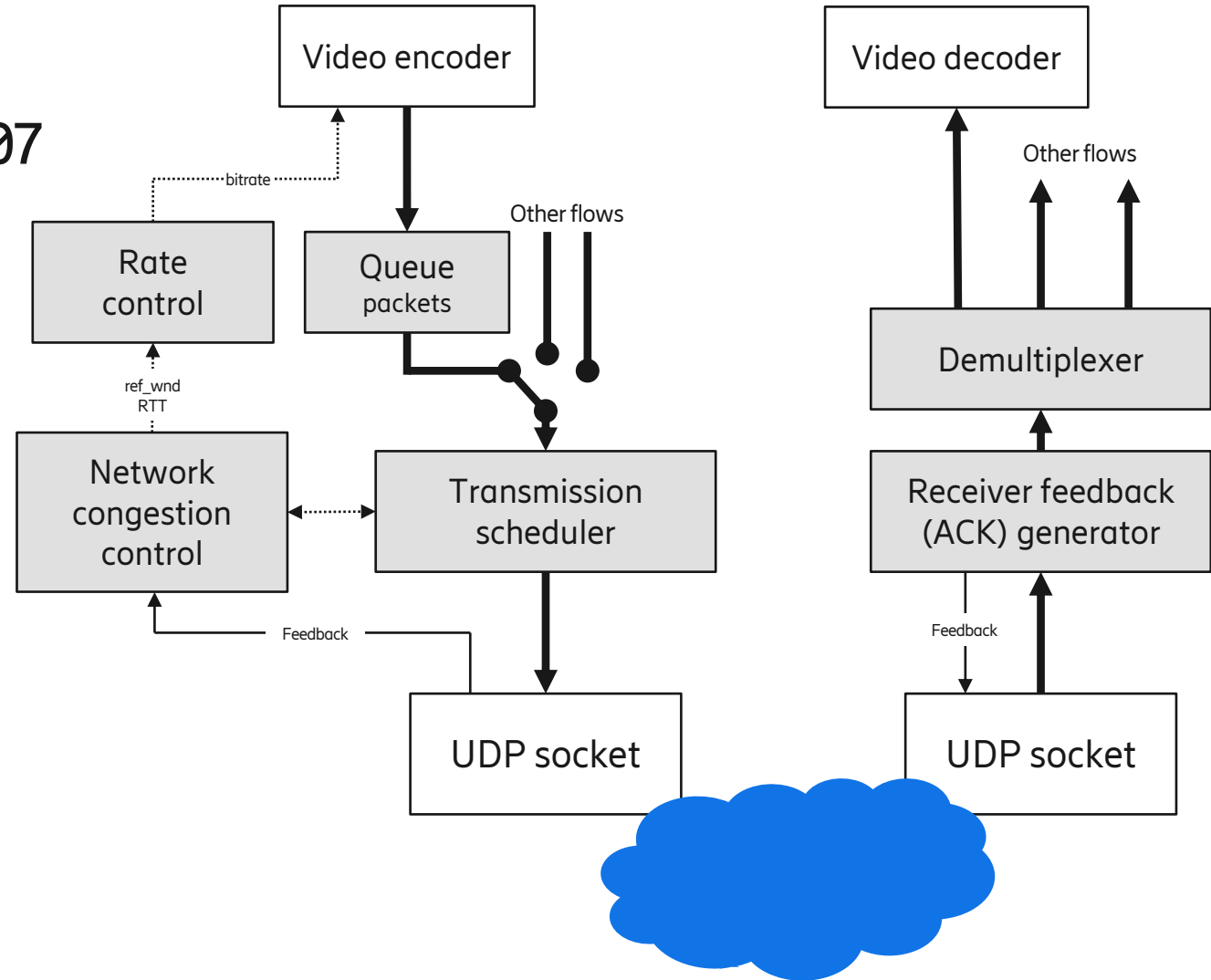


# SCReAM v2

draft-johansson-ccwg-rfc8298bis-screamv2-07

IETF-125 Shenzhen, China

Ingemar Johansson  
Magnus Westerlund  
Mirja Kühlewind  
Ericsson AB



# Basic function

## Network congestion- and transmission control

- Bytes in flight can be higher than `ref_wnd`
  - Avoid unnecessary sender side queuing of large video frames
  - New: **Adaptive `ref_wnd_overhead`**
- `ref_wnd` increase is continuous
  - $< 1\text{MSS}/\text{RTT}$  if close to last known max
  - $> 1\text{MSS}/\text{RTT}$  if long since congested (multiplicative increase)
  - New: **Additional restriction when congested**
- `ref_wnd` decrease on congestion
  - Packet loss detected
  - ECN/L4S
  - Estimated queue delay increases
- Packet pacing  $\sim 150\%$  of `target_bitrate`
  - Pacing rate increased when uncongested.

## Media rate control

- Objective: maintain just about enough data to fill available capacity
- $\text{target\_bitrate} = 8 * \text{ref\_wnd} / \text{s\_rtt}$
- Challenging part
  - Media coder can have a slow rate control loop
  - Large I-frames complicate congestion control
    - Gradual decoder refresh recommended
  - Sometimes media rate  $\neq$  `target_bitrate`
- Sender side queue can be discarded if too large
- New : **Additional method to avoid excessive data unit queue build-up**

# Progress since IETF-124

## draft versions -06 -- -07

- New metric **qdelay\_dev\_norm**
  - Indicates level of congestion (queue build-up)
  - Used for **ref\_wnd\_overhead** and restriction of **ref\_wnd** growth.
  - Helps to reduce rate variations and queue delay when..
    - non-L4S capable
    - L4S marking thresholds are high
    - large/frequent variations in link capacity
- Adaptive **ref\_wnd\_overhead**
  - Restricts bytes in flight when congested (**qdelay\_dev\_norm** is high)
    - Reduced rate variations, reduced queue delay
  - Allows bytes in flight to exceed **ref\_wnd** more when uncongested (**qdelay\_dev\_norm** is low)
    - Reduced application e2e delay → beneficial when non-congested or with e.g. virtual queue L4S marking

# Progress since IETF-124 draft versions -06 -- -07

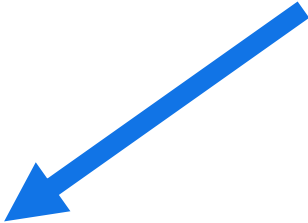
- Restrict ref\_wnd growth when (qdelay\_dev\_norm is high)
  - Reduced rate variations, reduced queue delay
- Adaptive adjustment of target rate to avoid excessive data unit queue build-up. Addresses..
  - Systematic mismatch between target and actual bitrate in media coders
  - Slow rate control in media coders
- Description of method for clock drift compensation

# Missing + additional improvements

## To be added in later draft versions

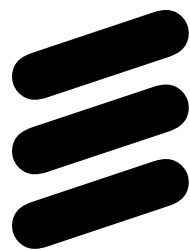
- Calculation of `qdelay_dev_norm` can likely be improved.  
Challenges :
  - Clock drift
  - Non-congestion related delay jitter, hard to distinguish against congestion related delay jitter
- Handling of extreme queue delay outliers
- Make pacing gain a function of `qdelay_dev_norm` ?
  - Istf relation between target bitrate and max bitrate
- Potential additional improvements pending experiments with L4S + SCReAM in WebRTC

# WebRTC implementation by Google WebRTC team

- Available in chrome canary builds  
<https://www.google.com/chrome/canary/>
  - Enabled with :  
`/opt/google/chrome-canary/chrome --enable-logging=stderr --disable-webrtc-encryption --vmodule=scream_v2=1 --force-fieldtrials="WebRTC-RFC8888CongestionControlFeedback/Enabled,offer:true/WebRTC-Bwe-ScreamV2/Enabled/" https://l4s-testbed-4smqe.ondigitalocean.app/`
  - Start chrome canary on two hosts with some kind of real or emulated channel inbetween
  - See **chrome://webrtc-internals/** for SCReAM + L4S logs
  - For the bleeding edge fans  
<https://webrtc.goglesource.com/src/+refs/heads/main/docs/native-code/development/#peerconnection>
  - Chat group "L4S in WebRTC"  
<https://chat.google.com/u/0/app/chat/AAQAe-dl2QXI>
- 

# WG item ?

- Algorithm is well tried out
  - Algorithm needs to address both transport layer as well as application layer media rate control for best performance.
- Evaluation in 5G L4S product
- Remote control car demos
- Trial test implementation in WebRTC, more extensive testing expected
- Minor changes in algorithm recently
  - Additional (minor) improvements may come based on results from experiments



**ERICSSON**

# @IETF-124: Progress since IETF-119

## draft versions -01 -- -05

- Draft is mainly transport protocol agnostic
- Major clean-up
- Discussion on large reference window overhead and “sluggish” video encoders
- Discussion on issues with clock drift and clock skipping
- Delay based CC is always active
- Modified delay based virtual L4S marking backoff
- Slow down ref\_wnd growth when close to last known max value
- ref\_wnd decrease/increase limited when ref\_wnd/mss is small
- Fast attack slow decay filter for l4s\_alpha
- ref\_wnd increase limited if L4S is likely non-active and queue delay increases