

Updates on SCReAM- An implementation experience

[draft-ietf-rmcat-scream-cc-02](#)

Zaheduzzaman Sarker
Ingemar Johansson

Ericsson Research

AGENDA



- › Updates on the draft
- › Implementation of SReAM in OpenWebRTC
- › What's next?

CHANGES IN THE DRAFT

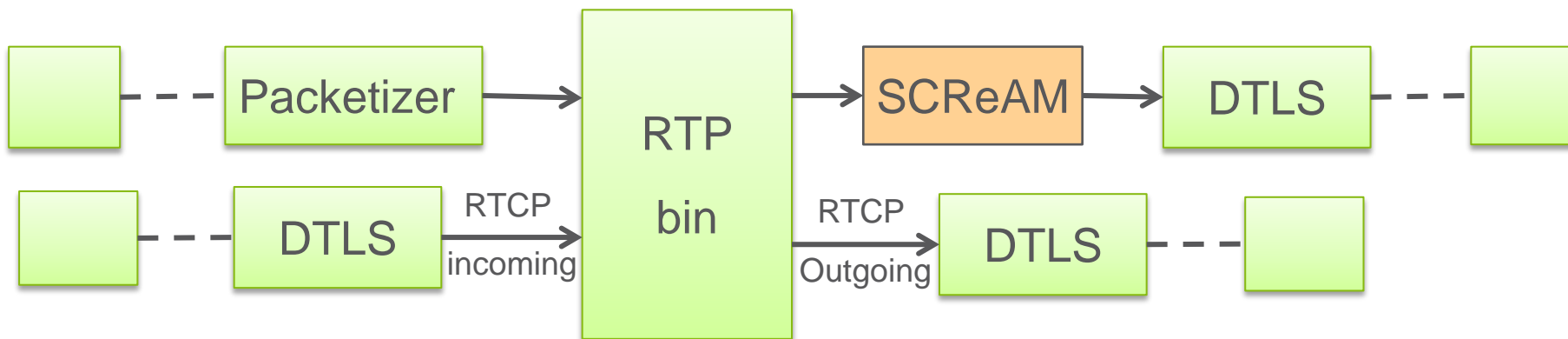


- › No changes in the actual algorithm
- › Rewritten and restructured the draft to
 - Increase readability
- › Addressed comments
 - Changed “sender transmission scheduling” to “sender transmission control”
 - › Now that section only describe SCReAM handling one stream but mentions the capabilities of handling multiple streams.
 - Added description on how to update the “bytes_newly_acked”
 - Added description on ECN usage
 - Added section for FEC and RTCP overhead consideration

IMPLEMENTATION STATUS (1/2)



- › We have implemented SCReAM in the OpenWebRTC (<http://www.openwebrtc.org/>)
 - <https://github.com/EricssonResearch/openwebrtc-gst-plugins>
 - We are testing the implementation with RMCAT test cases and will update the WG with detail results soon.
- › SCReAM is implemented as a gstreamer plug-in
- › The implementation experience has been good



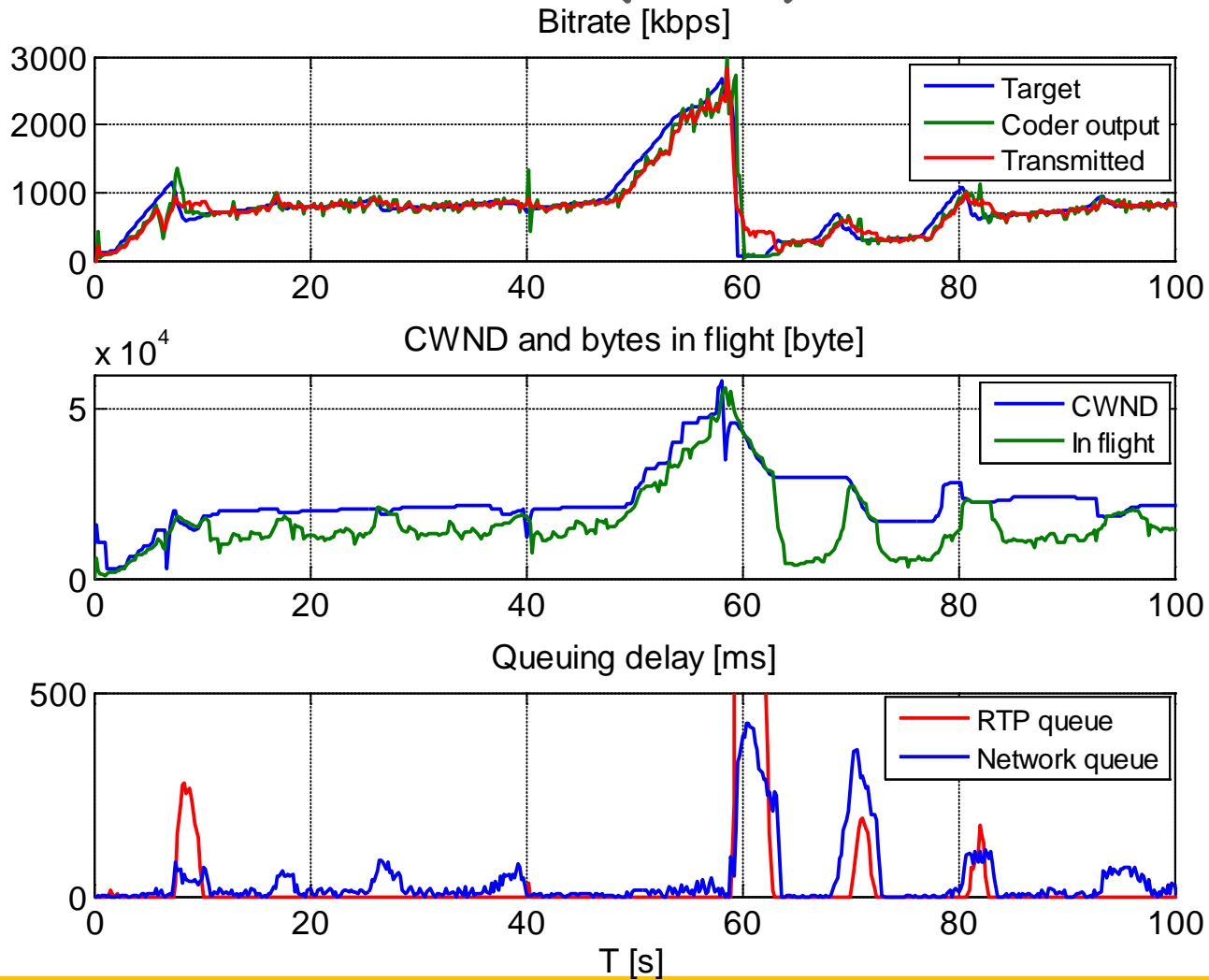
A very simplified view of the plug-in implementation

IMPLEMENTATION STATUS (2/2)



- › However, interaction towards video coded has been a big issue
 - SCReAM rate control is sometimes considerably faster than what the video rate control loop can deliver
 - This can cause unstable behavior
 - Either
 - › We need proper encoder configurations to make it more responsive towards rate change requests.
 - Or
 - › We need means to feed the SCReAM control loop with the information of the video encoder rate control

TEST CASE 5.1 (1/3)



Delay spikes due to lack of responsiveness of the video codec to the rate change request

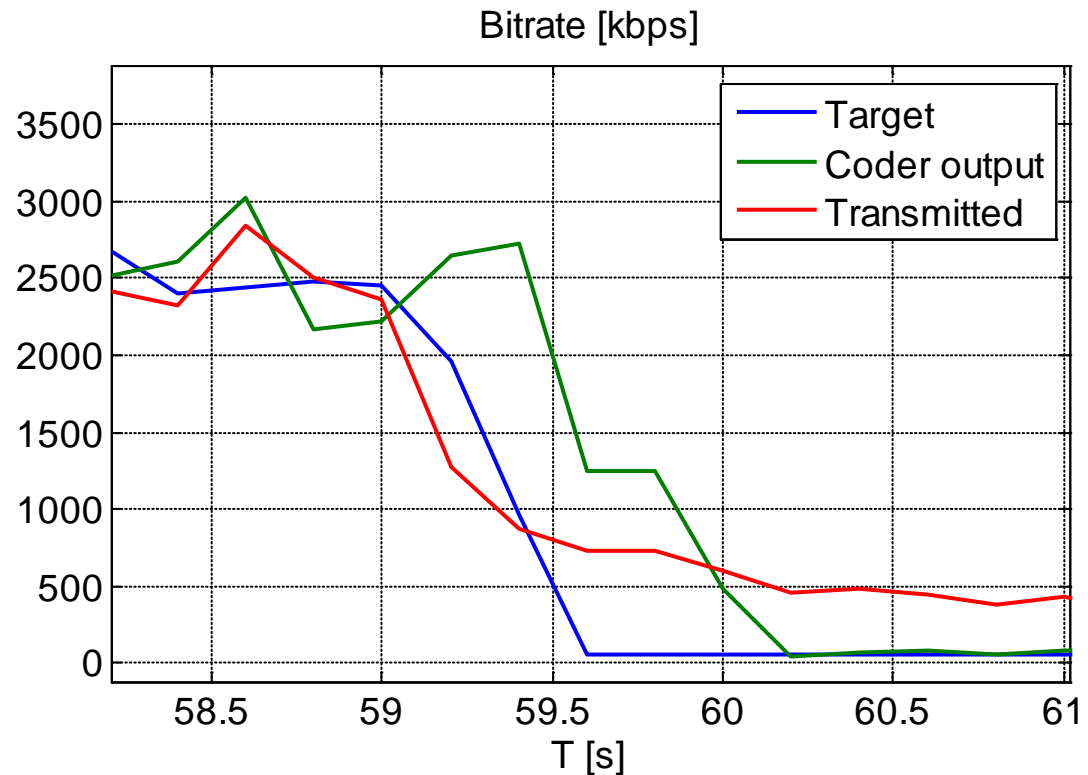
TEST CASE 5.1(2/3)



Video encoder (VP8)

Video encoder responsiveness when **rate decreases**

- Overshoots target rate with big margins (lags by more than 300ms)
- This leads to sharp decrease in the target rate
- More investigation needed to see how severe this problem is



It is expected to have better delay with more responsive encoder

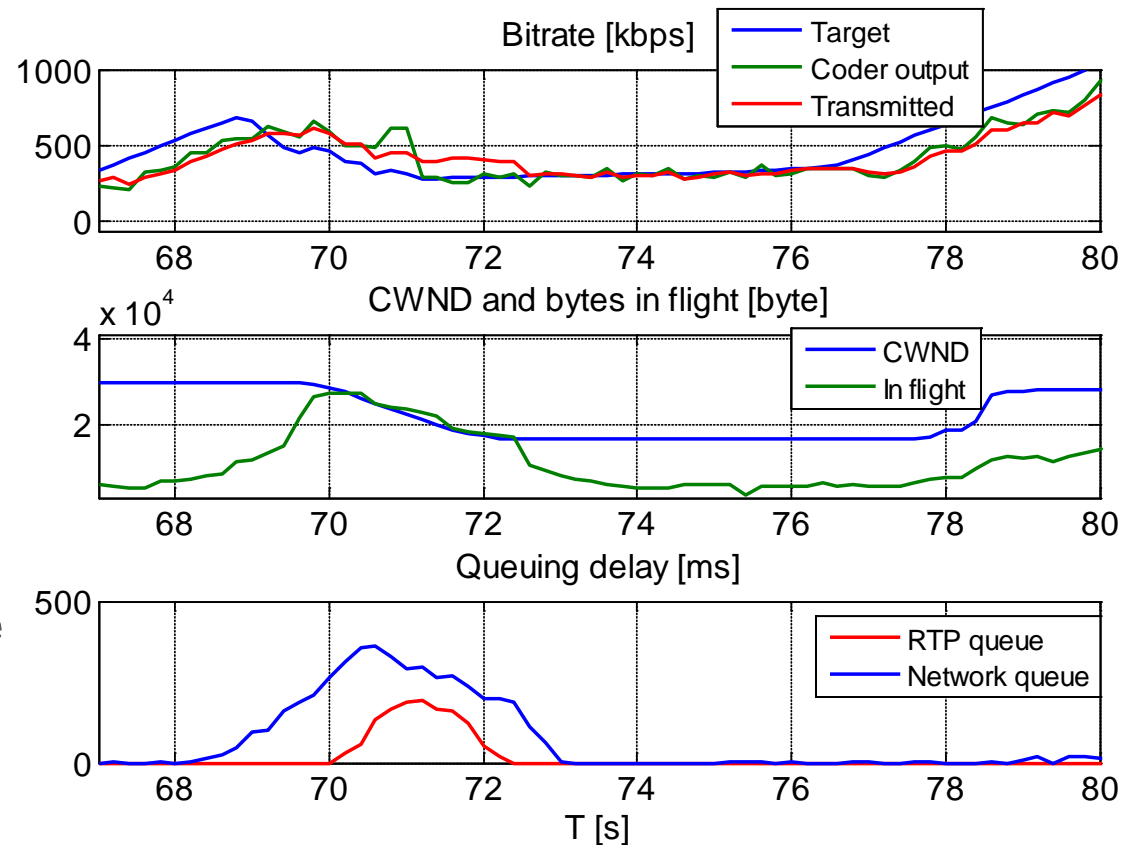
TEST CASE 5.1(3/3)



Video encoder (VP8)

Video encoder responsiveness when **rate increases**

- The encoder output rate lags behind the target rate by ~1 second.
- The SCReAM rate control is much faster than the video coder rate control loop
- Solution is either to decrease RAMP_UP_SPEED or to make VP8 rate control loop faster

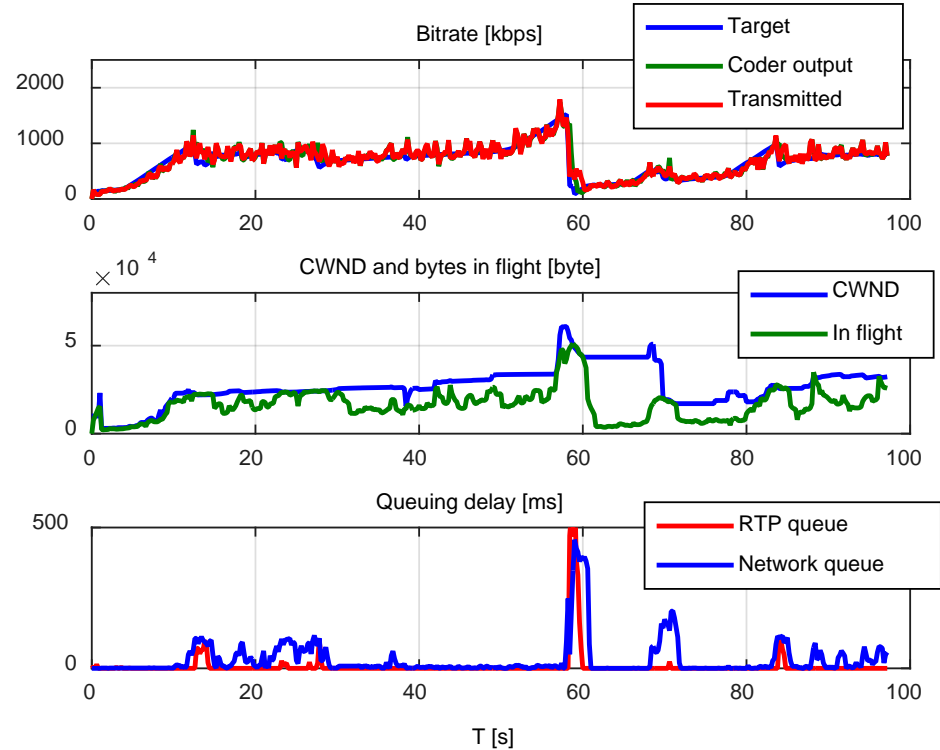
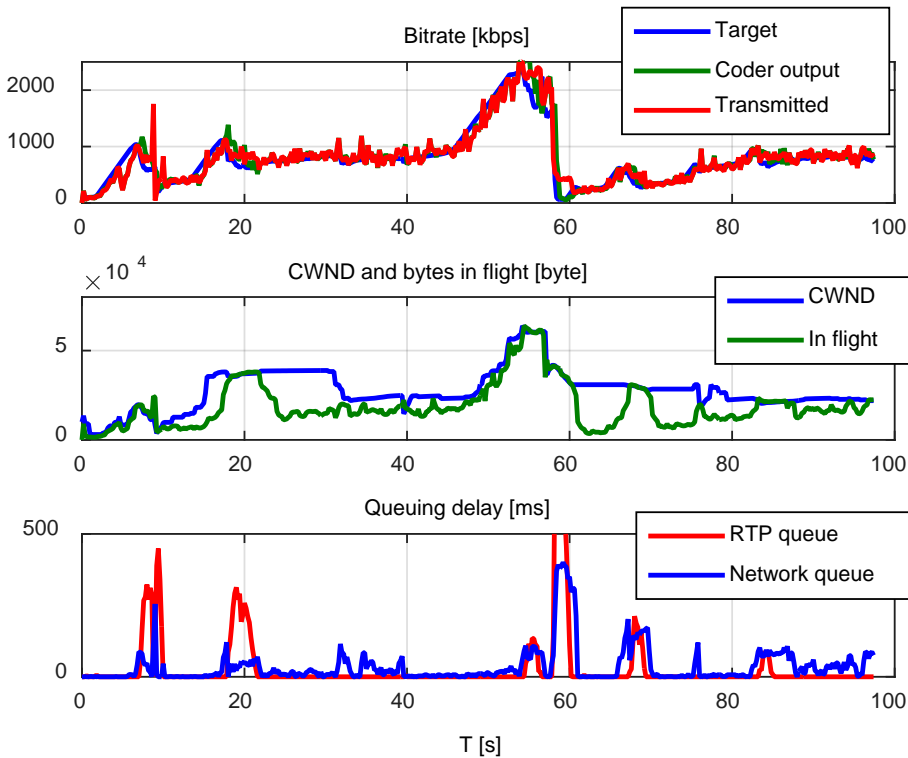


DIFFERENT RAMP_UP_SPEEDS



RAMP_UP_SPEED = 200kbps

RAMP_UP_SPEED = 100kbps



Less RTP queue spikes but slower increase affects the throughput

WHAT'S NEXT



- › Try out the SCReAM implementation
 - Get involved and give feedback on improvements
- › More results up coming stay tuned.

- › More wider reviews required on the draft to move to the next phase.
 - Please read and comment.



ERICSSON