

# SCReAM BW test tool

2020-03-20

Ingemar Johansson

Ericsson AB

[ingemar.s.johansson@ericsson.com](mailto:ingemar.s.johansson@ericsson.com)



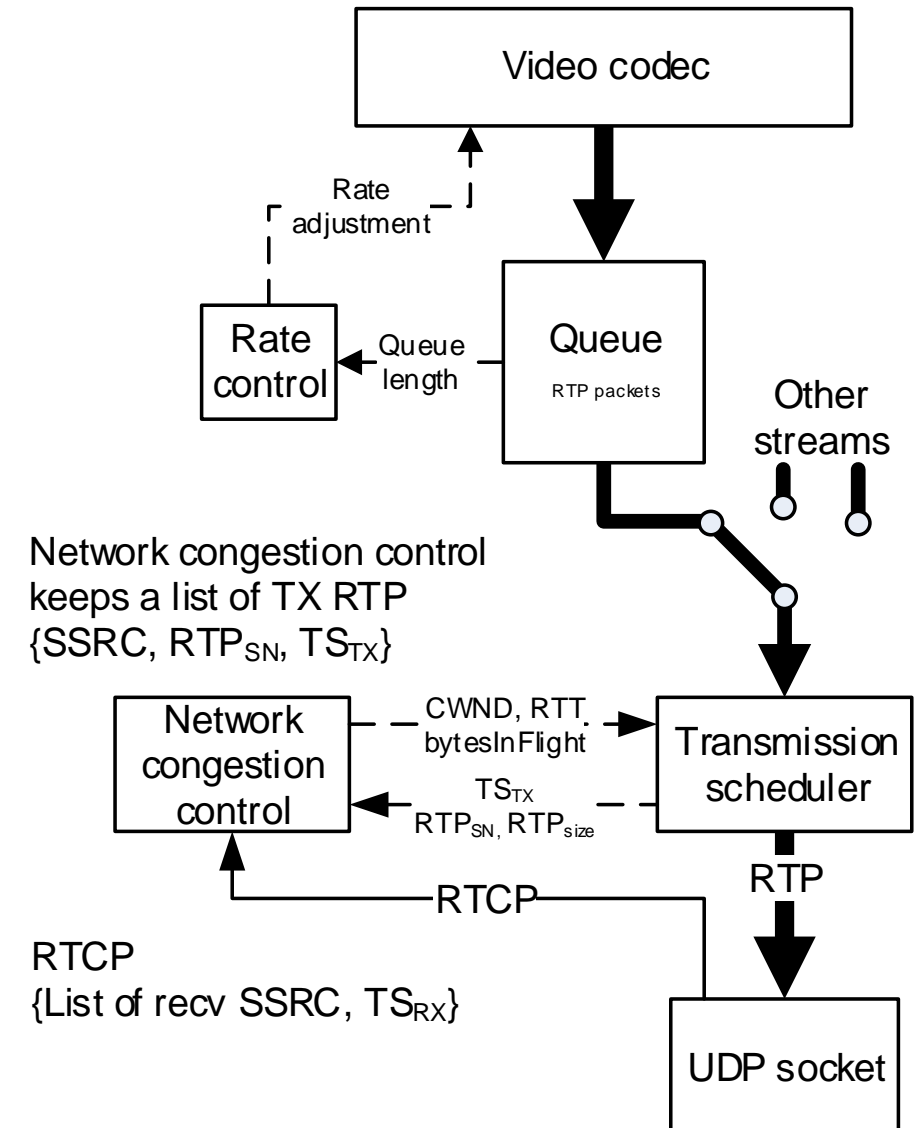
# SCReAM BW test tool



- The SCReAM BW test tool gives a measure of how high bitrate a reasonably good congestion controlled realtime media application can reach.
  - Enables testing of real time e2e performance in e.g. wireless access
  - Generates fake video frames (RTP media)
    - No media capture/rendering
    - Possible to generate key frames with a given interval
- SCReAM does congestion control
  - Default settings can be configured
  - Congestion control can be turned off completely
- Test tool is implemented on two Ubuntu PCs
  - One RTP sender
  - One Receiver
- Media rates up to ~150Mbps can be tested
  - Modest complexity (~20% CPU on a Intel i7 8-core @ 2.80GHz) at 80Mbps
- Application reports:
  - Bitrate
  - Loss
  - Estimated queue delay
  - RTT

# Key component : SCReAM

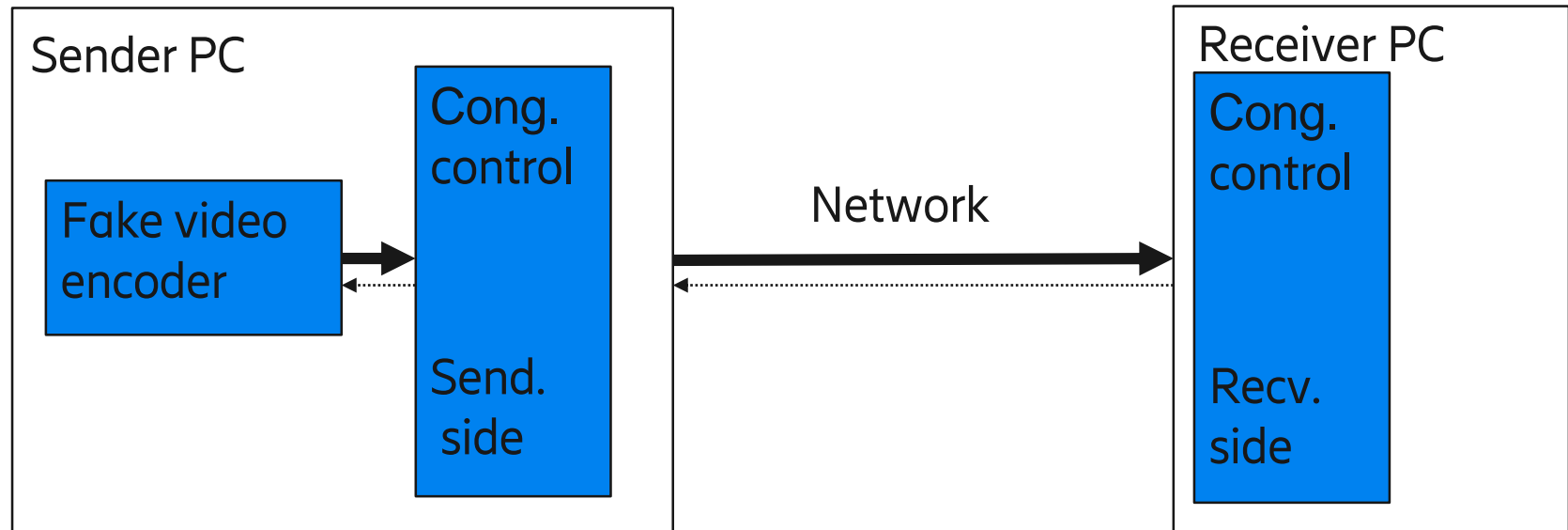
- SCReAM = **S**elf-Clocked **R**ate **A**daptation for **M**ultimedia
  - Window based congestion control → like TCP but without the retransmissions
  - Algorithm reacts on packet loss as well as delay
  - RTP packets can be queued up already in sender
- Developed since 2014
  - Design goal : Good performance for wireless access (LTE, 5G)
  - IETF RFC8298
- Available as open source
  - <https://github.com/EricssonResearch/scream>
- All RTP streams can be congestion controlled
  - Video, Audio, Haptics
- And... it works with ECN (Explicit Congestion Notification)
  - ECN makes it even more stable
  - And... it works with L4S!



# SCReAM test tool where ?



- Sender PC transmits fake RTP media
- Receiver PC generates feedback



# How to run it?



- Receiver side:  
> ./bin/scream-bw-test-rx <sender side IP> <port>
- Sender side:  
> ./bin/scream-bw-test-tx <receiver side IP> <port>
- Will run test with basic settings, more examples later

- Please note 1

The performance data (throughput, queue delay, RTT and packet loss) obtained by this tool reflects what you can roughly expect when you use SCReAM congestion control for real video streams

- Please note 2

Some Android based UEs have some odd NAT functionality that can re-map port numbers

To avoid this : Start the application on the modem side first

# Compile/build



- The SReAM-tester builds on Ubuntu
  - Sender and receiver built on different PC's
- cd to scream-bw-test folder
- cmake .
- make
- ./bin/scream-bw-test-tx is the sender application
- ./bin/scream-bw-test-rx is the receiver application

# Sender side



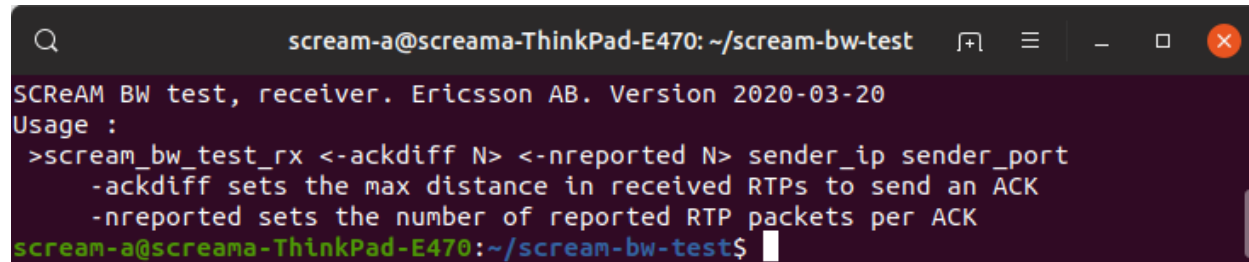
- Command line options explained to the right
- By default the tester will run in rate adaptive mode with a RTP media rate that is almost constant (no key frames), the max rate is 50000kbps. In addition a summary report is printed on stdout. Packet pacing is enabled by default
- A verbose, more detailed log can be printed on stdout with the *–verbose* option
- Detailed per-ACK log with *–log* option

- › For more truthful modeling of video
  - *–key interval multiplier*
  - Generates key frames every *interval/s* with a size *multiplier* times larger than the “P-frames”

```
scream-a@screama-ThinkPad-E470: ~/scream-bw-test
SCReAM BW test, sender. Ericsson AB. Version 2020-03-20
Usage :
> scream_bw_test_tx <options> decoder_ip decoder_port
  -nopace disables packet pacing
  -fixedrate sets a fixed 'coder' bitrate
  -key option set a given key frame interval [s] and size multiplier
    example -key 2.0 5.0
  -initrate sets a start bitrate [kbps]
    example -initrate 2000
  -minrate sets a min bitrate [kbps], default 1000kbps
    example -minrate 1000
  -maxrate sets a max bitrate [kbps], default 200000kbps
    example -maxrate 10000
  -rateincrease sets a max allowed rate increase speed [kbps/s],
    default 10000kbps/s
    example -rateinc 1000
  -ect n , ECN capable transport, n = 0 or 1 for ECT(0) or ECT(1),
    -1 for not-ECT (default)
  -scale value, scale factor in case of loss or ECN event (default 0.9)
  -dscale value, scale factor in case of increased delay (default 10.0)
  -delaytarget sets a queue delay target (default = 0.06s)
  -mtu sets the max RTP payload size (default 1200 byte)
  -fps sets the frame rate (default 50)
  -verbose print a more extensive log
  -nosummary don't print summary
  -log log_file save detailed per-ACK log to file
  -sierralog get logs from python script that logs a sierra modem
scream-a@screama-ThinkPad-E470:~/scream-bw-test$
```

# Receiver side

- Command line options explained to the right
- For very low bitrates (e.g VoLTE):
  - ackdiff 1 -nreported 1  
reduces size of ACKs and gives info about each received RTP packet



```
scream-a@screama-ThinkPad-E470: ~/scream-bw-test
SCReAM BW test, receiver. Ericsson AB. Version 2020-03-20
Usage :
>scream_bw_test_rx <-ackdiff N> <-nreported N> sender_ip sender_port
    -ackdiff sets the max distance in received RTPs to send an ACK
    -nreported sets the number of reported RTP packets per ACK
scream-a@screama-ThinkPad-E470:~/scream-bw-test$
```



# Sender output summary mode (default)



- The summary output presents the most important data every 2s
- Transmit rate
- Packet loss rate (PLR), two values
  - Average over a 5s window
  - Long term average (entire test session)
- RTT
- Queue delay, i.e. the estimated queue delay.

```
scream-a@screama-ThinkPad-E470: ~/scream-bw-test
Listen on port 50000 to receive RTCP from decoder
Scream sender started!
2.0 Transmit rate = 652kbps, PLR = 0.00% (0.00%), RTT = 0.018s, Queue delay = 0.004s
4.0 Transmit rate = 2087kbps, PLR = 0.00% (0.00%), RTT = 0.039s, Queue delay = 0.005s
6.0 Transmit rate = 5575kbps, PLR = 0.00% (0.00%), RTT = 0.049s, Queue delay = 0.006s
8.0 Transmit rate = 13093kbps, PLR = 0.00% (0.00%), RTT = 0.057s, Queue delay = 0.013s
10.0 Transmit rate = 17277kbps, PLR = 0.00% (0.00%), RTT = 0.064s, Queue delay = 0.018s
12.0 Transmit rate = 18941kbps, PLR = 0.00% (0.00%), RTT = 0.063s, Queue delay = 0.015s
14.0 Transmit rate = 20192kbps, PLR = 0.00% (0.00%), RTT = 0.062s, Queue delay = 0.014s
16.0 Transmit rate = 21269kbps, PLR = 0.00% (0.00%), RTT = 0.062s, Queue delay = 0.014s
18.1 Transmit rate = 22574kbps, PLR = 0.00% (0.00%), RTT = 0.070s, Queue delay = 0.029s
20.1 Transmit rate = 22459kbps, PLR = 0.00% (0.00%), RTT = 0.086s, Queue delay = 0.041s
22.1 Transmit rate = 22187kbps, PLR = 0.00% (0.00%), RTT = 0.077s, Queue delay = 0.026s
24.1 Transmit rate = 22628kbps, PLR = 0.00% (0.00%), RTT = 0.071s, Queue delay = 0.024s
26.1 Transmit rate = 22605kbps, PLR = 0.00% (0.00%), RTT = 0.068s, Queue delay = 0.019s
28.1 Transmit rate = 22940kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.019s
30.1 Transmit rate = 22594kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.020s
32.1 Transmit rate = 22387kbps, PLR = 0.00% (0.00%), RTT = 0.069s, Queue delay = 0.020s
34.1 Transmit rate = 22416kbps, PLR = 0.00% (0.00%), RTT = 0.066s, Queue delay = 0.017s
36.1 Transmit rate = 22845kbps, PLR = 0.00% (0.00%), RTT = 0.065s, Queue delay = 0.018s
38.1 Transmit rate = 23133kbps, PLR = 0.00% (0.00%), RTT = 0.075s, Queue delay = 0.031s
40.1 Transmit rate = 22582kbps, PLR = 0.00% (0.00%), RTT = 0.071s, Queue delay = 0.022s
42.1 Transmit rate = 22238kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.018s
44.1 Transmit rate = 22693kbps, PLR = 0.00% (0.00%), RTT = 0.065s, Queue delay = 0.018s
46.1 Transmit rate = 22883kbps, PLR = 0.00% (0.00%), RTT = 0.065s, Queue delay = 0.016s
48.1 Transmit rate = 23268kbps, PLR = 0.00% (0.00%), RTT = 0.068s, Queue delay = 0.023s
50.1 Transmit rate = 23094kbps, PLR = 0.00% (0.00%), RTT = 0.071s, Queue delay = 0.022s
52.2 Transmit rate = 22890kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.019s
54.2 Transmit rate = 23085kbps, PLR = 0.00% (0.00%), RTT = 0.071s, Queue delay = 0.028s
56.2 Transmit rate = 22676kbps, PLR = 0.00% (0.00%), RTT = 0.076s, Queue delay = 0.026s
58.2 Transmit rate = 22530kbps, PLR = 0.00% (0.00%), RTT = 0.070s, Queue delay = 0.027s
60.2 Transmit rate = 21427kbps, PLR = 0.00% (0.00%), RTT = 0.075s, Queue delay = 0.027s
62.2 Transmit rate = 21260kbps, PLR = 0.00% (0.00%), RTT = 0.068s, Queue delay = 0.020s
64.2 Transmit rate = 21884kbps, PLR = 0.00% (0.00%), RTT = 0.065s, Queue delay = 0.018s
66.2 Transmit rate = 22399kbps, PLR = 0.00% (0.00%), RTT = 0.073s, Queue delay = 0.033s
68.2 Transmit rate = 21550kbps, PLR = 0.00% (0.00%), RTT = 0.079s, Queue delay = 0.029s
70.2 Transmit rate = 21594kbps, PLR = 0.00% (0.00%), RTT = 0.070s, Queue delay = 0.021s
72.2 Transmit rate = 22177kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.019s
74.2 Transmit rate = 22945kbps, PLR = 0.00% (0.00%), RTT = 0.070s, Queue delay = 0.024s
76.2 Transmit rate = 22679kbps, PLR = 0.00% (0.00%), RTT = 0.072s, Queue delay = 0.022s
78.2 Transmit rate = 22526kbps, PLR = 0.00% (0.00%), RTT = 0.067s, Queue delay = 0.018s
80.2 Transmit rate = 22879kbps, PLR = 0.00% (0.00%), RTT = 0.069s, Queue delay = 0.024s
82.2 Transmit rate = 22867kbps, PLR = 0.00% (0.00%), RTT = 0.069s, Queue delay = 0.020s
84.2 Transmit rate = 22830kbps, PLR = 0.00% (0.00%), RTT = 0.071s, Queue delay = 0.022s
```

# Sender output verbose mode



- Detailed information is printed out every 200ms
- Enabled with `–verbose`
  - `-nosummary` disables the summary report
- 8 columns of data
  1. Time [s]
  2. Estimated queue delay [s]
  3. RTT [s] (average)
  4. Congestion window [byte]
  5. Bytes in flight [byte]
  6. Total transmit bitrate [kbps]
  7. Lost bitrate [kbps]

```
scream-a@screama-ThinkPad-E470: ~/scream-bw-test
41.925, 0.012, 0.064, 345787, 221223, 23605, 0,
42.1 Transmit rate = 22885kbps, PLR = 0.00%( 0.00%), RTT = 0.065s, Queue delay = 0.017s
42.126, 0.024, 0.065, 339721, 243311, 23290, 0,
42.327, 0.021, 0.065, 331517, 223840, 23208, 0,
42.527, 0.016, 0.066, 326825, 226246, 23170, 0,
42.728, 0.020, 0.065, 325940, 259028, 23071, 0,
42.928, 0.016, 0.067, 325940, 232536, 23060, 0,
43.129, 0.013, 0.066, 325940, 220416, 23049, 0,
43.330, 0.013, 0.066, 325940, 224808, 22889, 0,
43.531, 0.011, 0.065, 325940, 210320, 22665, 0,
43.731, 0.019, 0.063, 325940, 198272, 22455, 0,
43.932, 0.015, 0.064, 325940, 229756, 22332, 0,
44.1 Transmit rate = 22881kbps, PLR = 0.00%( 0.00%), RTT = 0.065s, Queue delay = 0.018s
44.133, 0.038, 0.065, 346601, 248647, 22557, 0,
44.334, 0.060, 0.071, 346735, 338433, 22988, 0,
44.534, 0.069, 0.077, 343578, 345762, 22849, 0,
44.735, 0.066, 0.086, 301746, 340626, 21942, 0,
44.936, 0.030, 0.088, 279004, 287412, 20866, 0,
45.137, 0.011, 0.083, 279169, 223252, 20550, 0,
45.337, 0.008, 0.074, 279169, 196426, 21016, 0,
45.538, 0.014, 0.069, 279169, 201309, 21233, 0,
45.739, 0.013, 0.066, 279169, 195249, 21110, 0,
45.939, 0.014, 0.064, 279169, 197353, 21256, 0,
```

# Detailed per-ACK log



- The detailed per-ACK log generates one log per incoming ACK, stored in a text file
- 13 columns of data
  1. Time [s]
  2. Estimated queue delay [s]
  3. RTT [s]
  4. Congestion window [byte]
  5. Bytes in flight [byte]
  6. Total transmit bitrate [bps]
  7. Stream ID
  8. Bytes newly ACKed
  9. Bytes newly ACKed and CE marked
  10. Media coder bitrate [bps]
  11. Transmitted bitrate [bps]
  12. ACKed bitrate [bps]
  13. Lost bitrate [bps]

```
• 0.071,0.000,0.065, 12500,7720,0,0,2424,0,0,0,0,0,
• 0.281,0.010,0.174, 12500,112,25343,0,15104,0,126715,25343,0,0,
• 0.349,0.016,0.068, 21106,8820,240232,0,12792,0,253449,240232,126734,0,
• 0.403,0.001,0.053, 35659,8932,418697,0,13892,0,405461,418697,382325,0,
• 0.491,0.010,0.062, 25106,7720,671988,0,13892,0,658752,671988,659978,0,
• 0.612,0.029,0.082, 25106,10144,1026836,0,12792,0,1013600,1026836,937632,0,
• 0.711,0.009,0.060, 25229,7143,1059941,0,14141,0,1033487,1059941,1066566,0,
• 0.811,0.008,0.060, 25229,5931,1058417,0,15137,0,1058417,1058417,815952,0,
• 0.940,0.033,0.086, 25229,11604,1105407,0,14836,0,1105407,1105407,840743,0,
• 1.039,0.014,0.065, 29084,11331,1157763,0,16512,0,1157763,1157763,1137264,0,
• 1.132,0.006,0.059, 30627,7665,1214285,0,17118,0,1214285,1214285,1211413,0,
• 1.253,0.023,0.077, 30627,10434,1271821,0,17754,0,1271821,1271821,1270809,0,
• .
• .
• .
• 30.535,0.013,0.060, 313866,166287,22373506,0,41847,0,22840770,22373506,22187054,0,
• 30.542,0.014,0.064, 313866,186891,22373506,0,10908,0,22840770,22373506,22187054,0,
• 30.549,0.023,0.070, 313866,197476,22373506,0,26664,0,22840770,22373506,22187054,0,
• 30.554,0.023,0.072, 313866,180508,22373506,0,16968,0,22840770,22373506,22187054,0,
• 30.554,0.012,0.058, 313866,169923,22373506,0,27553,0,22840770,22373506,22187054,0,
• 30.561,0.014,0.064, 313866,191739,22373506,0,37249,0,22840770,22373506,22187054,0,
• 30.561,0.016,0.062, 312510,182043,22373506,0,9696,0,22840770,22373506,22187054,0,
• 30.566,0.018,0.066, 312510,195375,22373506,0,19392,0,22840770,22373506,22187054,0,
• 30.572,0.020,0.071, 312510,188992,22871936,0,29088,0,22850056,22871936,22226894,0,
• 30.572,0.021,0.069, 312510,179296,22871936,0,9696,0,22850056,22871936,22226894,0,
• 30.578,0.012,0.062, 312510,177195,22871936,0,20281,0,22850056,22871936,22226894,0,
• 30.578,0.014,0.061, 312510,167499,22871936,0,29977,0,22850056,22871936,22226894,0,
• 30.585,0.015,0.065, 312510,195375,22871936,0,39673,0,22850056,22871936,22226894,0,
```

# Receiver side log



- Indication of lost packets
- The number of consecutive lost packets is
  - $SN_{received} - SN_{previous-received} - 1$

```
eplijoh@elxc0000033: ~/scream-bw-test/receiver
Packet(s) lost : 49873 was received, previous rcvd is 49869
Packet(s) lost : 51027 was received, previous rcvd is 51025
Packet(s) lost : 51411 was received, previous rcvd is 51407
Packet(s) lost : 51750 was received, previous rcvd is 51747
Packet(s) lost : 56544 was received, previous rcvd is 56542
Packet(s) lost : 56820 was received, previous rcvd is 56818
Packet(s) lost : 56826 was received, previous rcvd is 56820
Packet(s) lost : 57225 was received, previous rcvd is 57220
Packet(s) lost : 59041 was received, previous rcvd is 59037
Packet(s) lost : 59043 was received, previous rcvd is 59041
Packet(s) lost : 59317 was received, previous rcvd is 59315
Packet(s) lost : 59569 was received, previous rcvd is 59567
Packet(s) lost : 59572 was received, previous rcvd is 59569
Packet(s) lost : 60398 was received, previous rcvd is 60396
Packet(s) lost : 60682 was received, previous rcvd is 60679
Packet(s) lost : 60689 was received, previous rcvd is 60682
Packet(s) lost : 60696 was received, previous rcvd is 60689
Packet(s) lost : 60698 was received, previous rcvd is 60696
Packet(s) lost : 61224 was received, previous rcvd is 61218
Packet(s) lost : 61461 was received, previous rcvd is 61459
Packet(s) lost : 62024 was received, previous rcvd is 62022
Packet(s) lost : 63649 was received, previous rcvd is 63646
Packet(s) lost : 64617 was received, previous rcvd is 64615
```

# Other



- Firewalls may need to be opened up but it is most often not needed
  - Use e.g firewall-cmd
    - `sudo firewall-cmd --zone=public --add-port=30122/udp`
- Priority for SCReAM sender and receiver may need to be elevated to avoid that the SCReAM processes are halted by other processes
  - `>sudo nice -- 10 ./bin/scream-bw-test-rx .....`
  - `>sudo nice -- 10 ./bin/scream-bw-test-rx .....`
  - Note .. Two minus (-) characters !
- UDP sockets may need more memory at high bitrates to avoid packet loss
  - See <https://medium.com/@CameronSparr/increase-os-udp-buffers-to-improve-performance-51d167bb1360> for setting of `net.core.rmem_max`, `net.core.rmem_default`, `net.core.wmem_max` and `net.core.wmem_default`

# Examples



- Below a few examples, the options can be combined
- `>./bin/scream-bw-test-tx -verbose <receiver side IP> <port>`
  - Print statistics in verbose mode
- `>./bin/scream-bw-test-tx -log test.txt <receiver side IP> <port>`
  - Save detailed log to test.txt that can be visualized in matlab or octave
- `>./bin/scream-bw-test-tx -key 2 10 <receiver side IP> <port>`
  - Emulate a key frame (refresh frame) every 2s that is 10 times larger than the normal frame size
- `>./bin/scream-bw-test-tx -minrate 2000 <receiver side IP> <port>`
  - Set min bitrate to 2Mbps (default 100kbps)
- `>./bin/scream-bw-test-tx -maxrate 20000 <receiver side IP> <port>`
  - Set max bitrate to 20Mbps (default 100Mbps)
- `>./bin/scream-bw-test-tx -delaytarget 0.05 <receiver side IP> <port>`
  - Set delay target to 50ms (default 100ms), this makes the congestion controller more sensitive to congestion and gives lower jitter but can result in lower throughput

# Visualization of data

- With octave or matlab installed, it is possible to plot and visualize the performance measurements.
- Examples:
  - Load the data file generated by the -log option
  - `>> a = load('test.txt')`
  - Plot the throughput, queue delay and RTT, note that the drops in throughput every 30s is a SCReAM BW test feature.
    - `>> plot_thp_delay(a,[0 100], 55, 0.2)`
  - Plot a CDF of the RTT and queue delay
    - `>> plot_cdf(a,[0 100]), 0.1)`

