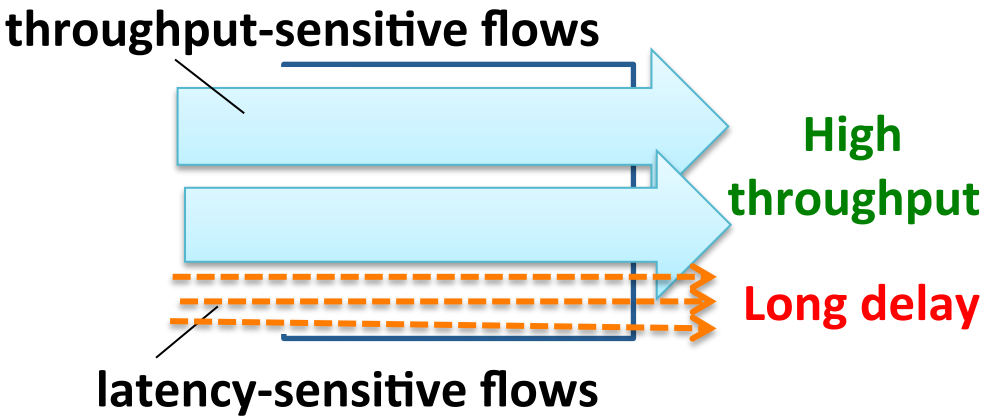
IETF87 Berlin

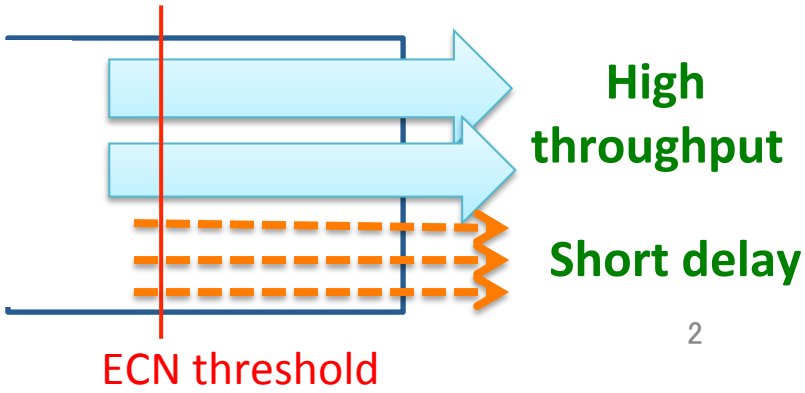
DCTCP implementation in FreeBSD

Midori Kato
Richard Scheffenegger
Lars Eggert
Alexander Zimmermann

DCTCP (Data Center TCP) overview

- Problem statement
  - Large FIFO buffers in the presence of bulk transfers impacts latency-sensitive flows
- DCTCP
  - Maintain the queue size short using ECN (Explicit Congestion Notification)
  - Endpoint-only mechanism
    - Switches/routers' mechanism is same as standard ECN





DCTCP algorithm

Sender side

1. Maintain the fraction of ECN marked seg. for each RTT

$$F = \frac{\#\ \text{of marked segs}}{\#\ \text{of segs}}$$

and update average fraction of marked seg. (α)

$$\alpha \leftarrow (1-g)\ \alpha + gF$$

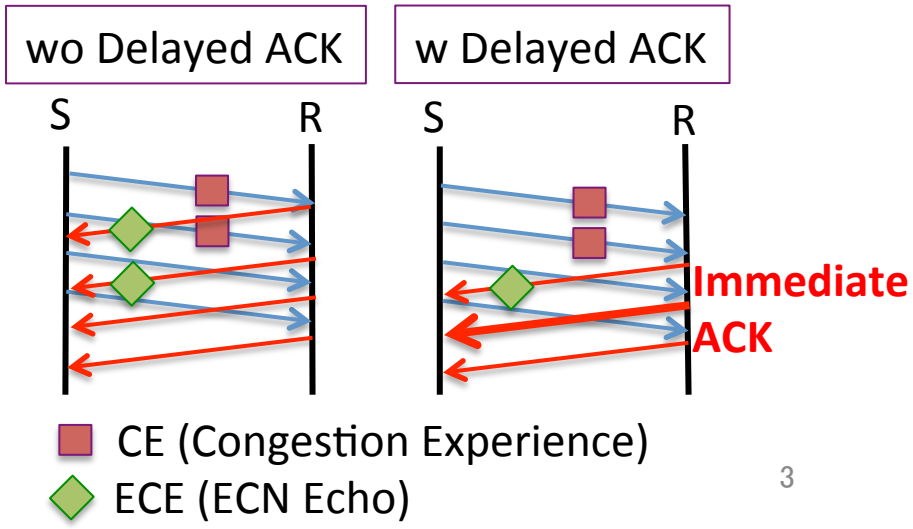
2. Adopt alpha to cwnd decrease

$$\text{cwnd} \leftarrow (1 - \frac{\alpha}{2})\ \text{cwnd}$$

** reaction to packet loss (e.g., dup ack and timeout) is same with NewReno

Receiver side

- Mark ECE only when CE packet is received
- send immediate ACK when CE state is changed (regardless of delayed ACK)



FreeBSD implementation status [1/2]

- Total # of lines: 479 lines
  - tcp_input.c (24 lines), tcp_output.c (12 lines), cc.h (4 lines), cc_dctcp.c (439 lines)
  - Add ECN handling functions to CC module
  - Add DCTCP as one of CC algorithms

- References for FreeBSD DCTCP
  - Released Linux patch [1]
  - DCTCP paper "Data Center TCP (DCTCP) [SIGCOMM'10]"

[1] http://simula.stanford.edu/~alizade/Site/DCTCP.html

FreeBSD implementation status [2/2]

- Design issues
  1. Cwnd behavior on first ECN — Topic in this talk
     - Halve or Ignore cwnd
  2. Reconsideration of alpha update interval
     - every RTT or more frequent (e.g., every ack)
  3. Alpha value handling after idle time
     - Discard or keep alpha
  4. Alpha calculation affected by receive buffer control
     - Change nothing or handle alpha as special case

In progress

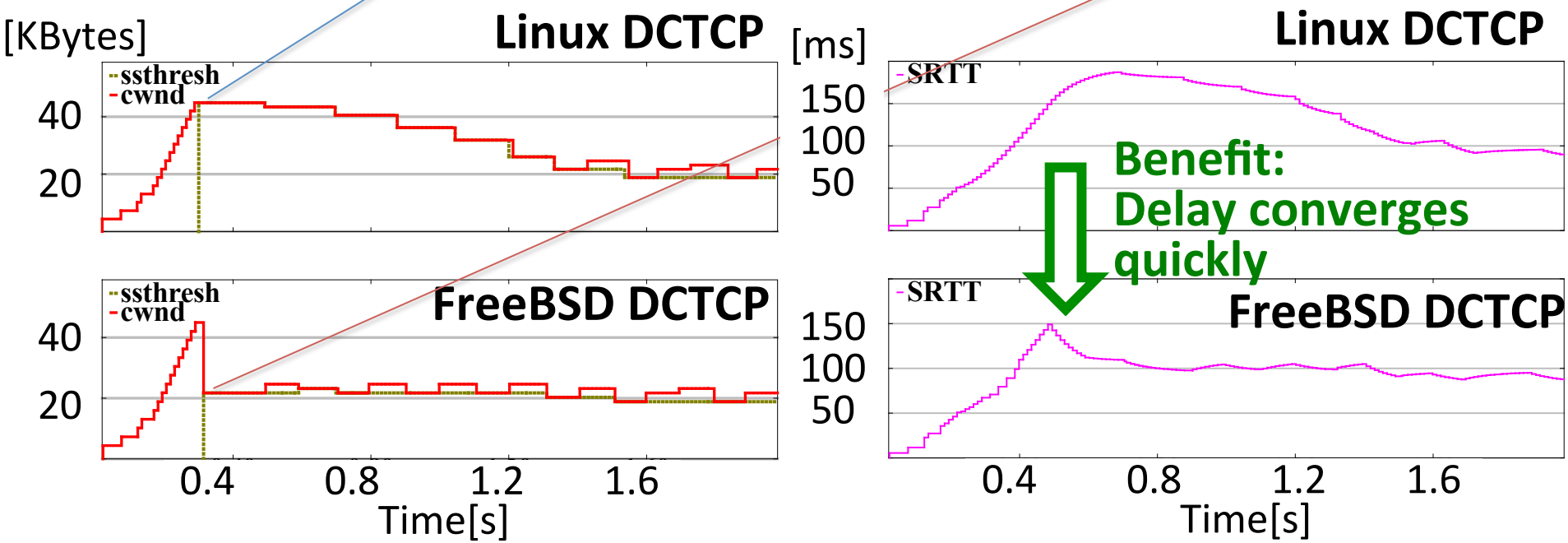
Open issue: cwnd behavior on first ECN

Option 1: DCTCP never reduces cwnd at first congestion recovery

$$\text{cwnd} \leftarrow (1 - \frac{\alpha}{2})\ \text{cwnd}$$

Option 2: halves cwnd at first congestion recovery

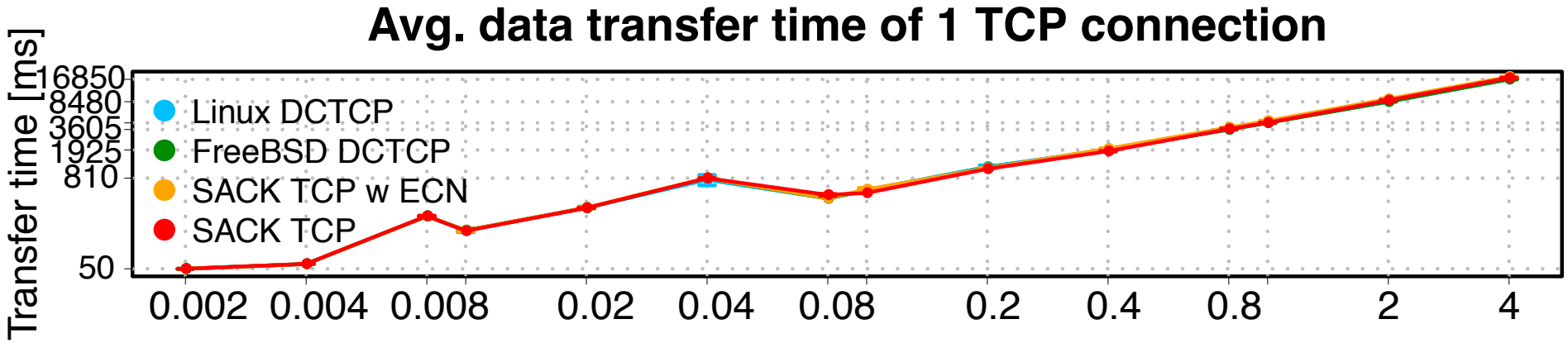
$$\text{cwnd} \leftarrow (1 - \frac{\alpha}{2})\ \text{cwnd}$$

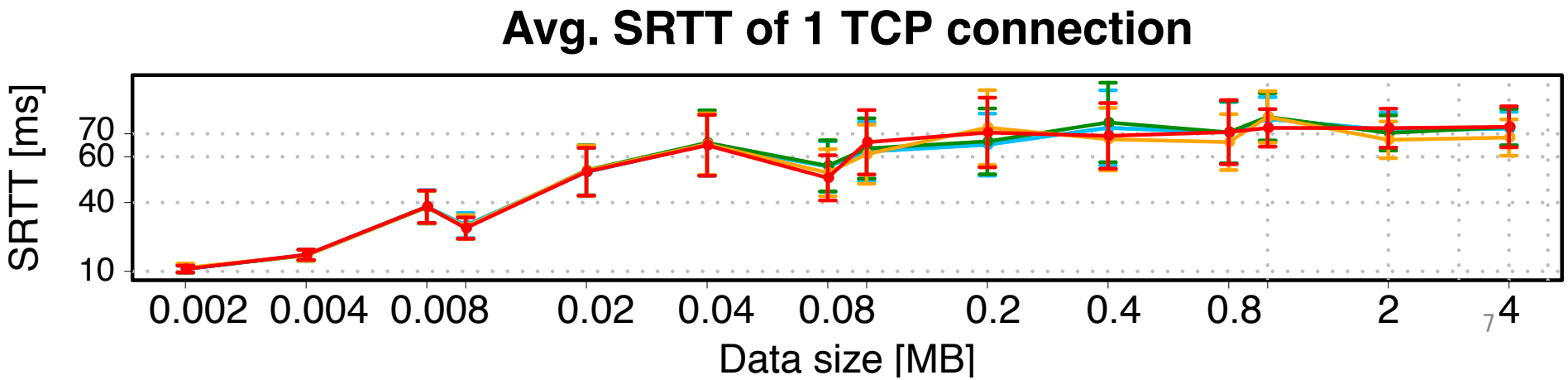


Performance with **small queue size**

Experiment
- Measure transfer time for *X* MB and SRTT (Smoothed RTT) using flowgrind

BW: 2Mbps, delay: 20ms
qlen: 7, ecn_thresh: 5

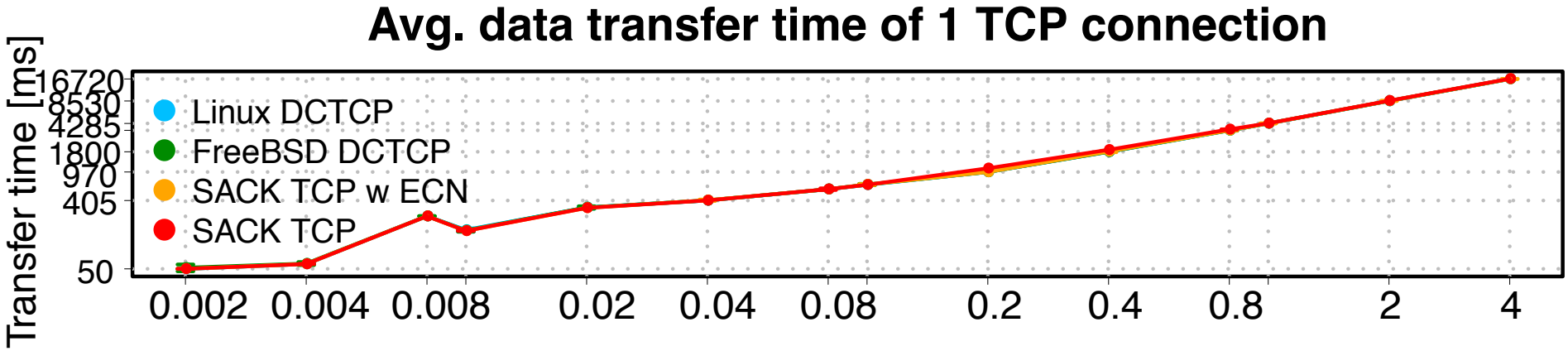


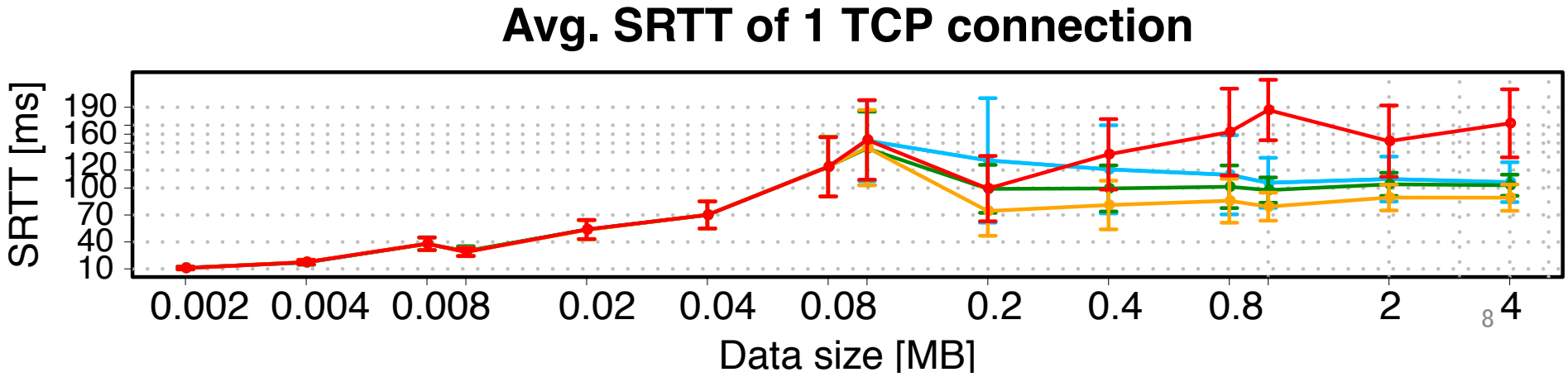


Performance with **large queue size**

Experiment
- Measure transfer time for *X* MB and SRTT (Smoothed RTT) using flowgrind

BW: 2Mbps, delay: 20ms
qlen: 30, ecn_thresh: 10





Summary

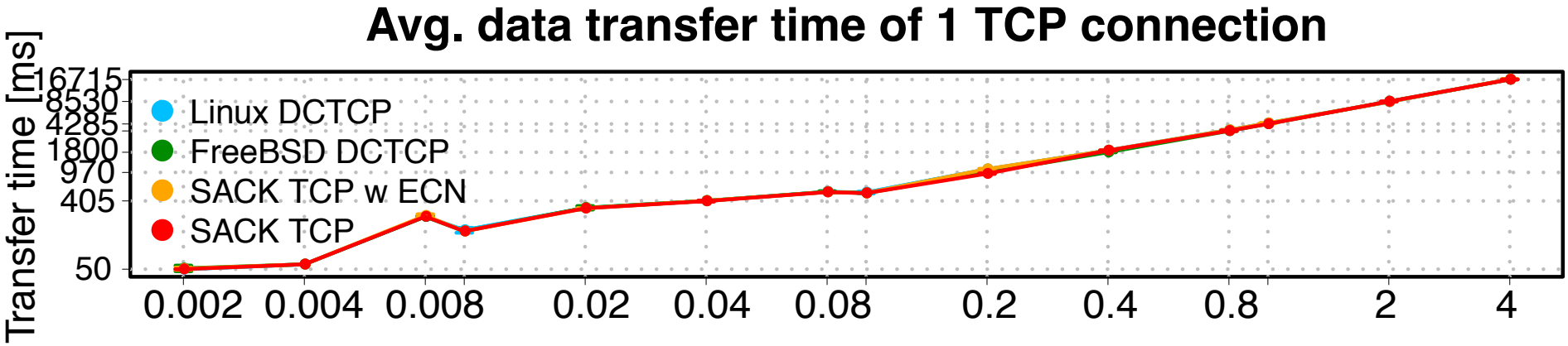
- Introduce DCTCP implementation in FreeBSD and design issues
- Plan to submit our code to FreeBSD

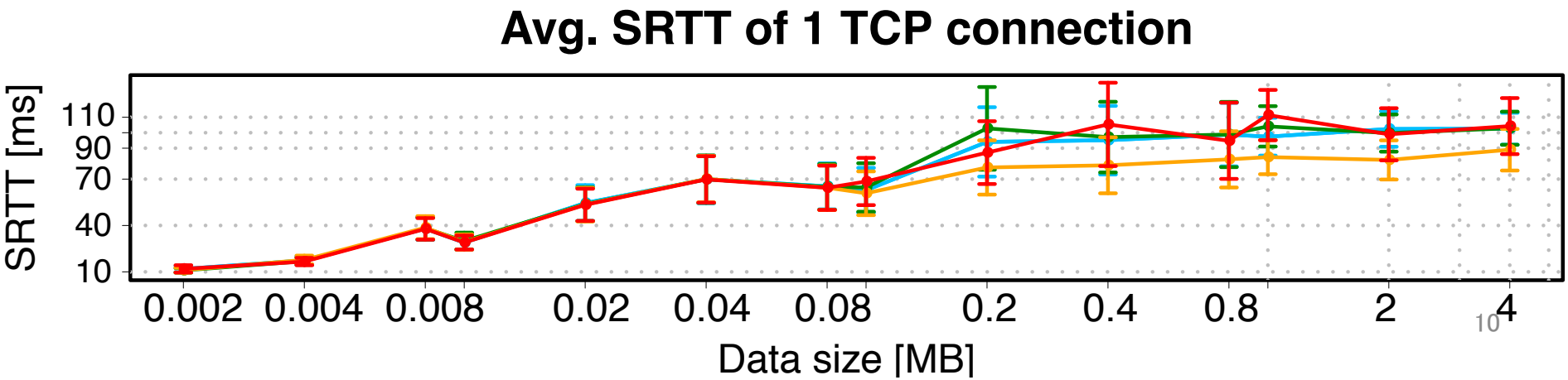
- Future work
  - Interaction with RFC3168
  - CUBIC and DCTCP
  - Change architecture for coexist of DCTCP and other CC algorithms

Performance evaluation

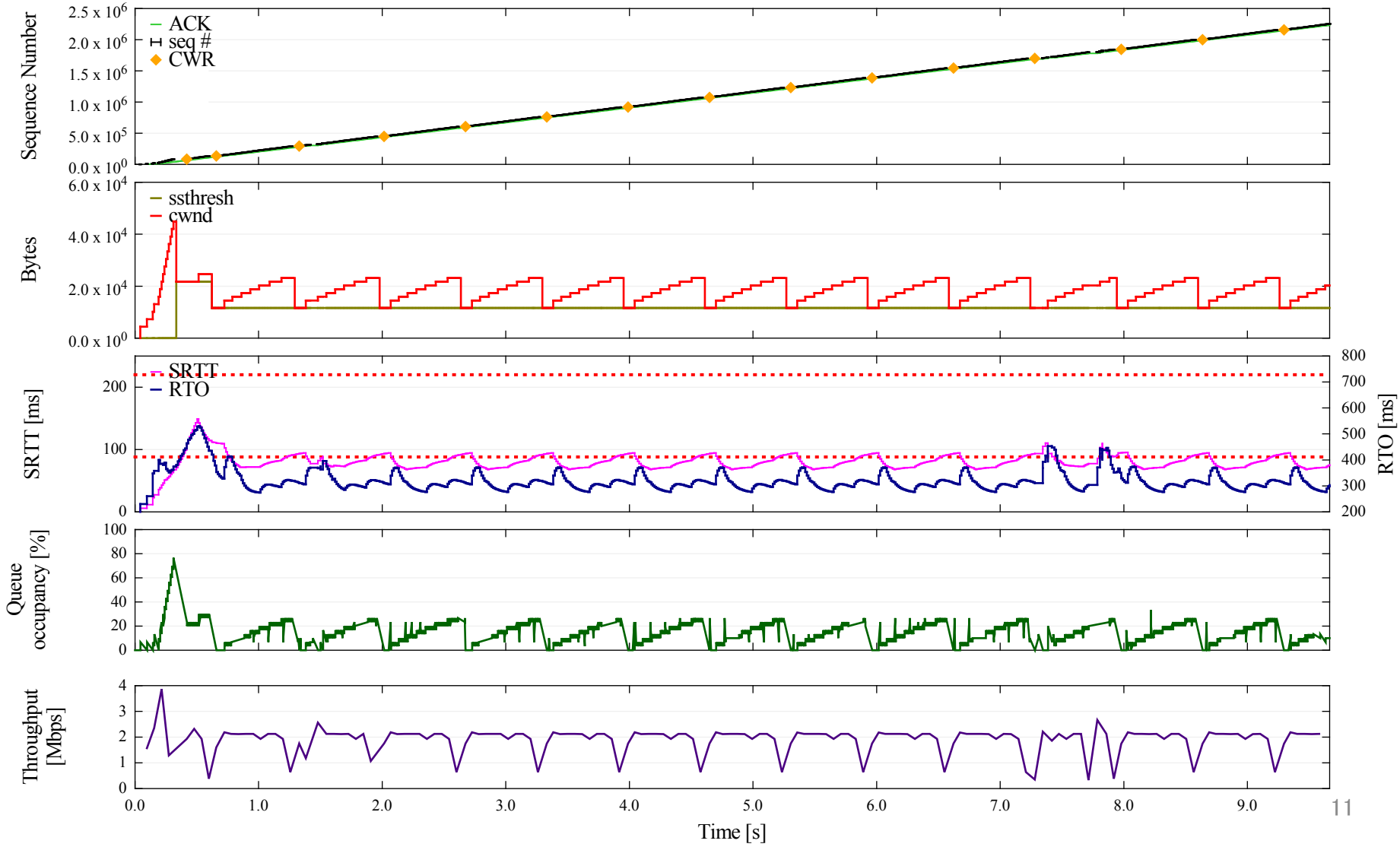
Experiment
- Measure transfer time for *X* KB and SRTT (Smoothed RTT) using flowgrind

BW: 2Mbps, delay: 20ms
qlen: 14, ecn_thresh: 10

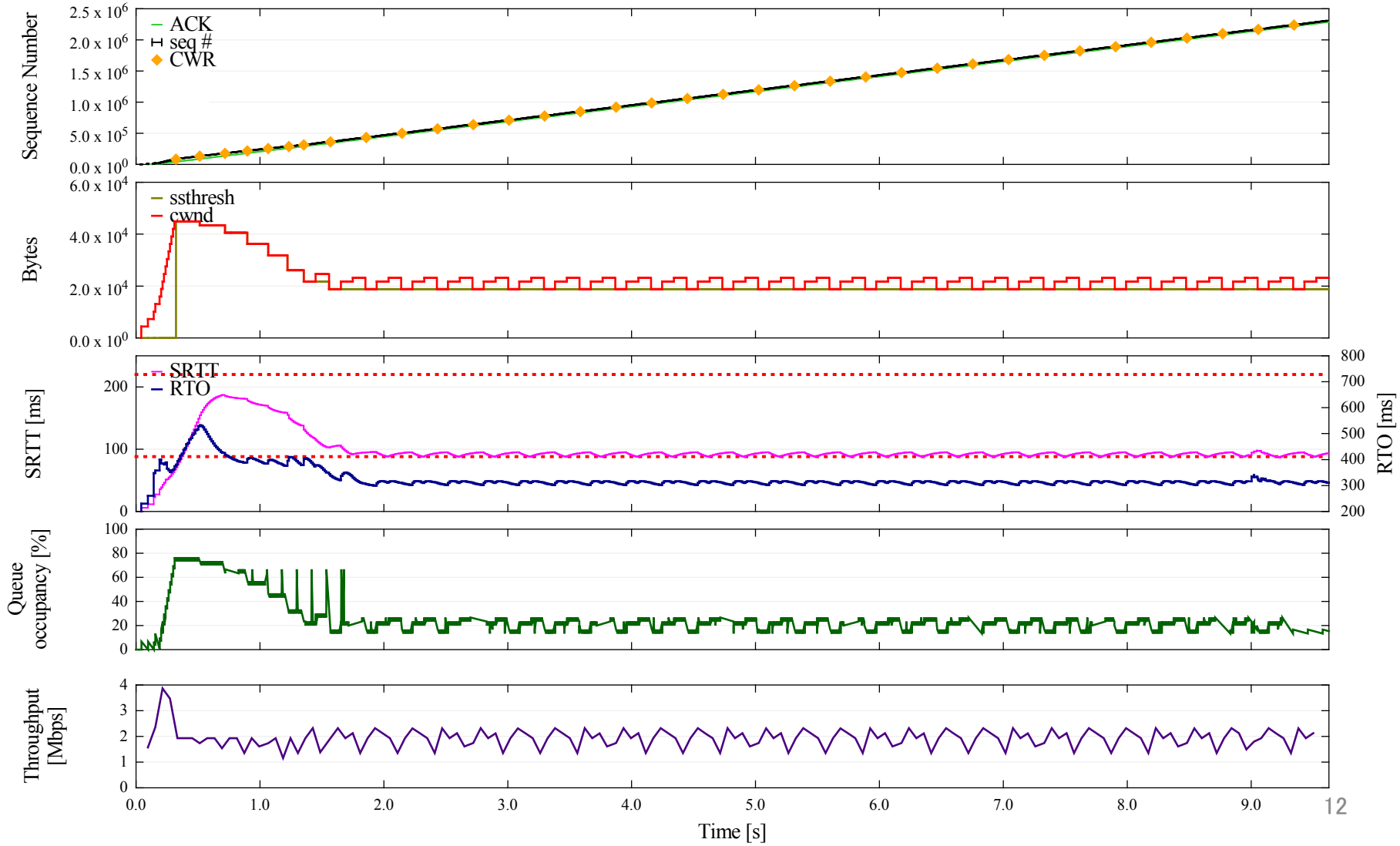




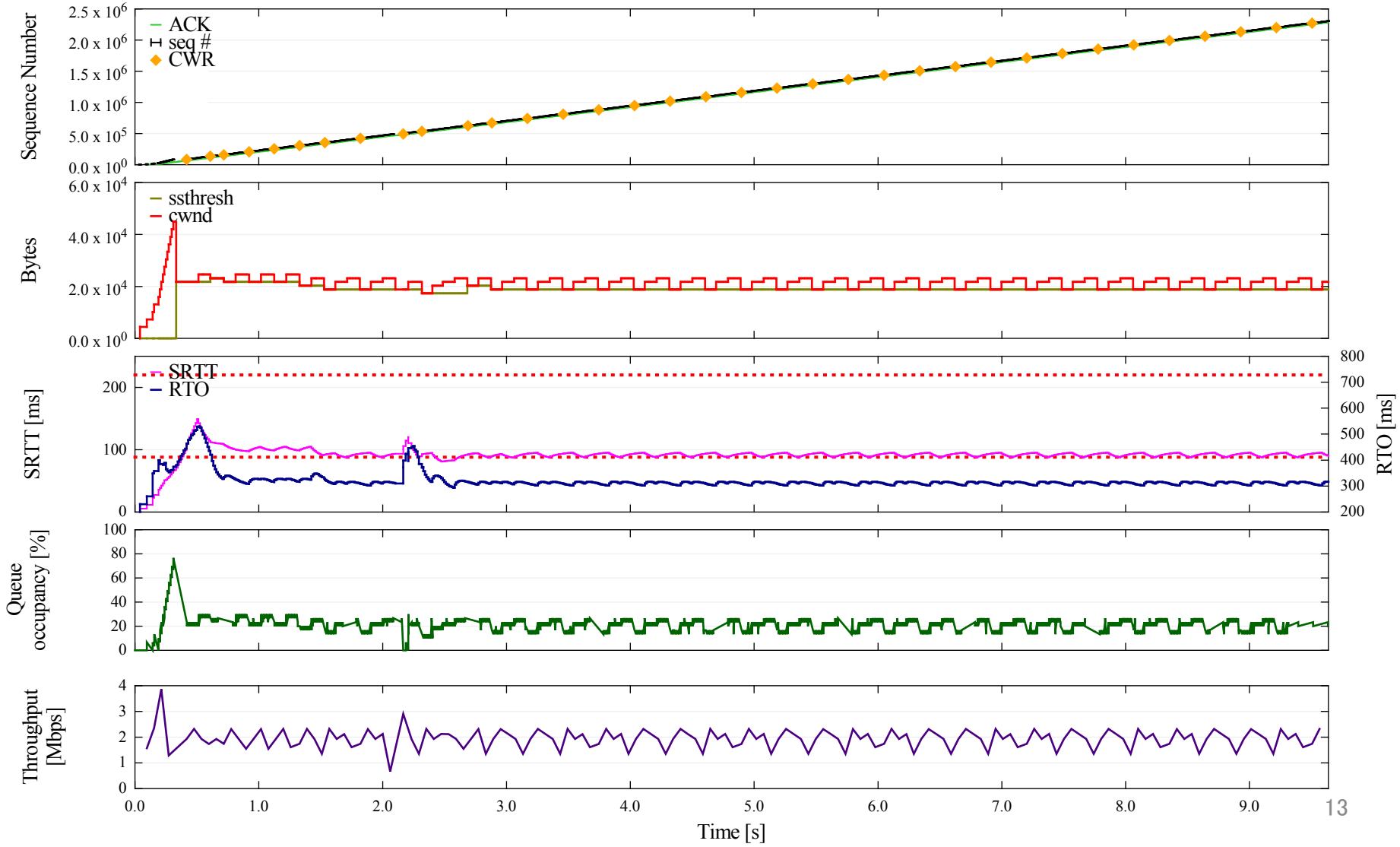
Microscope view of Sack TCP with ECN



Microscope view of Linux DCTCP



Microscope view of FreeBSD DCTCP



TCP(RFC3168) and DCTCP

	Question	TCP (RFC 3168)	DCTCP
sender	When is cwnd reduced?	Reduce cwnd to half as soon as it receives ECE seg.	Reduce cwnd using the proportion of marked bytes
	When CWR is marked?	Only when the host receives CWR	When the host reduces cwnd
Receiver (wo Delayed ACK)	When is ECE marked?	Keep sending segment with ECE until the host receives CWR	Send a segment with ECE only when the packet is marked CE (*)
Receiver (w Delayed ACK)			(*) + Send immediate ack when CE state has changed