Making TCP more Robust against Packet Reordering

Alexander Zimmermann
<alexander.zimmermann@netapp.com>

Lennart Schulte
<lennart.schulte@aalto.fi>

TCPM, IETF-91, Honolulu, HI, USA

November 10, 2014

TCPM's feedback at IETF 89 & 90

- **… to the reordering problem in general**
  - Strong consensus that we should make TCP more robust against reordering
  - Question: what kind of packet reordering do we want to make TCP robust against?

- **… to reordering data & TCP-aNCR measurements**
  - Measuring reordering as a fraction of time
  - Considering loss mixed with reordering
  - TCP-aNCR performance vs. Linux Kernel 3.1x

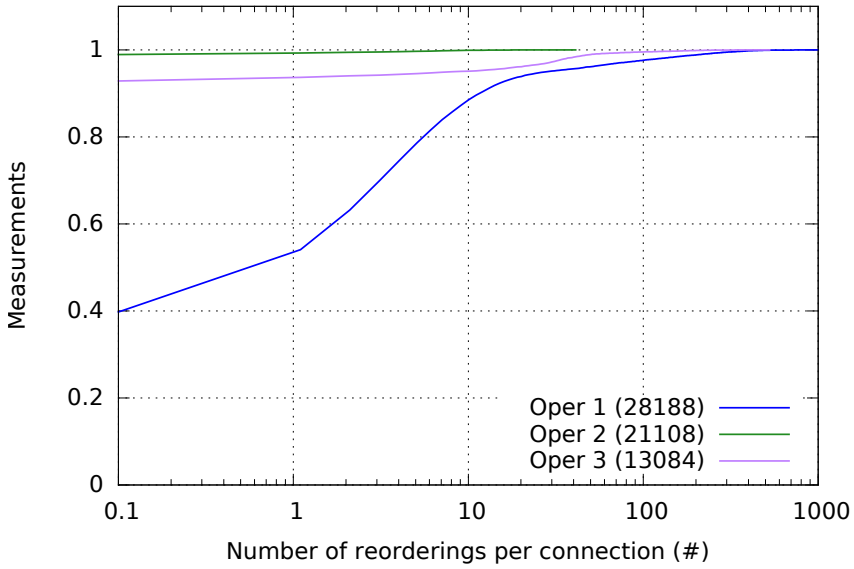
Reordering in Mobile Networks – an Update

- **Netradar.org measurement platform**
  - Provides information about the quality of mobile Internet connections and mobile devices
  - App available for many mobile device OSes

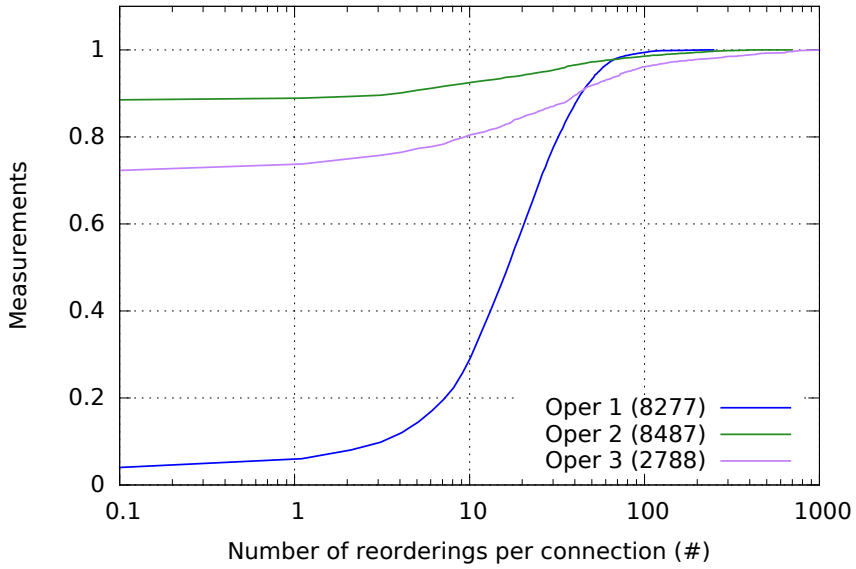
- **Measurement setup**
  - Measurement server @ Aalto University, Finland
  - Performs 10sec bulk TCP transfer
  - PCAP files are analyzed for reordering
    `<https://github.com/lennart-schulte/tcpdump-analyzer>`
  - Implements `draft-zimmermann-tcpm-reordering-detection`

Reordering Events per Connection (3G & 4G)

- **Observations**
  - 3G: w/ Operator 1 60% of connections with reordering
  - 4G: even more connections with reordering
  - Reordering is operator and network depended



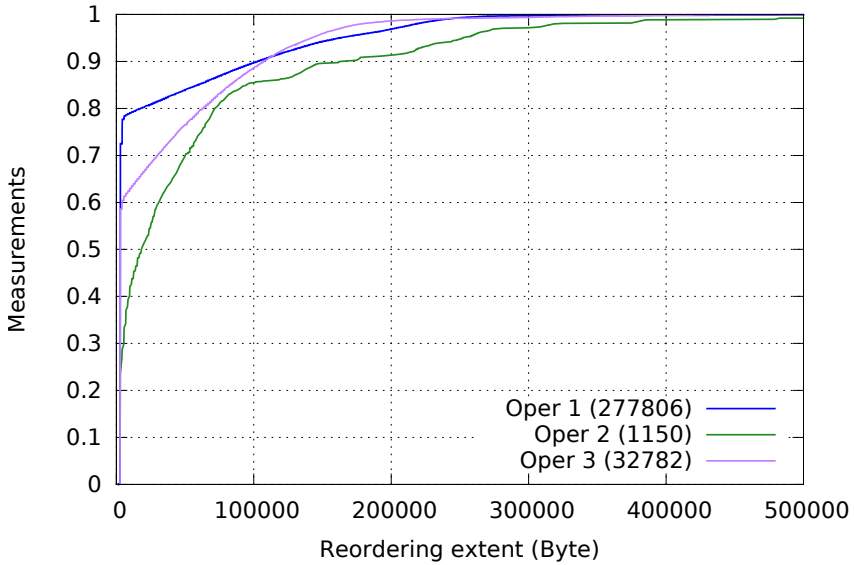
3G Connections

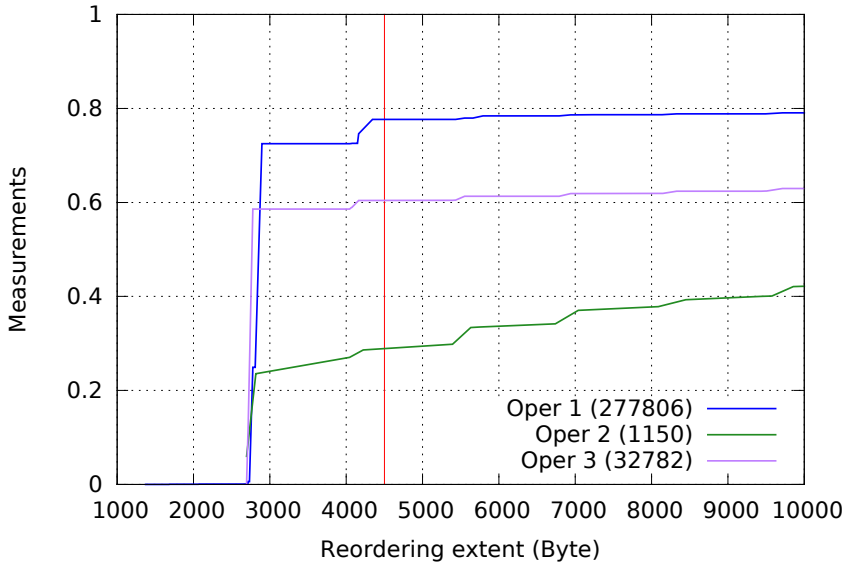


4G Connections

Absolute Reordering Extent (3G)

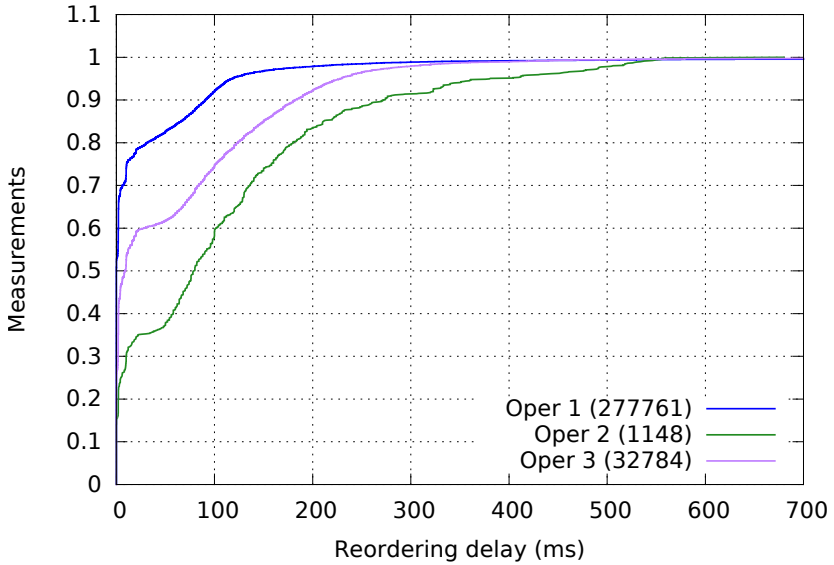
- **Observations**
  - Absolute reordering extent can be huge
  - Absolute reordering extent is operator depended
    - Operator 1: ± 78% of reordering events < 3 segments
    - Operator 2: ± 70% of reordering events > 3 segments



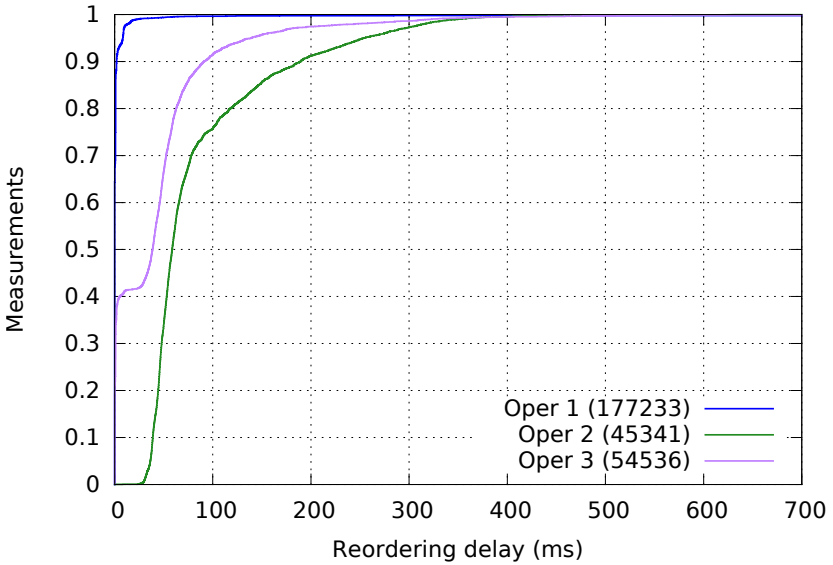


Reordering Delay (3G & 4G)

- **Observations**
  - Two different kinds of packet reordering can be seen
    - Operator 1: higher, fine grained reordering
    - Operator 2: lower, coarse grained reordering
  - Illustrate: *absolute* reordering extent is a func. of CWND



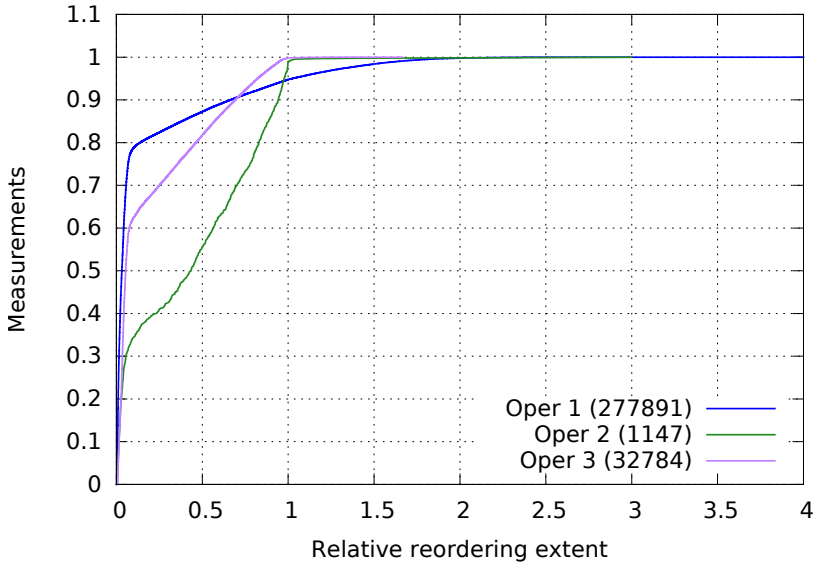
3G Connections



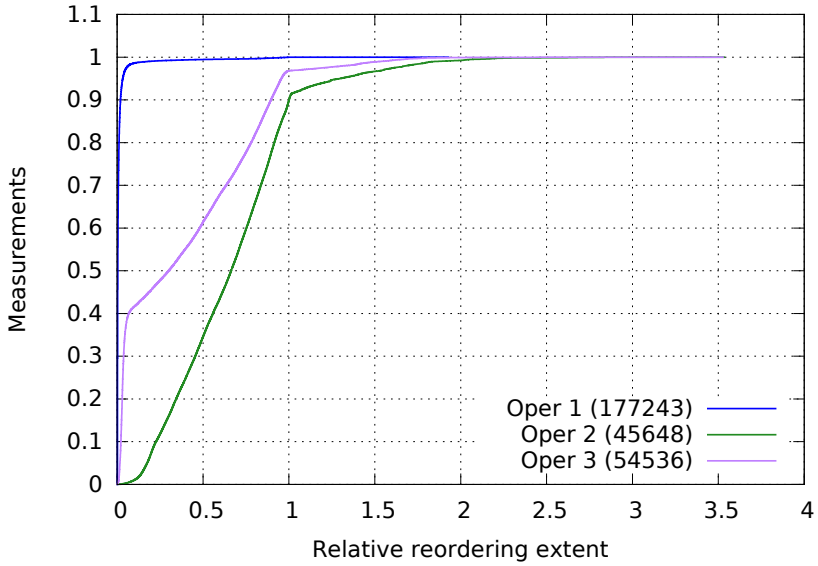
4G Connections

Relative Reordering Extent (3G & 4G)

- **Observations**
  - Relative reordering extent less 1 in > 90% of all cases
  - No strong need to delay a Fast Retransmit more than ± 1 CWND (≈ ± 1 RTT)



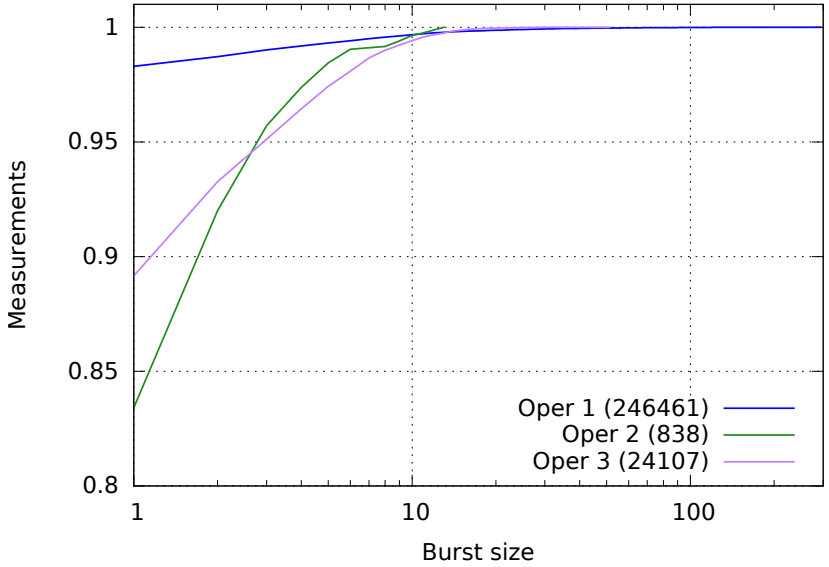
3G Connections

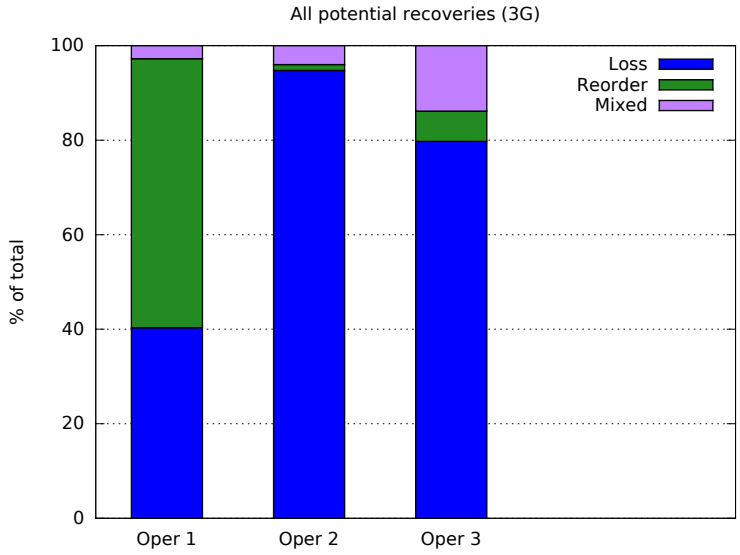


4G Connections

Bursts and Reordering mixed w/ Losses (3G)

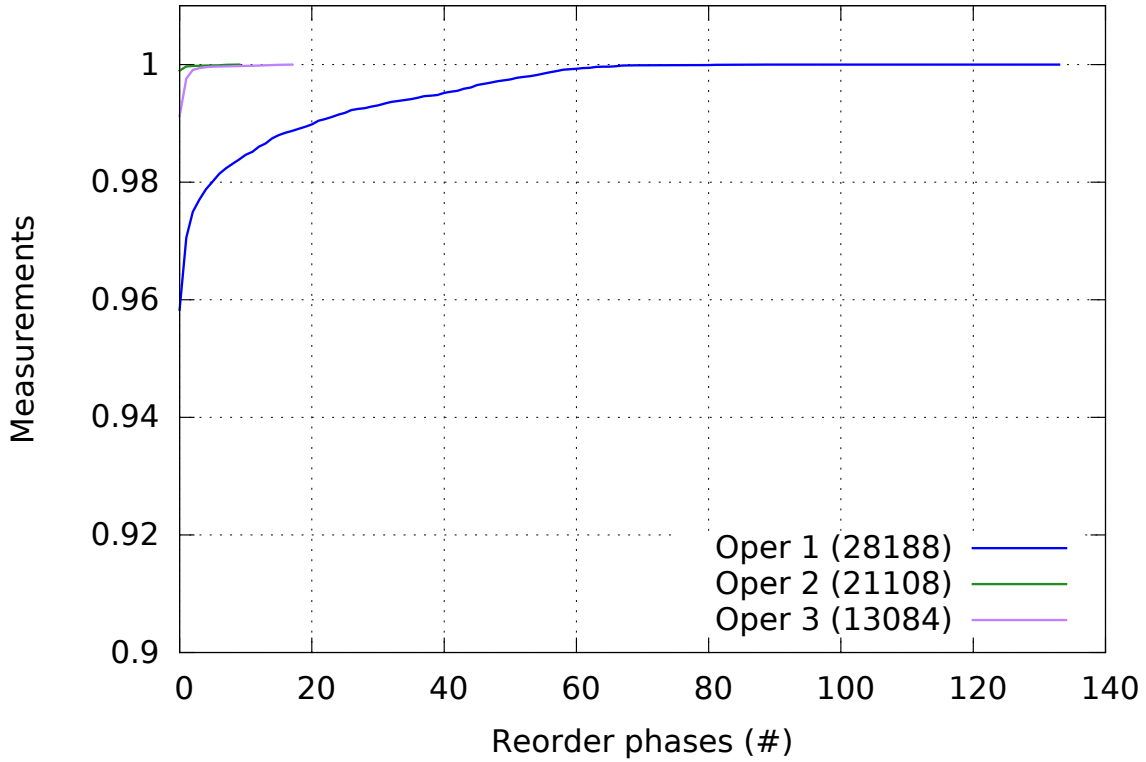
- **Observations**
  - Reordering events can be bursty, or just a lot in one recovery / disorder phase
  - Reordering events can be co-located with losses → no spurious recovery





Number of Avoidable Recoveries (3G)

- **Putting all together…**
  - Operator 1: in 4% of all connections one or more recovery phases could have been avoided
  - For other operators: fewer connections were affected (1% & 0.2%)



Summary & Next Steps

- **We should…**
  - tackle reordering in all dimension → besides prevention strategies, we need a state recovery strategy too
  - detected all kinds of reordering, despite of any reaction → `draft-zimmermann-tcpm-reordering-detection`
  - start to work a reordering detection to gain more inside → adapt `draft-zimmermann-tcpm-reordering-detection`

- **We may…**
  - make TCP *not* robust against any kind of reordering, i.e. reordering > ±1 RTT
  - revive `draft-blanton-tcp-reordering` as overview draft?

Document Status

- **draft-zimmermann-tcpm-reordering-detection-02**
  - Extended algorithm for calculation of reordering extents greater than one RTT (steps C.2, S.5 and D.3)
  - Moved reasoning for relative reordering extent to discussion

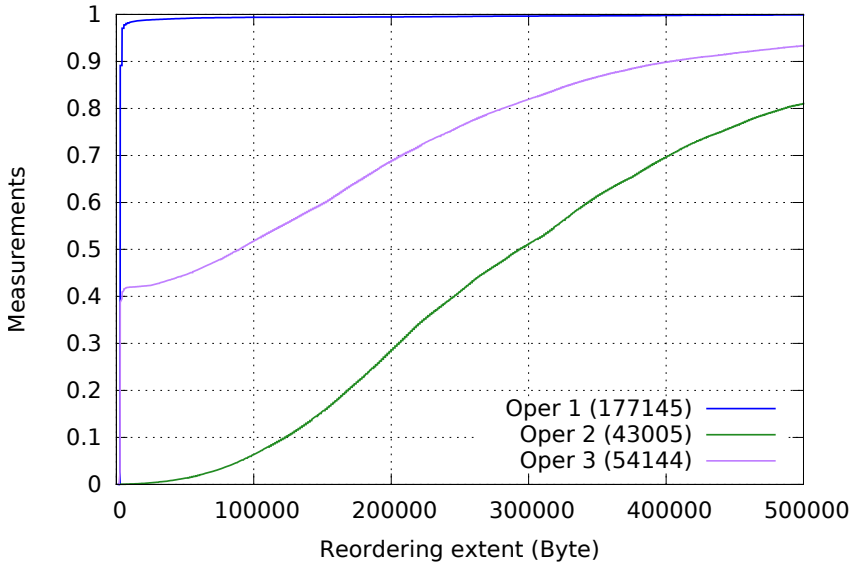
- **draft-zimmermann-tcpm-reordering-reaction-02**
  - Specify interaction between TCP-aNCR and PRR
  - Fix typo in `DupThresh` calculation (steps I.5 and E.9)

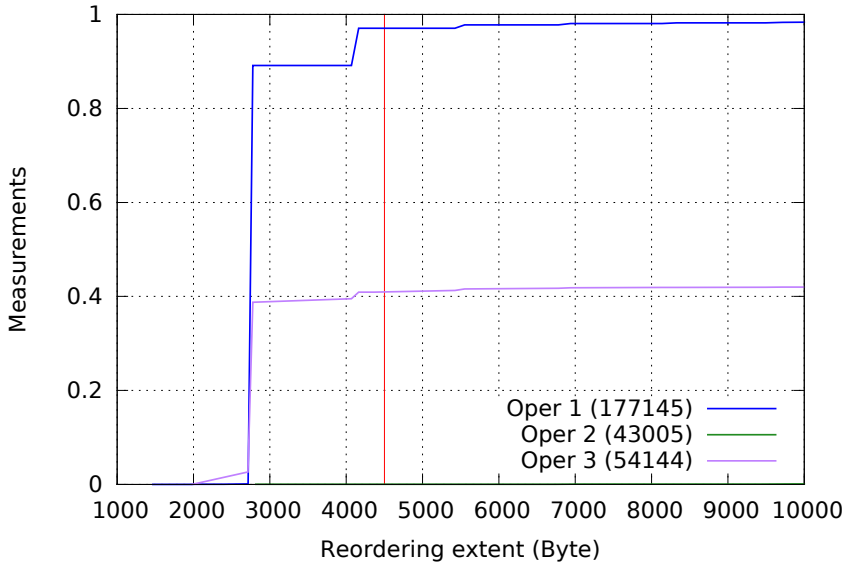
Appendix

Additional 4G measurement results

Absolute Reordering Extent (4G)

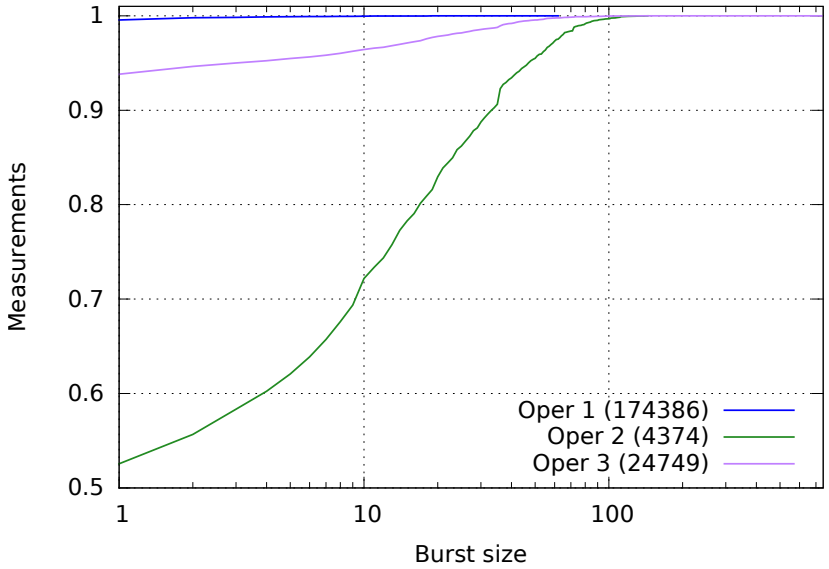
- **Observations**
  - Operator 1: ± 98% of reordering events < 3 segments
  - Operator 2: 100% of reordering events > 3 segments!
  - Same two different kinds of packet reordering as in 3G networks

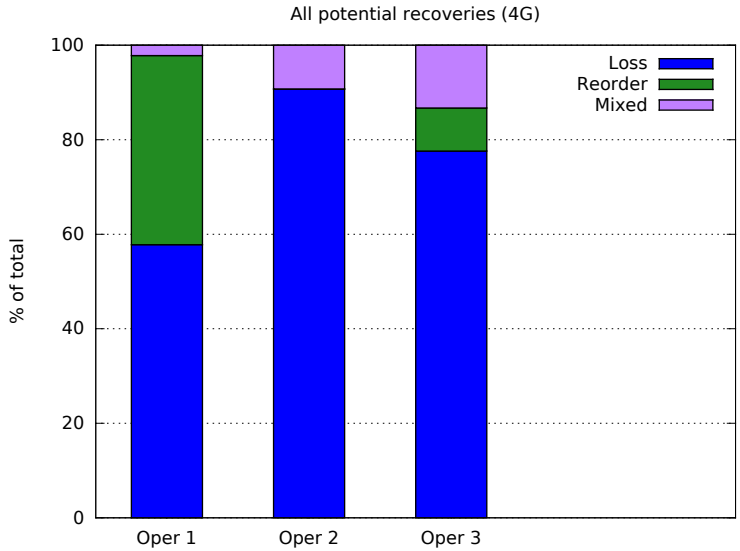




Bursts and Reordering mixed w/ Losses (4G)

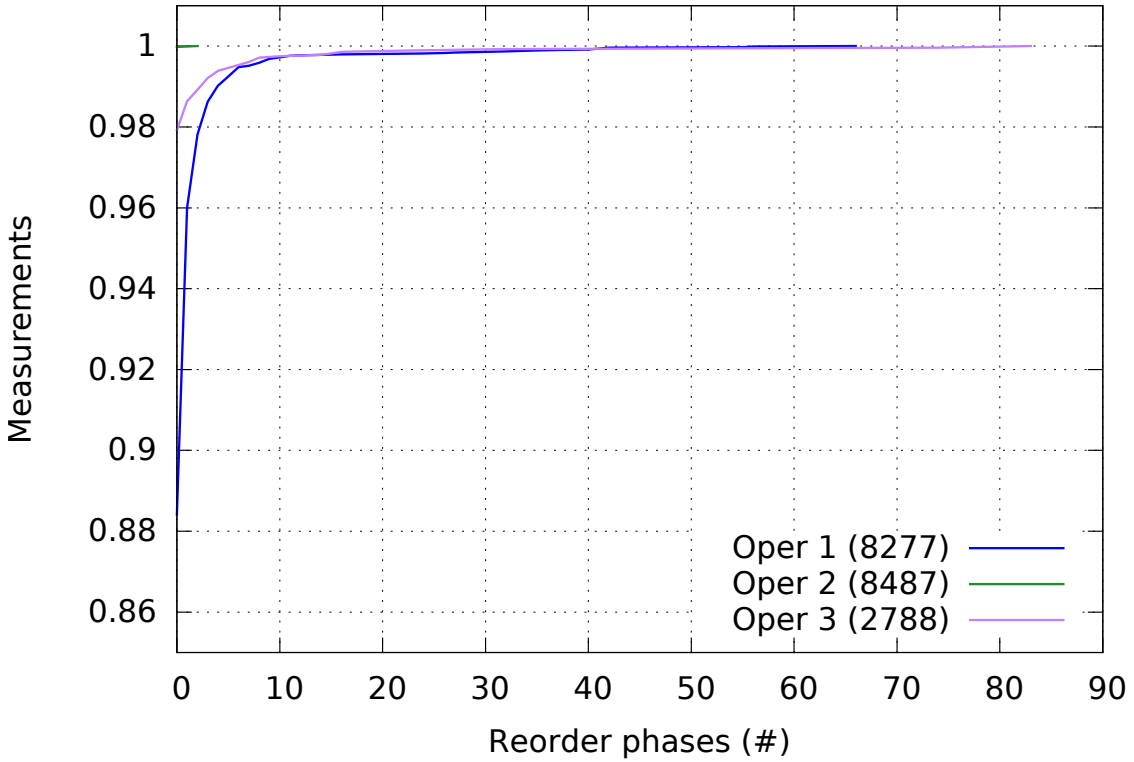
- **Observations**
  - Operator 1: reordering events are more bursty than in the 3G networks
  - Reordering events are as much co-located with losses as in the 3G networks





Number of Avoidable Recoveries (4G)

- **Observations**
  - Worst is Operator 1: in 12% of all connections one or more recovery phases could have been avoided
  - Reflects same behavior as in 3G networks



TCP-aNCR
Measurement Results

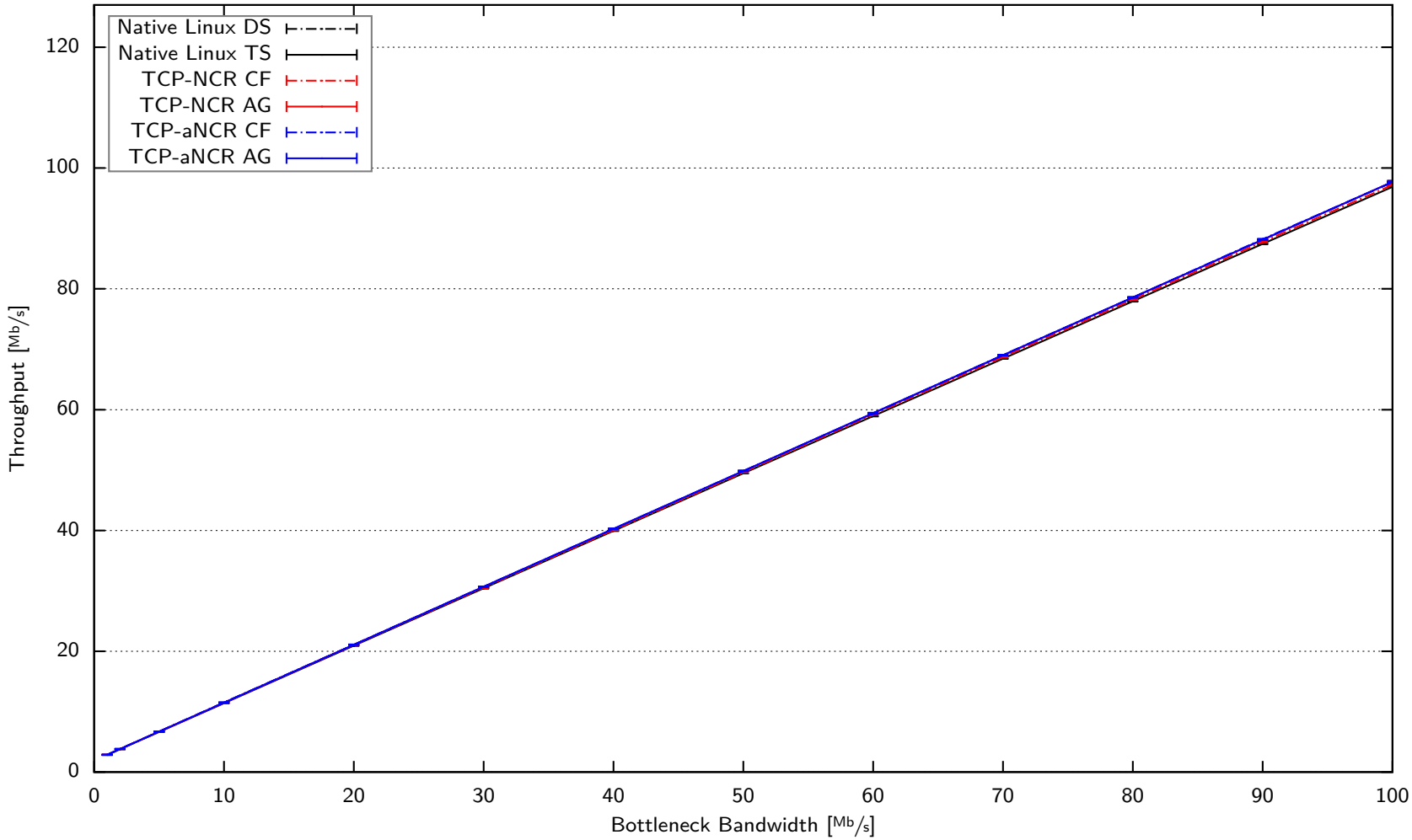
draft-zimmermann-tcpm-reordering-detection-02
draft-zimmermann-tcpm-reordering-reaction-02

Evaluation Methodology

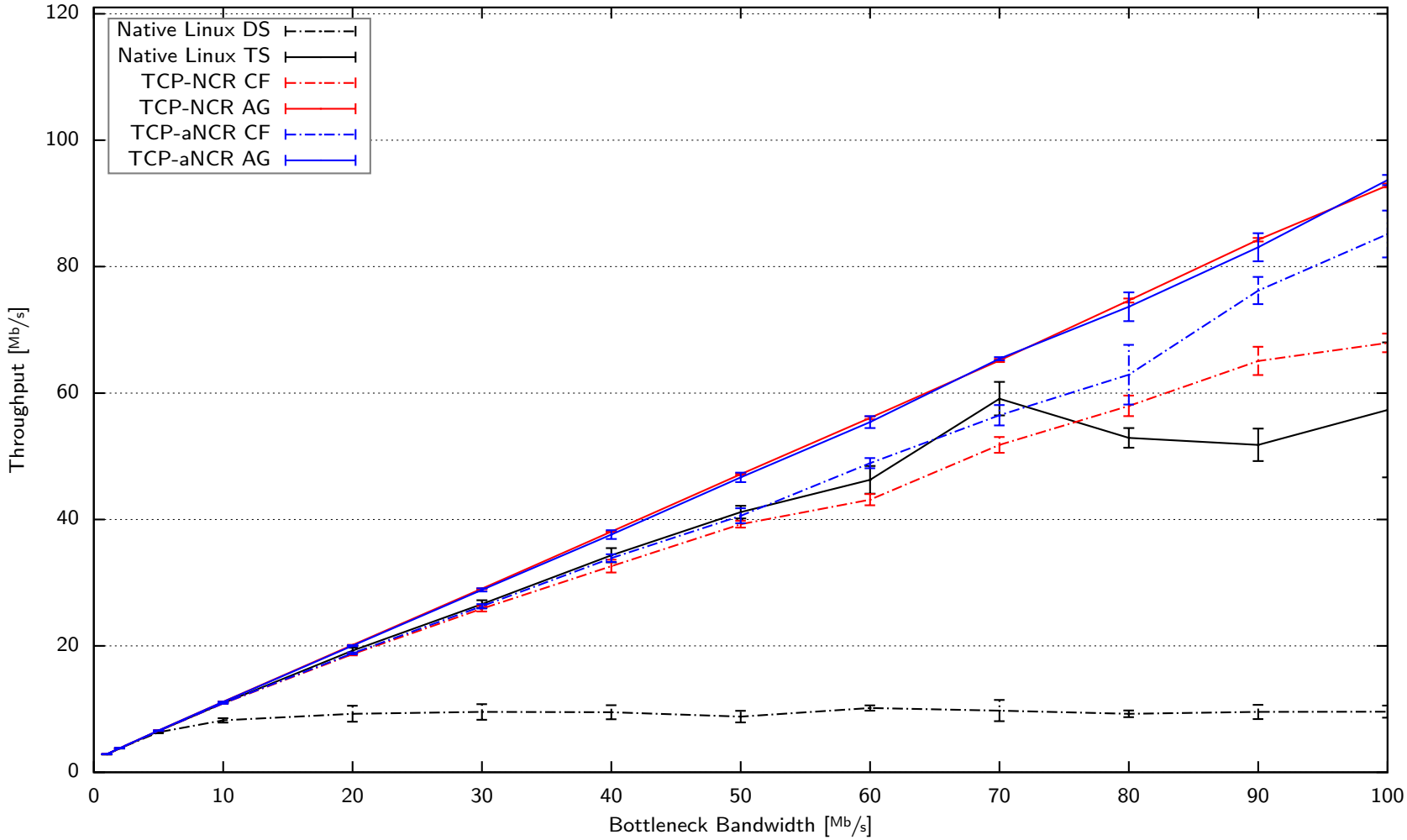
- **Traffic model**
  - One unidirectional long-lived bulk TCP NewReno flow
  - No competing flows; no cross traffic, no short-lived flows
  - Traffic generation: flowgrind <http://www.flowgrind.net>
  - Network emulation: netem

- **Flows under test**
  - Linux 3.16 (w/(o) Timestamps); TCP-(a)NCR (CF & AG)
  - Duration 120s; Repetition: 10x
  - Bottleneck Bandwidth 20 Mbps, RTT 40ms
  - Reordering Rate 2%, Reordering Delay 20ms

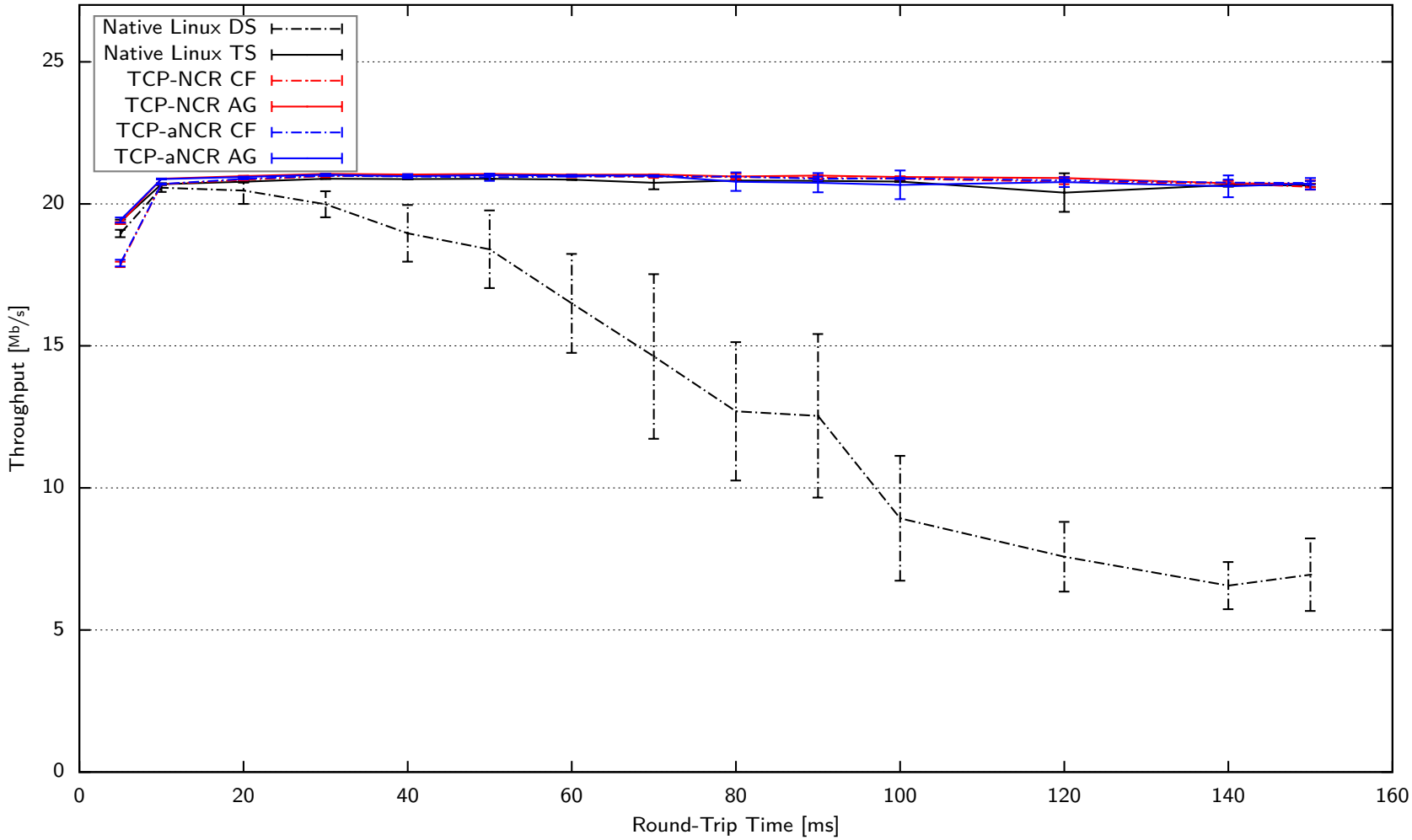
Bandwidth Variation – No Reordering



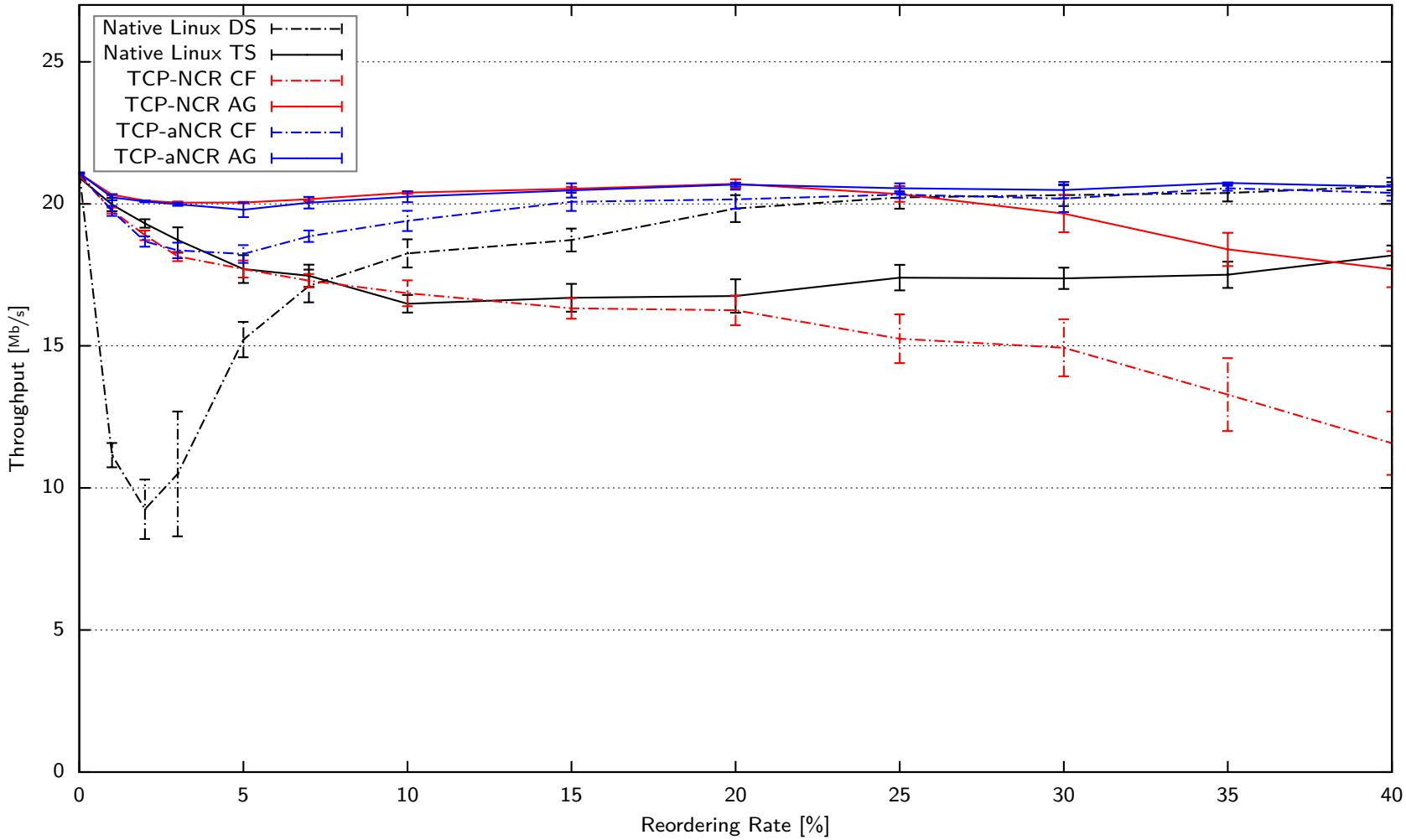
Bandwidth Variation – Reordering



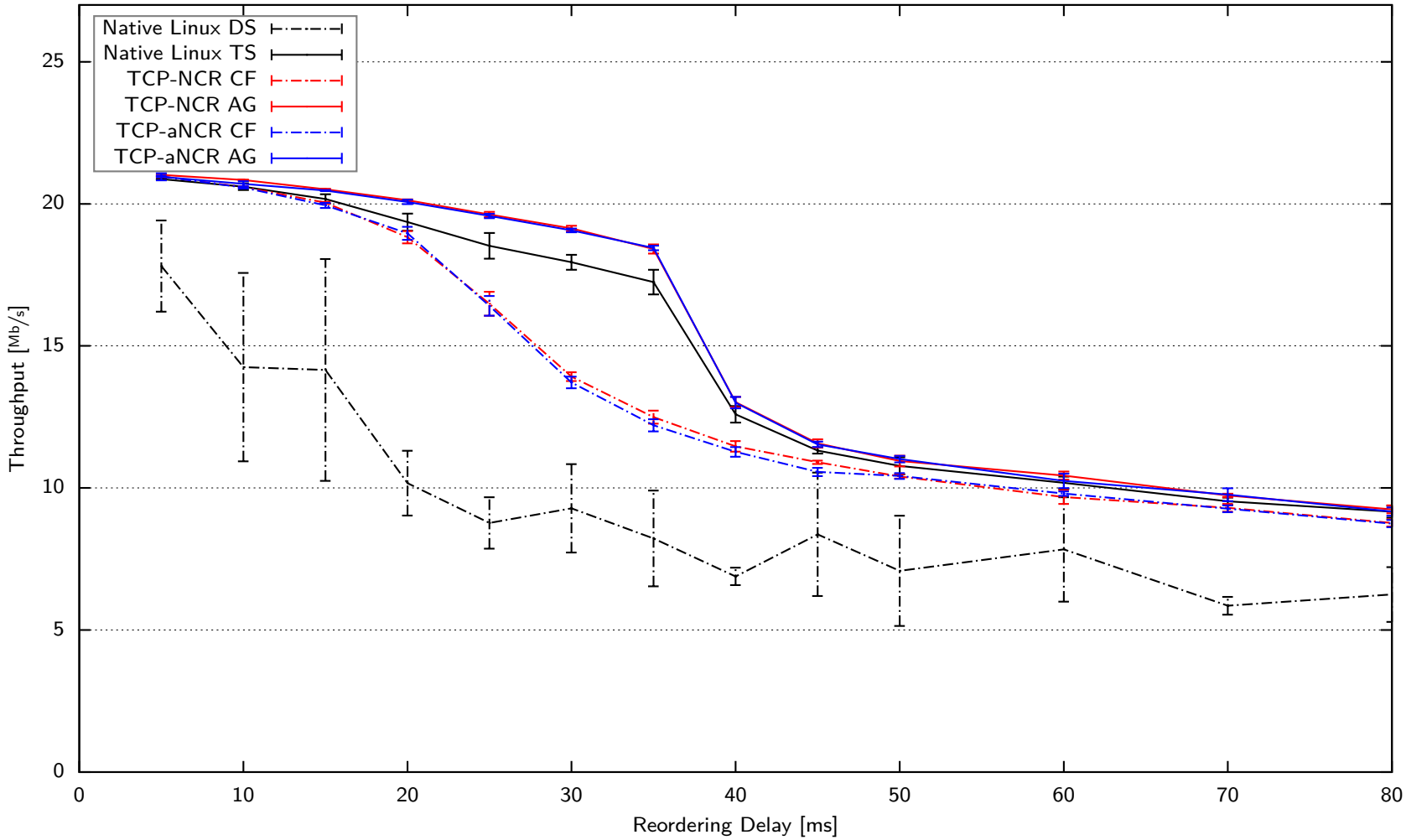
Round-Trip Time Variation



Reordering Rate Variation



Reordering Delay Variation



ACK Loss Variation