TCP-aNCR

draft-zimmermann-tcpm-reordering-detection-00
draft-zimmermann-tcpm-reordering-reaction-00

Alexander Zimmermann `<alexander.zimmermann@netapp.com>`
Lennart Schulte `<lennart.schulte@aalto.fi>`

Brief Summary of Internet Drafts

- **draft-zimmermann-tcpm-reordering-detection**
  - Standardize Linux reordering behavior
    → detection with SACK / DSACK / Timestamps
  - Extend Linux reordering behavior (SACK scoreboard)
    → use reordering events below `SND.UNA`

- **draft-zimmermann-tcpm-reordering-reaction**
  - Maintain RFC 4653 (TCP-NCR): Bug fixing
    → premature leaving Slow-Start, burst protection could fail, performance issues if reordering is persistent
  - Making TCP-NCR adaptively robust to non-congestion events → "must have" for Linux Kernel integration

TCPM's feedback at IETF 89

- **… to the reordering problem in general**
  - Long history fighting reordering → make protocols more adaptive to reordering, it will become an issue
  - People in the IETF go around to other WGs and warn how bad reordering can be for TCP → we need to fix this

- **… to reordering in the wild**
  - Akamai: trouble by detecting conditions under which reordering is a significant problem
  - Google: saw lots of reordering in some African studies

Document & Implementation Status

- **Status of Internet Drafts**
  - Minor editorial changes to the drafts -01
  - No changes to the method
  - Next: reviews needed!

- **Status of TCP-aNCR implementation**
  - Update code base to Linux 3.15
  - On-going: re-running old measurements + latency measurements
  - Next: Clean-up code base and provide patches

Excursus 1: Reordering in Host

- **Reordering is not only in "Far Far Away"…**
  - No need to setup full blown SIGCOMM measurement setups to see reordering at transport level
  - Run `netstat -st` on your Linux box…

- **Quick example**
  - Xeon 2x E5-2650 (8 cores, 2.0 GHz),128 GB RAM
  - 40 Gb/s Ethernet B2B, Vanilla Linux kernel 3.14, Debian 7.0

```
alexandz@phobos02:~$ netstat —st
TcpExt:
    ...
    2257 times recovered from packet loss by selective acknowledgements
    Detected reordering 12 times using FACK
    Detected reordering 4738 times using SACK
    Detected reordering 269 times using time stamp
    64 congestion windows fully recovered without slow start
    360 congestion windows partially recovered using Hoe heuristic
    532 congestion windows recovered without slow start by DSACK
<truncated>
```

Excursus 2: Reordering in Mobile Networks

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

- **Setup**
  - Measurement server in Aalto University network, Finland
  - Performs 10 second bulk TCP transfer
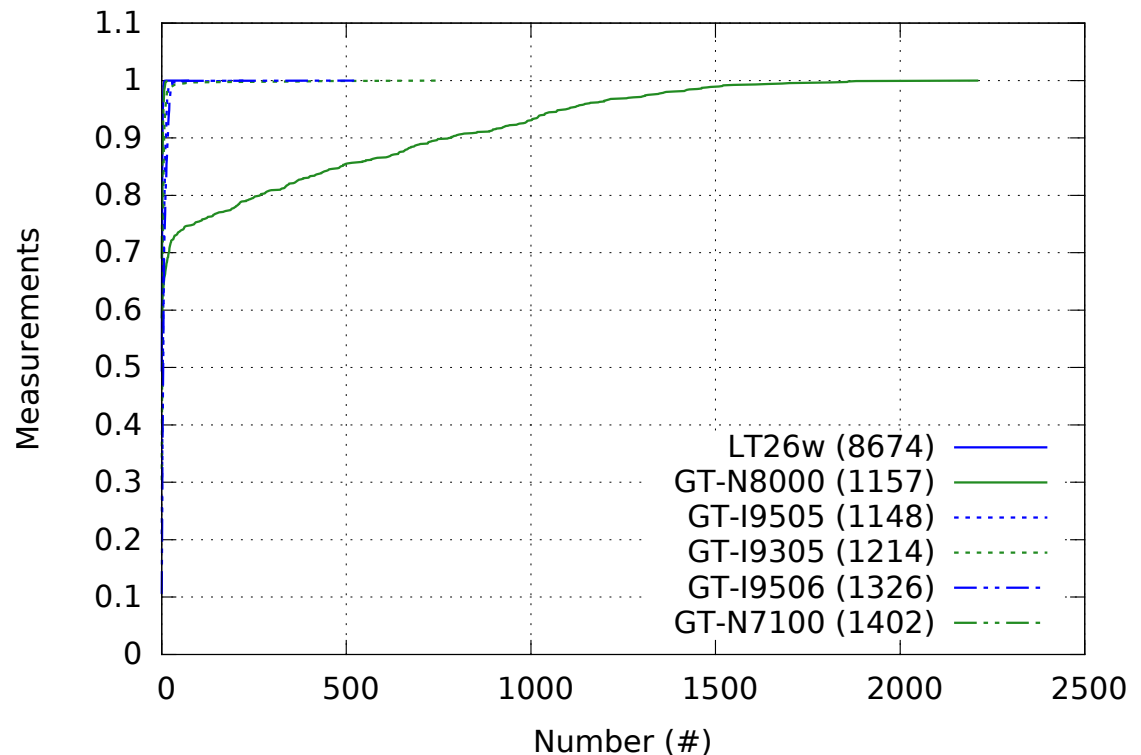 - Tcpdumps are analyzed for reordering

Reordering in Mobile Networks: Devices

- **Observation**
  - One phone model has reordering more often
  - But reordering extents are small
  - Number of analyzed connections in brackets

- **Bug?**
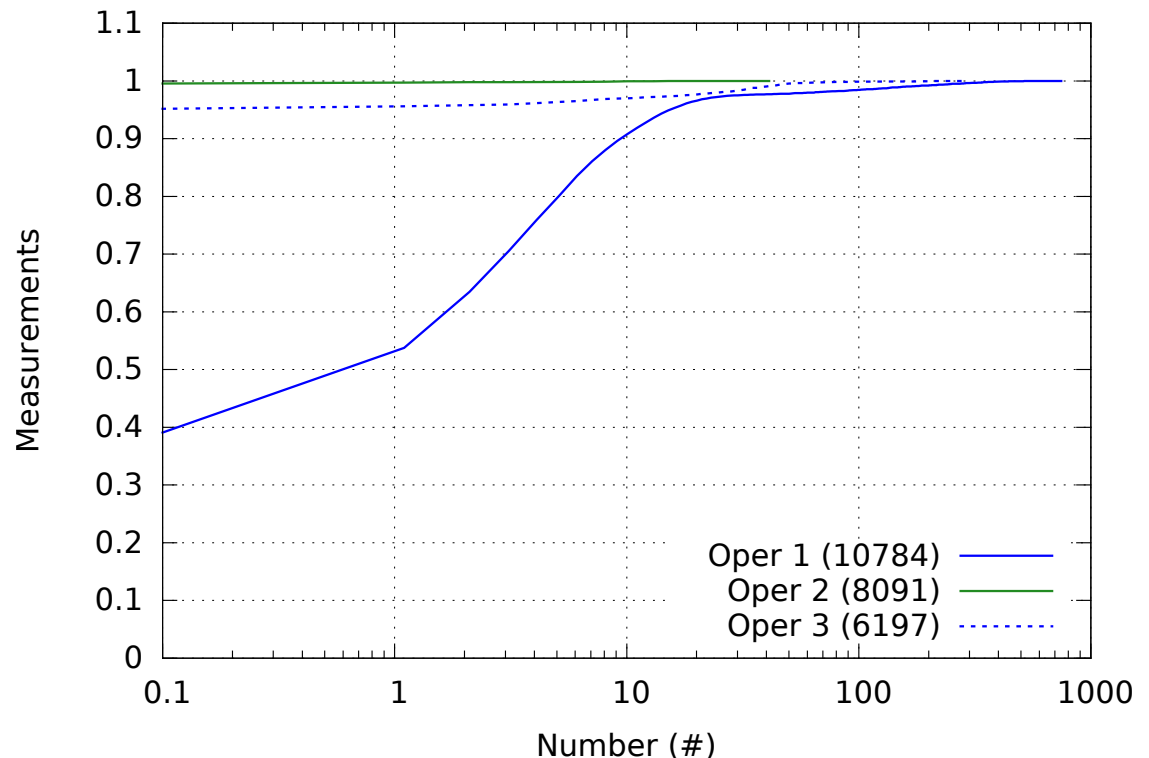 - GT-N8000 is removed from dataset of next study

Reordering in Mobile Networks: Operators

- **Backbone connectivity**
  - Finnish operator networks and Netradar server are directly connected to IXP

- **Observation**
  - Reordering is operator depended
  - Number of analyzed connections in brackets
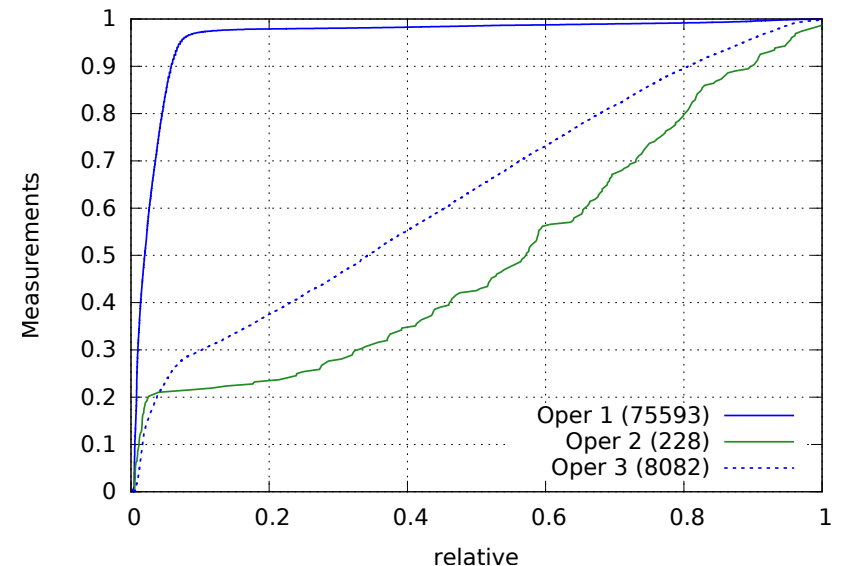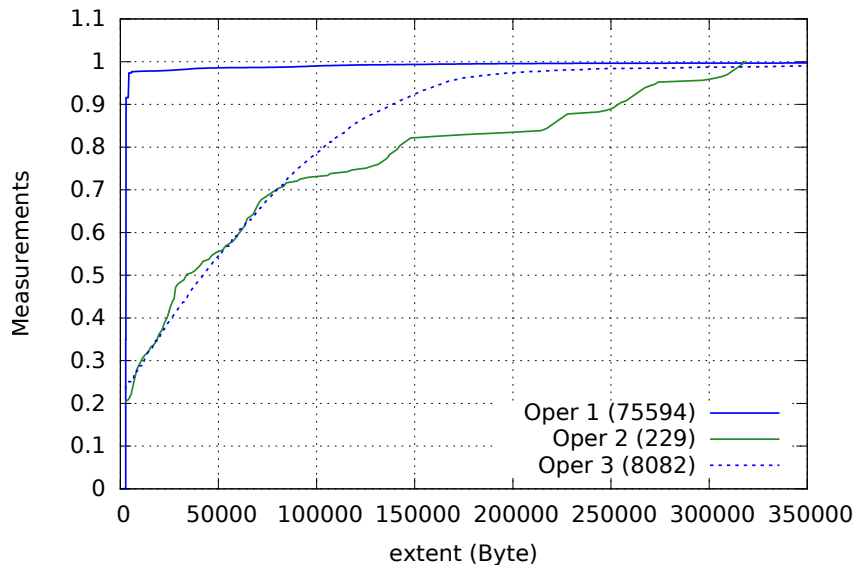
Reordering in Mobile Networks: Operators

- **Observation**
  - OP1: many reordering, low extents
  - OP2: very few reordering, high extents
  - OP3: moderate amount of reordering, high extents
  - Number of reordering events in brackets

What's next?

- **Why we should adopt the drafts as WG item?**
  - Clear consensus that we should take care about reordering
  - Both documents are typical TCPM documents
    - **draft-zimmermann-tcpm-reordering-detection**: basically document & standardize Linux reordering behavior
    - **draft-zimmermann-tcpm-reordering-reaction:** basically maintain RFC 4653 (TCP-NCR) & fix bugs
  - Running code available

- **Possible next steps for TCPM WG**
  - Adoption as WG item
  - Feedbacks and reviews