Note: More detailed speaker's notes for these slides are available at https://www.snellman.net/blog/archive/2015-08-25-tcp-optimization-in-mobile-networks/

Mobile TCP Optimization

Lessons Learned in Production

Juho Snellman
Teclo Networks

Introduction

- Background
- TCP optimization
- Lessons learned

Background

- Teclo Networks is a 5 yo startup based in Zurich
- TCP optimization for mobile networks
- About 20 commercial deployments
  - From MVNOs to major operators and operator groups
  - Largest deployment for >100Gbps of peak traffic

Implementation 1/2

- Off the shelf hardware (Xeons, 825xx NICs)
- Optical bypass for reliability
- Up to 20Gbps of optimization, 10 million connections for a 2U node
- Bump in the wire integration, usually on the Gi link of the GGSN

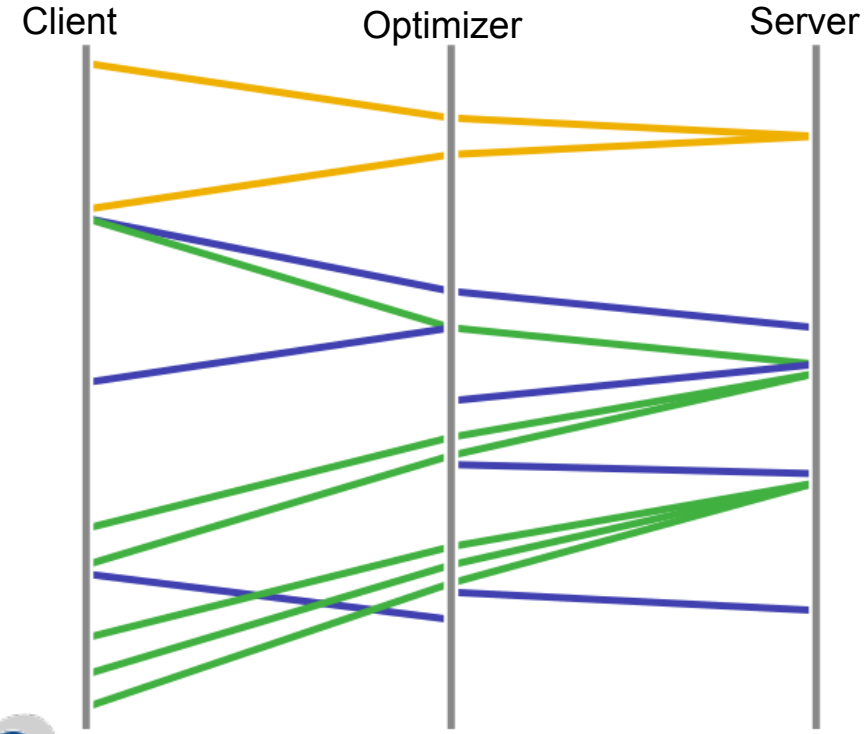
teclo

Implementation 2/2

- Completely custom user space TCP/IP stack
- User space NIC drivers
  - Completely zero copy, even for traffic that is buffered for arbitrary periods of time
- Having no kernel components is huge

TCP Optimization

An optimized connection



- Observe 3WHS, don’t terminate
- If SYN and SYNACK are ok, optimize
- Start ACKing data, take over delivery responsibility

teclo

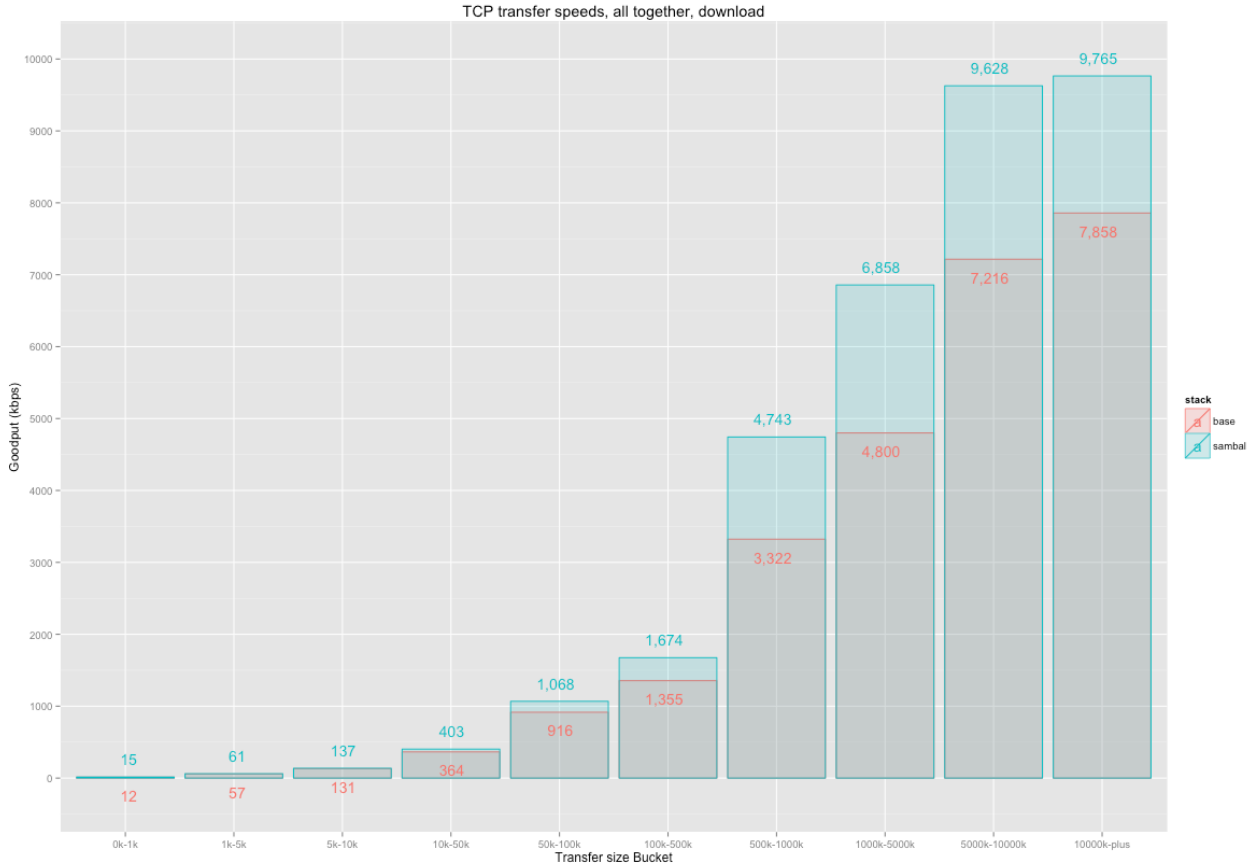
Transparency

- Can stop optimizing a connection at any time
- Deals with asymmetric routes
- Friendly to new TCP options
- Protocols that pretend to be TCP but aren’t

Simple optimizations

- Latency splitting
  - Slow start
  - Steady state limited by receive window
- Retransmit from closer to source packet loss
- No fancy congestion control, but some heuristics for non-congestion packet loss
- Tail probing instead of retransmit timeouts

Speedups

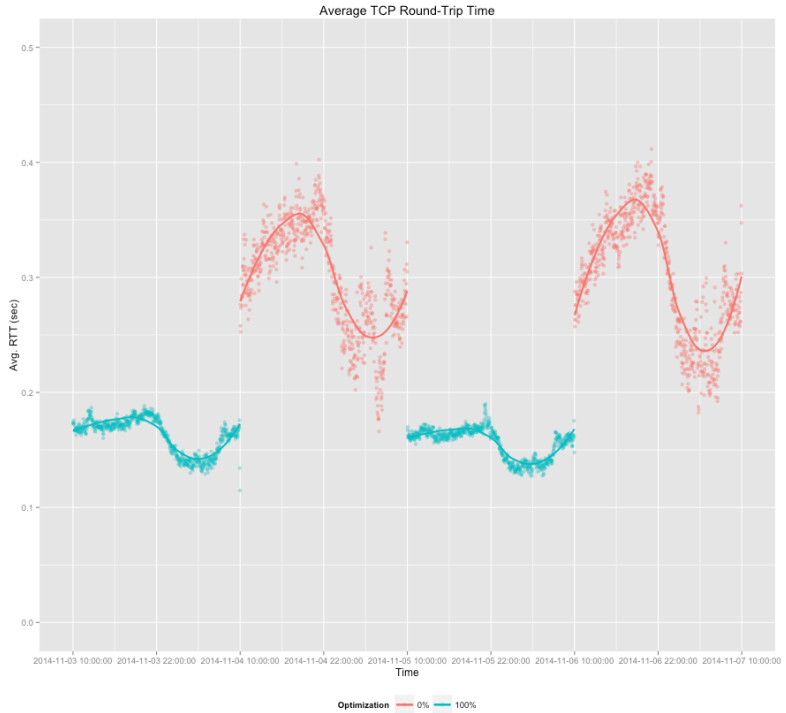


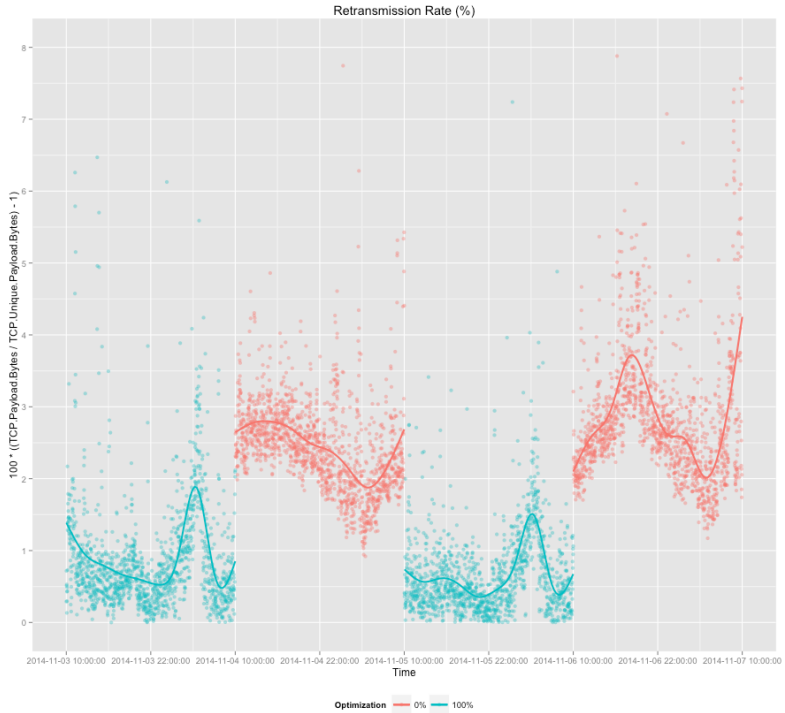
teclo

Buffer management

- Mitigate buffer bloat
- In mobile networks queues are per-user
  - Treat all TCP flows for a single mobile subscriber as a unit
  - Determine optimal level of in flight data to keep RTTs stable
  - Fair scheduling between flows
- Independent of per-flow congestion control

Effect on RTTs and packet loss





teclo

Burst control

- Easy to generate burst of outgoing packets:
  - ACK bunching, ACKs lost, full receive window SACKed
- Even small bursts can cause full buffers + packet loss on 1G/10G boundaries
- Don’t send 200kB at once, instead pace the packets and send 20kB every 1 ms
- Reduced loss rate from >1% to <0.2%

Things we learned along the way

Don’t rely on hardware features

- Every time we depend on fancy hardware features we end up regretting it
- Always need pure software fallback
- Encapsulation most common problem
- Checksum offload
- Multiple RX queues + flow director

Two mobile networks never equal

- Constantly see new network pathologies, new types of integration
- New features often specific to only a few customers
- Automated testing is absolutely crucial

Reordering

- Mobile should have no reordering
- In some networks small packets can be massively reordered ahead of large ones
  - Seen reordering of over 30 segments / 50ms
- Particularly bad if proxy repacketization causes small packets to be generated regularly

Strange packet loss patterns

- One network regularly losing some or all packets at start of connection
  - About the worst thing you can do to TCP
  - Only in one region, different radio vendor from other regions
  - Probably somehow related to 3G state machine transitioning from low power to high power

Bad or conflicting middleboxes

- Lots of middleboxes from multiple vendors, with complex interactions
- MTU clamping
- Proxies
  - Bad tcp settings, repacketization, zero window problems
- PCEF / traffic shaping vs. optimization

O&M is a lot of work

- Can’t sell just the traffic handling, need support infrastructure
  - CLI
  - Web UI
  - Historical counter database
  - SNMP, RADIUS, TACACS, etc
  - Analytics

Thanks

Questions?

Email: juho@teclo.net
Twitter: @juhosnellman

teclo