QoS at Meta

… and one way it went wrong

Ashley Hatch
Network Engineer

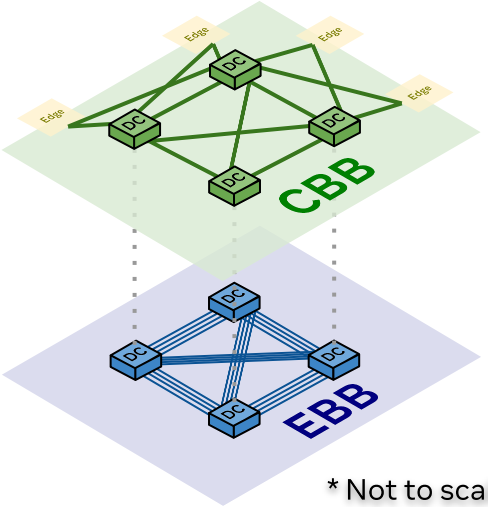
Meta

Agenda

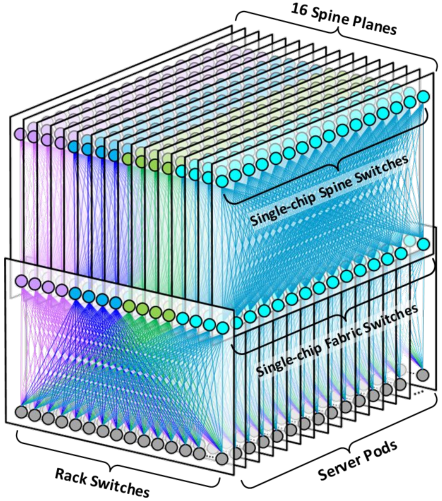
01 Overview of Meta’s Networks

Meta's Networks

- Datacenter - Large multi-dimensional fabrics
- Express Backbone (EBB) - Intercontinental in-house SDN for DC-DC traffic
- Classic Backbone (CBB) - Intercontinental RSVP-TE network for DC to Edge peering
- Edge - Globally distributed CDN edge networks in colocation facilities



* Not to scale



02 Our QoS Journey and Policy

The Need For QoS - Managed Unfairness

- Where possible, overbuild for failures and down capacity
- Where unavoidable… QoS!
  - Demands can grow faster than we can build
  - Failures (Weather, Hardware, Power, Sprinklers, Trees, Rifles, Squirrels…)
  - Upgrades and migrations

“*All bits are equal, but some are more equal than others.*” - Unknown

The Path to Unified QoS

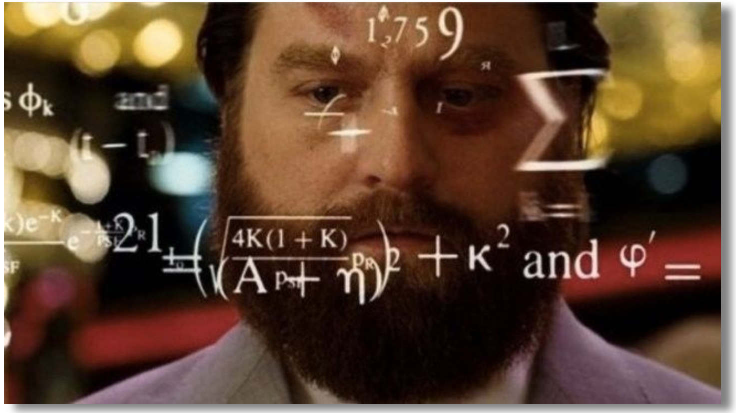
- Self assigned prioritization
- Important services request special treatment
- Different chipsets and platforms
- Queue configurations and buffer allocations
- Pain = Prioritization

QoS for Everyone!

- Different topologies
- Different platforms/chipsets
- Different hardware generations
- Different traffic demands
- Different traffic engineering
- Different product groups

QoS - KISS

- Stacked strict priority
  - Strict-priority Waterfall in backbones
  - WRR in some layers, heavily biased
- Rare buffer tuning
- End-to-end treatment normalized hop-by-hop
- Host marking using DSCP/TC
- Centralized service database with enforcement



Olympic QoS - One Plan To Rule Them All

Five traffic classes mapped to hardware queues

- Network Control - IGPs, BGP
- Platinum - Platform control plane
- Gold - Critical product services
- Silver - Default services
- Bronze - Cost optimized bulk services

Olympic QoS - Floodgate

- Disaster recovery control
- Enforce traffic quota per service
- Flow by flow accounting
- Berkeley Packet Filters (BPF)
    - Flow-by-flow remarking

03 Results and Lessons Learned

Olympic QoS - Lessons Over the Years

- Someone has to say NO!
- QoS interacts strongly with Traffic Engineering
  - Fill links lightly with critical traffic
  - Pack links with less important traffic
- Strict queuing is easy to reason and explain
- Avoid endless reoptimization cycles

Olympic QoS - Success!

- We, almost, never drop Platinum traffic
  - Buffers to down interfaces
  - Packets on cut fibers
  - Old labels during convergence
- We don't spin on queues or tuning
- Occasionally, we have a 5 year old bug

04 A Story of How It Once Failed

Hunting the Elusive Out-of-Order

Production Engineer complains about OOO packets in Platinum

- Impacting critical global database synchronization
- Out-of-order packets?
  - Pathing changes?
    - BGP
    - Traffic Engineering
    - Hashing failures
- Loss.. in Platinum???

Tooling and signals

Service/Host Metrics

Database lag

Host ooo counters

Source and destination sites

Dashboards/Data

EBB dashboards

NetNORAD

Server packet sampling

Device telemetry data

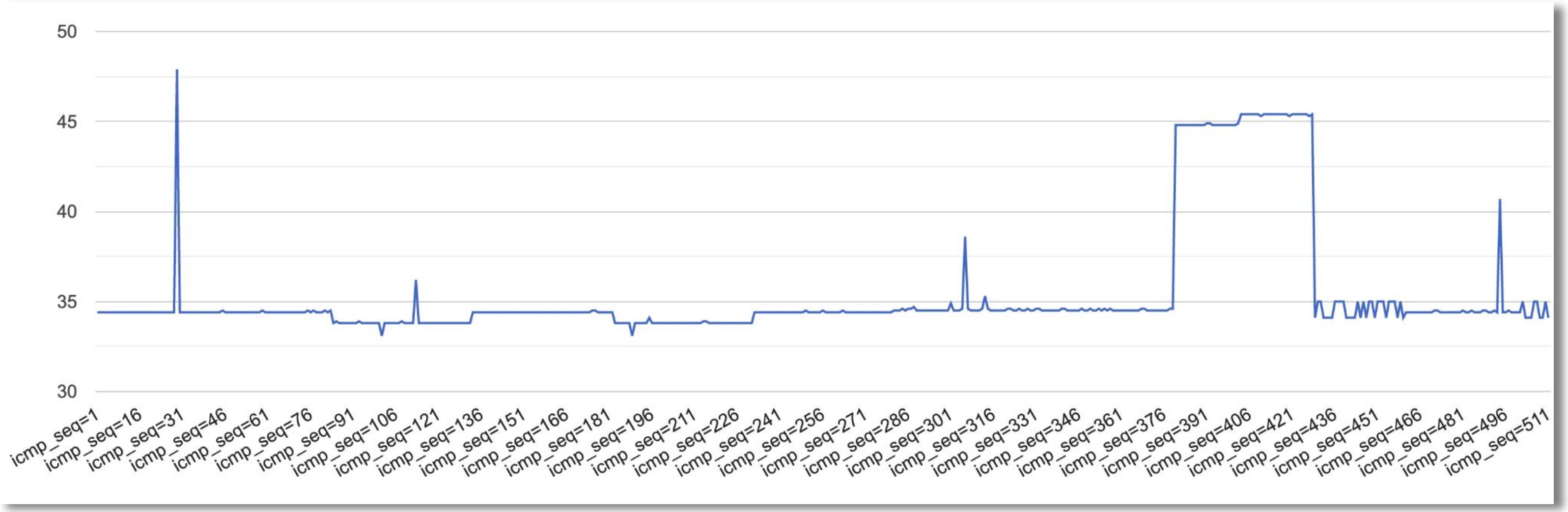
Custom Tools

tcpdump + Wireshark

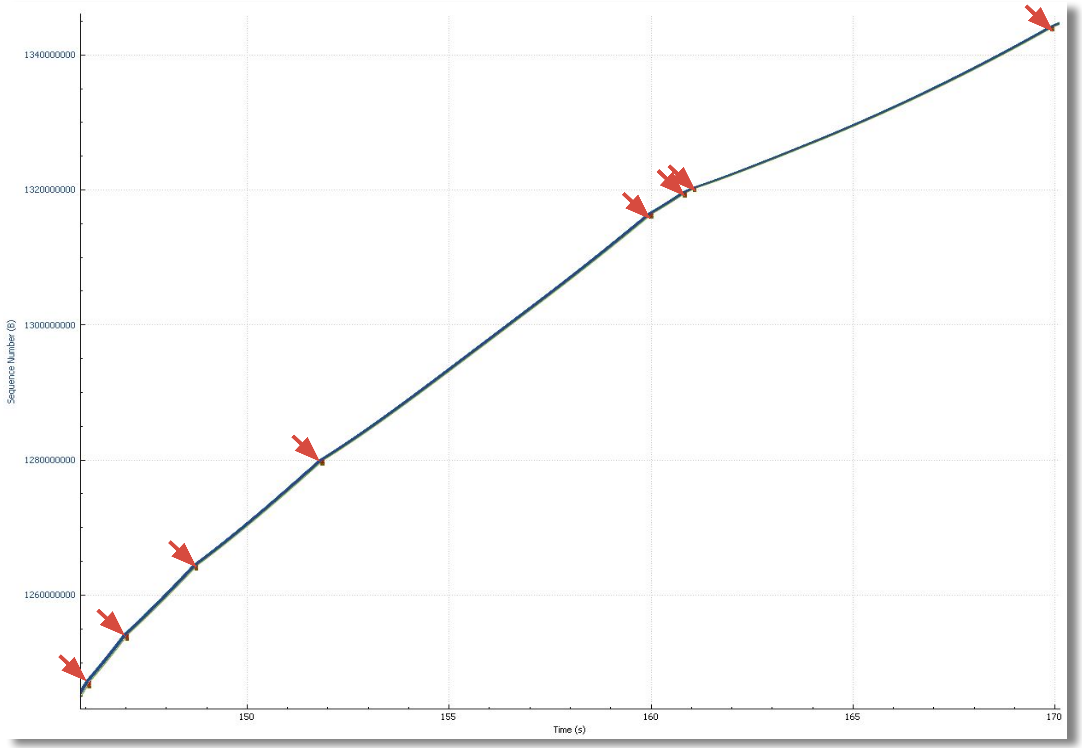
iPerf triangulation

Targeted iPerf + ACL logging

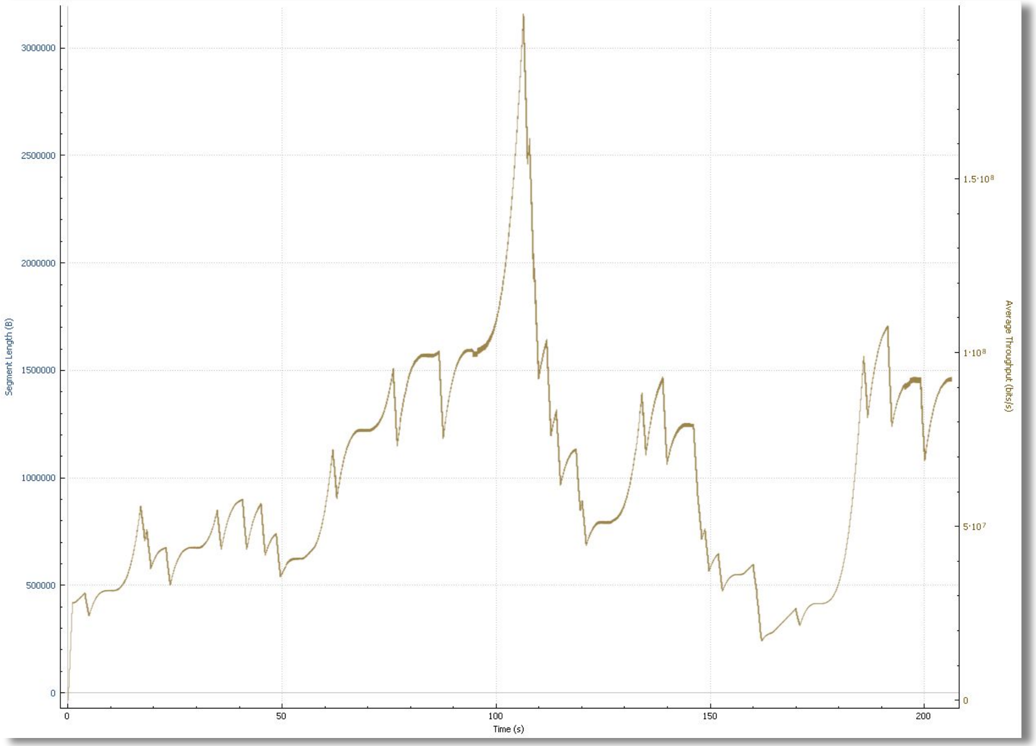
ICMP - Basic RTT (pathing) Tracking



tcpdump + Wireshark (tcptrace)



tcpdump + Wireshark (throughput)



About out-of-order

```
[user@tools ~]$ netstat -s | grep -C2 TCPOFOQueue
  TCPReqQFullDoCookies: 102869
  TCPRcvCoalesce: 7480666215
  TCPOFOQueue: 19877257
  TCPOFOMerge: 2855
  TCPChallengeACK: 1001338
```

- **OFO** = Out Of Order
- TCPOFOQueue means there was a gap in TCP sequences
- Delayed delivery - repaired on delivery without duplicate ACK
  – 10, 11, 12, **GAP**, 14, 15, **13** (**DELAYED**), 16, 17, 18, 19, 20
- Lost packet - Repaired when receiver sends a duplicate ACK
  – 10, 11, 12, **GAP**, 14, 15, 16, (**DUP ACK**), 17, 18, 19, 13 (**RETRANSMIT**), 20

iPerf Triangulation - Setup

```
[user@tools ~]$ iperf3 -u -p 5201 -S 140 -i 10 -t 10 -c  2001:db8::9a30:1 -B  2001:db8::2148:1 -b 5m -P 128
Connecting to host 2001:db8::9a30:1, port 5201
[  5] local 2001:db8::2148:1 port 39985 connected to 2001:db8::9a30:1 port 5201
[  7] local 2001:db8::2148:1 port 48388 connected to 2001:db8::9a30:1 port 5201
[  9] local 2001:db8::2148:1 port 59213 connected to 2001:db8::9a30:1 port 5201
[ 11] local 2001:db8::2148:1 port 41572 connected to 2001:db8::9a30:1 port 5201
[ 13] local 2001:db8::2148:1 port 53198 connected to 2001:db8::9a30:1 port 5201
[ 15] local 2001:db8::2148:1 port 58742 connected to 2001:db8::9a30:1 port 5201
[ 17] local 2001:db8::2148:1 port 53768 connected to 2001:db8::9a30:1 port 5201
[ 19] local 2001:db8::2148:1 port 38241 connected to 2001:db8::9a30:1 port 5201
[ 21] local 2001:db8::2148:1 port 57208 connected to 2001:db8::9a30:1 port 5201
[ 23] local 2001:db8::2148:1 port 42364 connected to 2001:db8::9a30:1 port 5201
[ 25] local 2001:db8::2148:1 port 46471 connected to 2001:db8::9a30:1 port 5201
[ 27] local 2001:db8::2148:1 port 36711 connected to 2001:db8::9a30:1 port 5201
.
.
.
```

- UDP Mode - Pure loss, no TCP effects, unidirectional
- 128 wide - Sample lots of paths, chance for intersection
- Provides source, destination IP and port for tracing later
- Mark with correct DSCP

iPerf Triangulation - Detection

```
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
[  5]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.067 ms  0/4377 (0%)  receiver
[  7]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
[  7]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.069 ms  0/4377 (0%)  receiver
[  9]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
[  9]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.071 ms  0/4377 (0%)  receiver
[ 11]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
[ 11]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.083 ms  0/4377 (0%)  receiver
[ 13]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
[ 13]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.092 ms  0/4377 (0%)  receiver
[ 15]   0.00-10.00  sec  5.96 MBytes  5.00 Mbits/sec  0.000 ms  0/4377 (0%)  sender
.
.
.
```

- Loss report per flow
- Also reports OOO events (without retransmissions)
- Correlate lossy flows to src/dst IP/ports for tracing

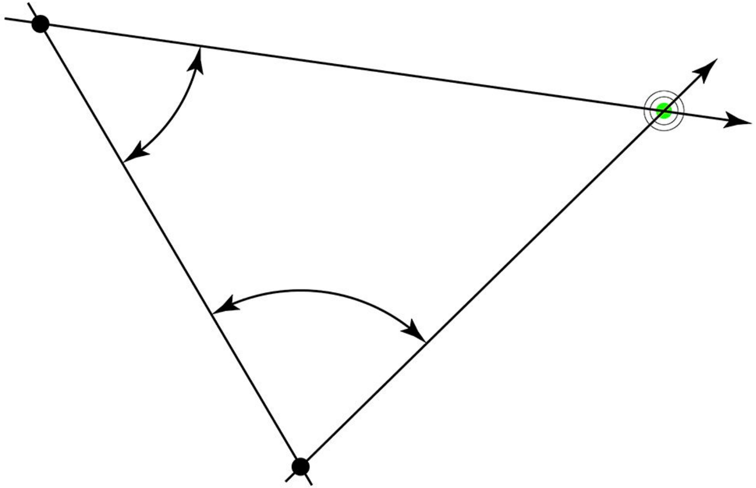
iPerf Triangulation - Tracing

```
[user@tools ~]$ traceroute -q 1 -t 140 -U -p 5201 -w 0.2 -m 20 --sport=38892 2001:db8::9a30:1
traceroute to 2803:6081:608c:9a30::1 (2803:6081:608c:9a30::1), 20 hops max, 80 byte packets
 1  2001:db8::a (2001:db8::a)  0.435 ms
 2  eth101-11-1.fsw004.p109.f01.demo.tfbnw.net (2001:db8::10)  0.584 ms
 3  fc00::2:0 (2001:db8::2:0)  20.698 ms
 4  fc00::15:1 (2001:db8::15:1)  14.563 ms
 5  eth8-7-1.ssw007.s004.f01.demo.tfbnw.net (2001:db8::d8)  0.479 ms
 6  eth4-13-1.fa004-du007.demo.tfbnw.net (2001:db8::46)  0.339 ms
 7  eth4-12-1.ssw007.s006.f01.demo.tfbnw.net (2001:db8::9b)  0.326 ms
 8  eth3-13-1.fsw006.p036.f01.demo.tfbnw.net (2001:db8::47)  0.383 ms
 9  eth1-26-1.rsw039.p036.f01.demo.tfbnw.net (2001:db8::4d)  12.736 ms
10  server01.demo.tfbnw.net (2001:db8::9a30:1)  0.640 ms
```

- UDP Traceroute with the src/dst ports
- Same DSCP to ensure TE pathing
- Collect lossy paths and look for common hops

iPerf Triangulation - RESULTS!

- Spans all of EBB
- No specific paths
- Doesn't cross specific hardware
- Spread out across DC, POPs, Regions, Metros
- Loss ~0.001% or 1 in 100,000 packets



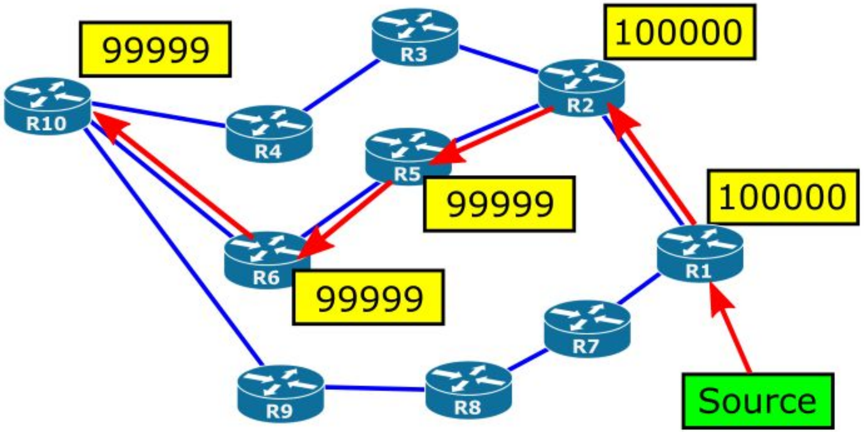
iPerf + ACL - From broad to narrow

```
[user@tools ~]$ iperf3 -u -p 5201 -S 140 -i 10 -t 10 -c 2001:db8::9a30:1 -B 2001:db8::2148:1 -b 100m --cport 32889
Connecting to host 2001:db8::9a30:1, port 5201
[  5] local 2001:db8::2148:1 port 32889 connected to 2001:db8::9a30:1 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  87528
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.000 ms  0/87528 (0%)  sender
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.018 ms  8/87528 (0%)  receiver

iperf Done.
```

- Find a bad path using 128 wide
- Run iPerf at high rate on one tuple 60s
- Place logging ACL on all possible boxes and count packets that arrive

iPerf + ACL - From broad to narrow



- iPerf tells how many packets sent and received
- Per-hop ACL can identify multiple lossy hops

Root cause

- Zoom in on exact router(s) causing loss
- Meta engineer identifies VoQ Egress Drops
- Egress scheduler thinks it has 100Gbps, but actually has ~98Gbps
- Egress scheduler not accounting for MACSEC overhead

Hints to find the elusive

- Exhaust existing data sources
- Create new data sources and tooling
- Extend collections to catch the next one
- tcpdump, wireshark, iperf are you friends
- Zoom in and out from the general and the specific

Questions?

Meta