NANOG

BERKELEY LAB
Lawrence Berkeley National Laboratory

ESnet
ENERGY SCIENCES NETWORK

UNIVERSITY OF CALIFORNIA

U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

ESnet
ENERGY SCIENCES NETWORK

U.S. DEPARTMENT OF ENERGY

# Evaluating Network Buffer Size requirements for Very Large Data Transfers

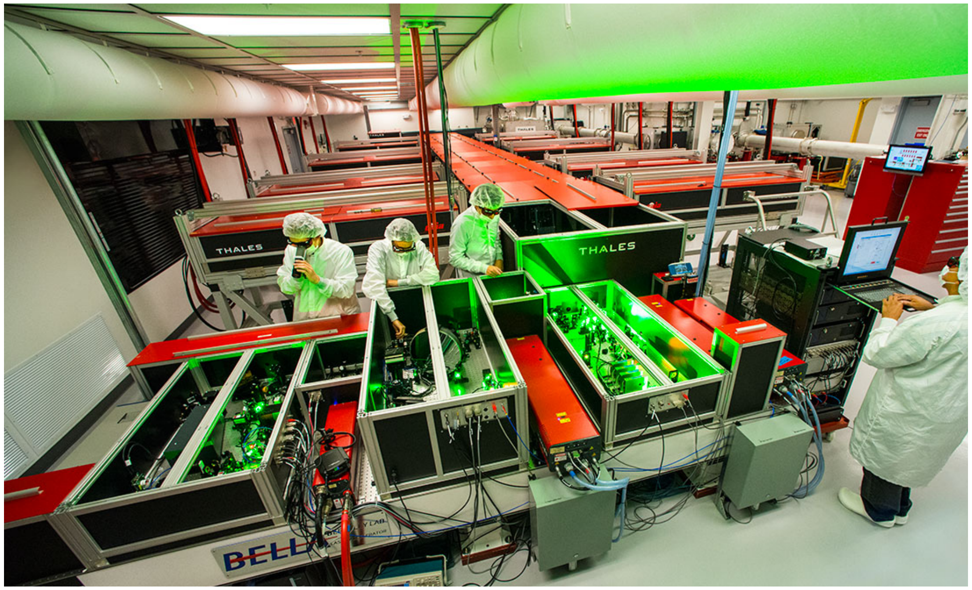
**Michael Smitasin**
Network Engineer
LBLnet Services Group
Lawrence Berkeley National Laboratory

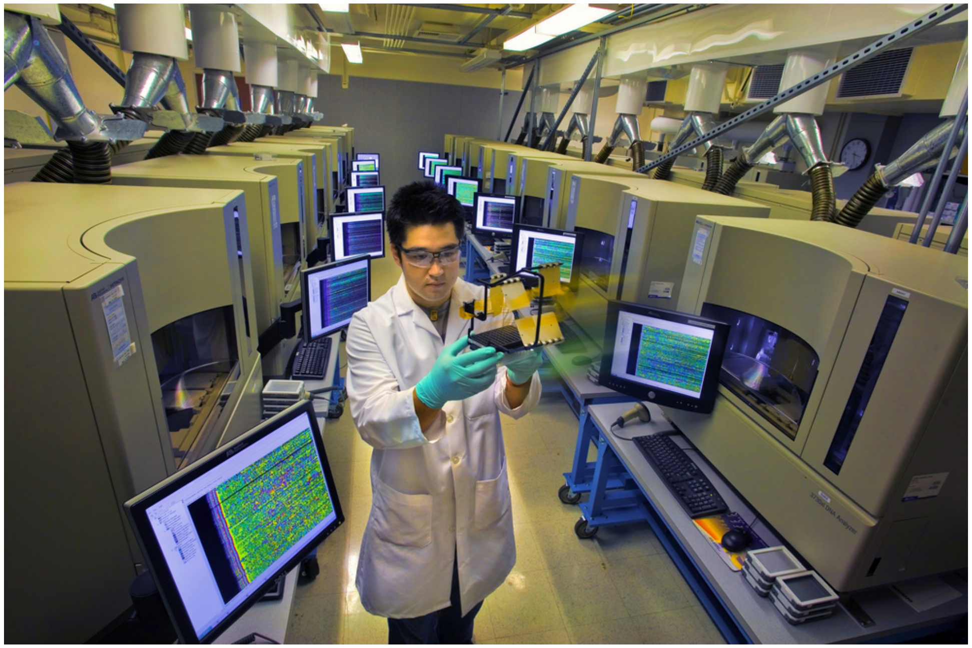
**Brian Tierney**
Staff Scientist & Group Lead
Advanced Network Technologies Group
Energy Sciences Network

**NANOG 64**

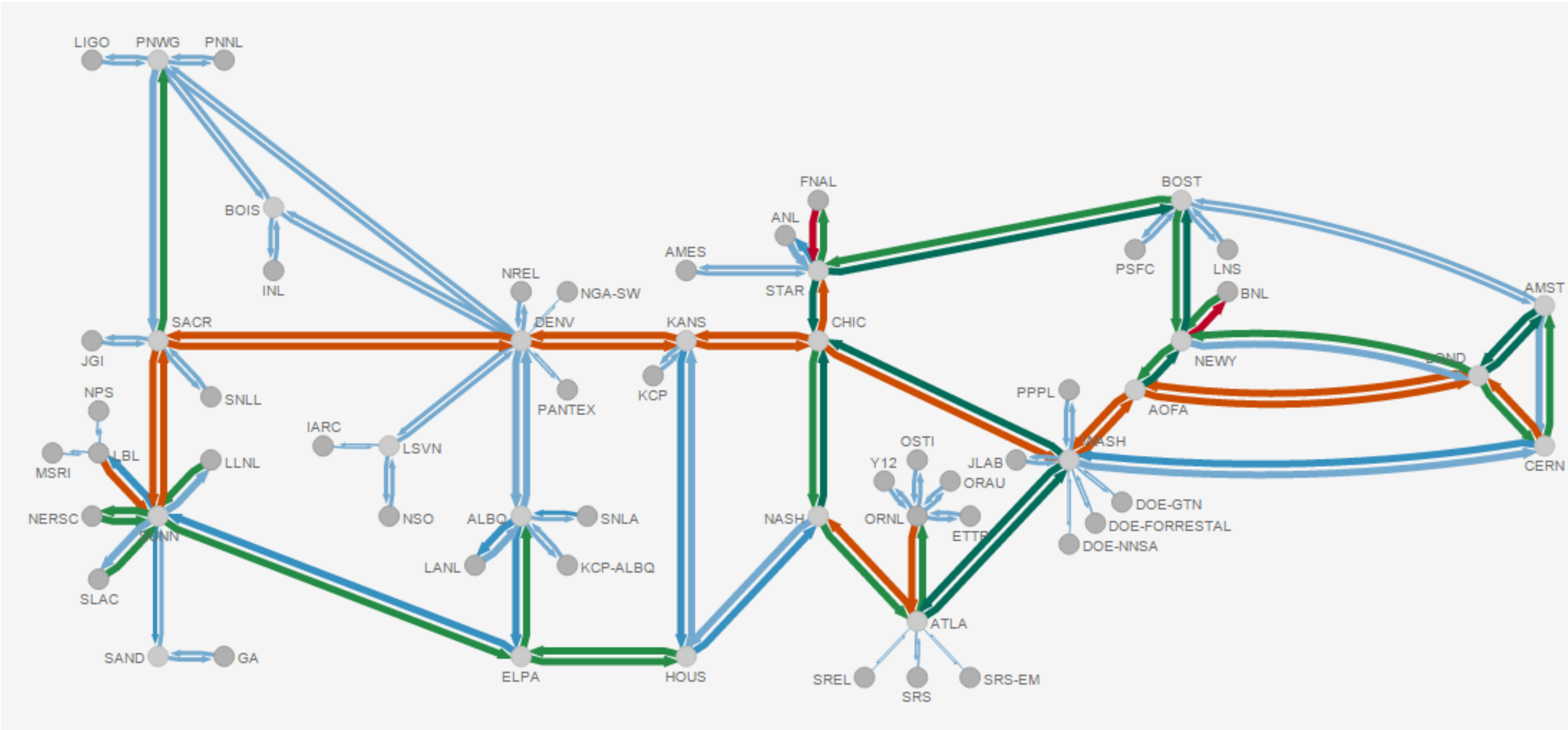
June 2015

# Lawrence Berkeley National Laboratory



# Energy Sciences Network




Connects Department of Energy National Laboratories to universities and research institutions around the world (LBNL's primary provider)

Many sites with 100G connections to ESnet today - Berkeley, Livermore, Stanford, Fermi, Brookhaven, Oakridge, Argonne

# ESnet / DOE National Lab Network Profile

Small-ish numbers of very large flows over very long distances:

Between California, Illinois, New York, Tennessee, Switzerland

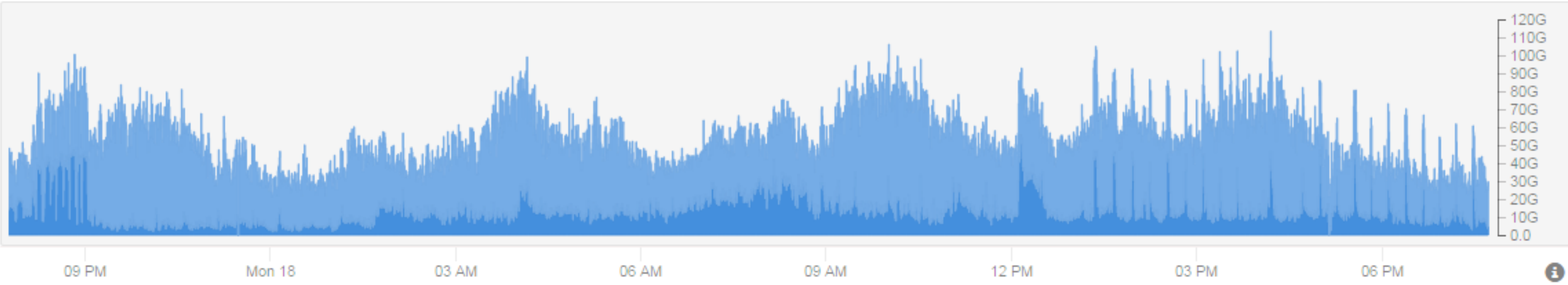
High-speed “Access” links - 100G sites connected to 100G core

Nx10G hosts, future Nx40G hosts, dedicated to Data Transfer

GridFTP / Globus Online / Parallel FTP

LHC detectors to data centers around the world (future 180Gbps)

Electron microscopes to supercomputers (20k – 100k FPS per camera)



# Buffer Bloat at a glance

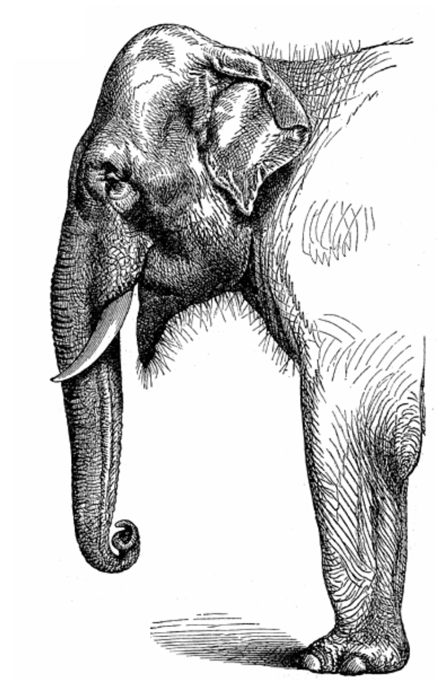
Premise: Big buffers = high latency, which is bad

Typically talking about relatively low-speed flows over short distances

Or, highly-multiplexed core links… 10,000+ simultaneous flows

Case of **mouse** flows vs. **elephant** flows

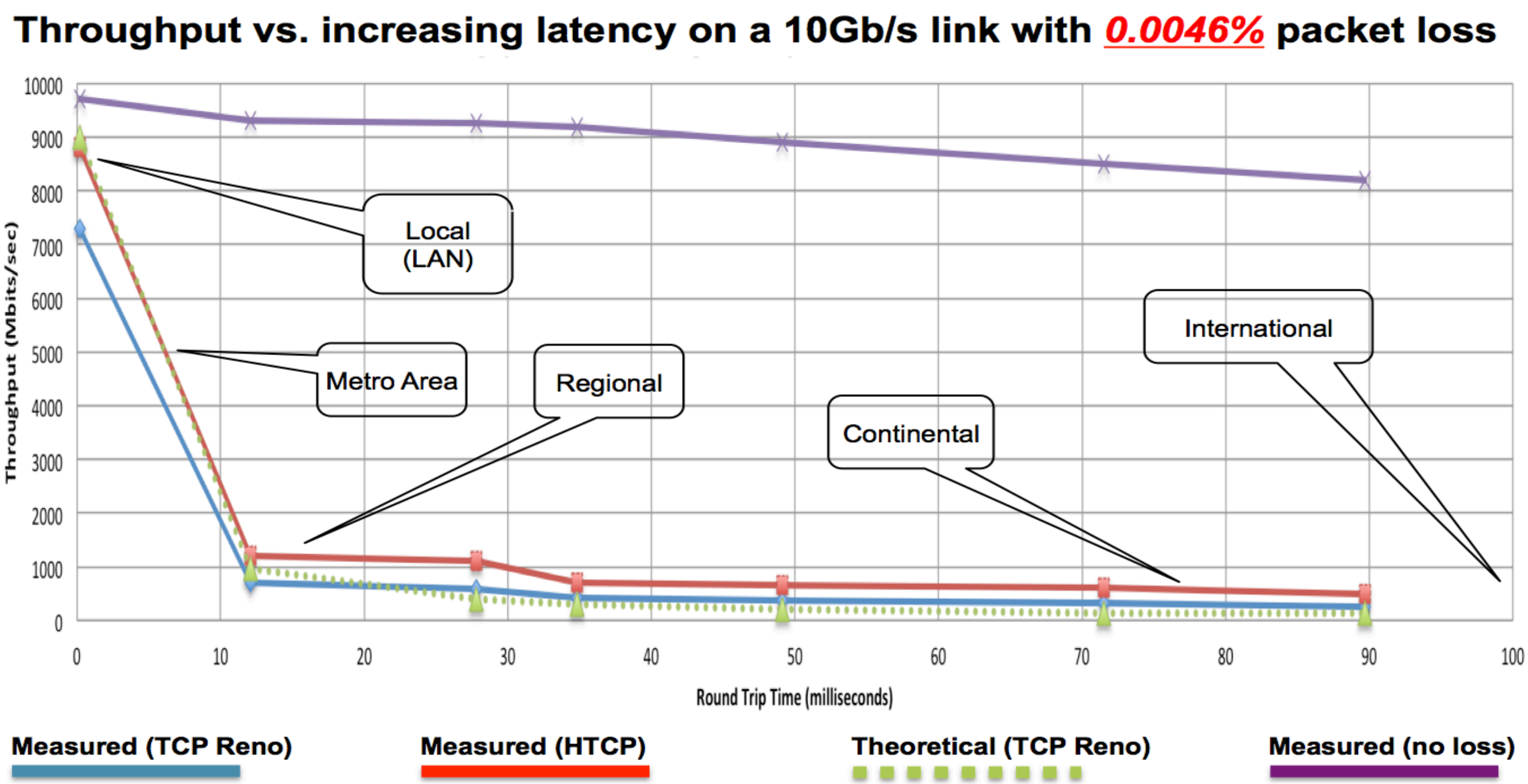
**vs.**



Bufferbloat: Dark Buffers in the Internet, CACM Vol 55
http://cacm.acm.org/magazines/2012/1/144810-bufferbloat

# On Elephants and Packet Loss

We need to send **lots of data** over **long distances**. Insufficient buffers cause us to drop packets frequently, which hinders our throughput.



# Then “Big” Buffers = good?

By “big” we’re still only talking **megabytes** of buffer per 10G port, not **gigabytes**.

Only addressing **very large data transfers** (TB, PB) + **large pipes** (10G & up) + **long distances** (50ms+) between small numbers of hosts.

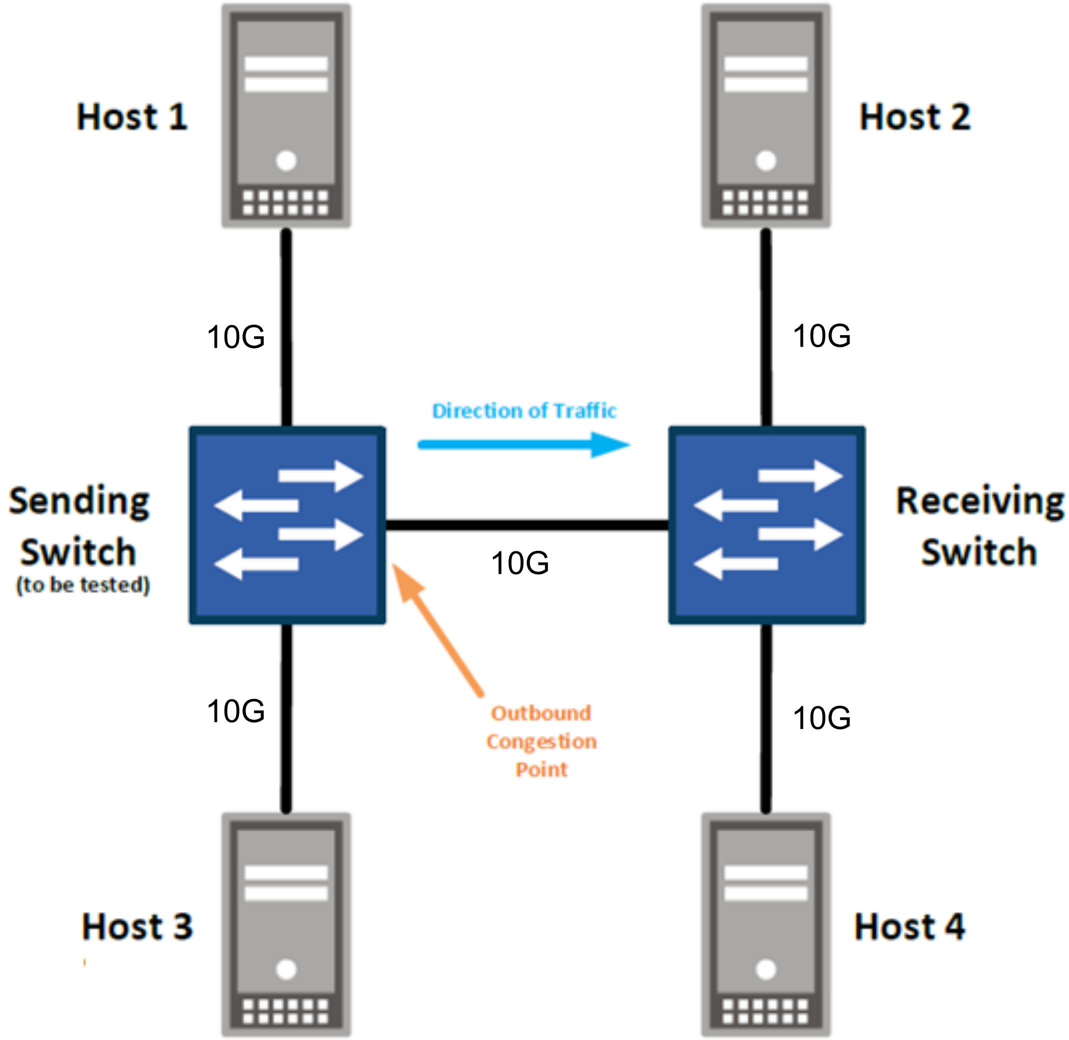
Important to have enough buffering to ride out micro-bursts. A TCP flow may need to drop a packet or two to fit itself to available capacity, but to maintain performance we need to keep TCP from getting stuck in loss recovery mode.

# How can we tell what's sufficient?

**Test with tools that are:**

- Readily Available
- Open Source
- Easy to Use
- Free

# iperf3 in a simulated WAN




**Add latency on hosts 1 and 2:** tc qdisc add dev EthN root netem delay 25ms

# Test Procedures:

Add a 25ms delay to each of hosts **1** and **2**:

```
host1# tc qdisc add dev ethN root netem delay 25ms
host2# tc qdisc add dev ethN root netem delay 25ms
```

Start the iperf3 server on hosts **2** and **4**:

```
host2# iperf3 –s
host4# iperf3 –s
```

On host 3, begin a 2Gbps UDP transfer to host **4** to add congestion:

```
host3# iperf3 -c host4 -u -b2G -t3000
```

On host **1**, begin a 10Gbps TCP transfer, 2 parallel streams for 30 seconds (first 5s omitted from results):

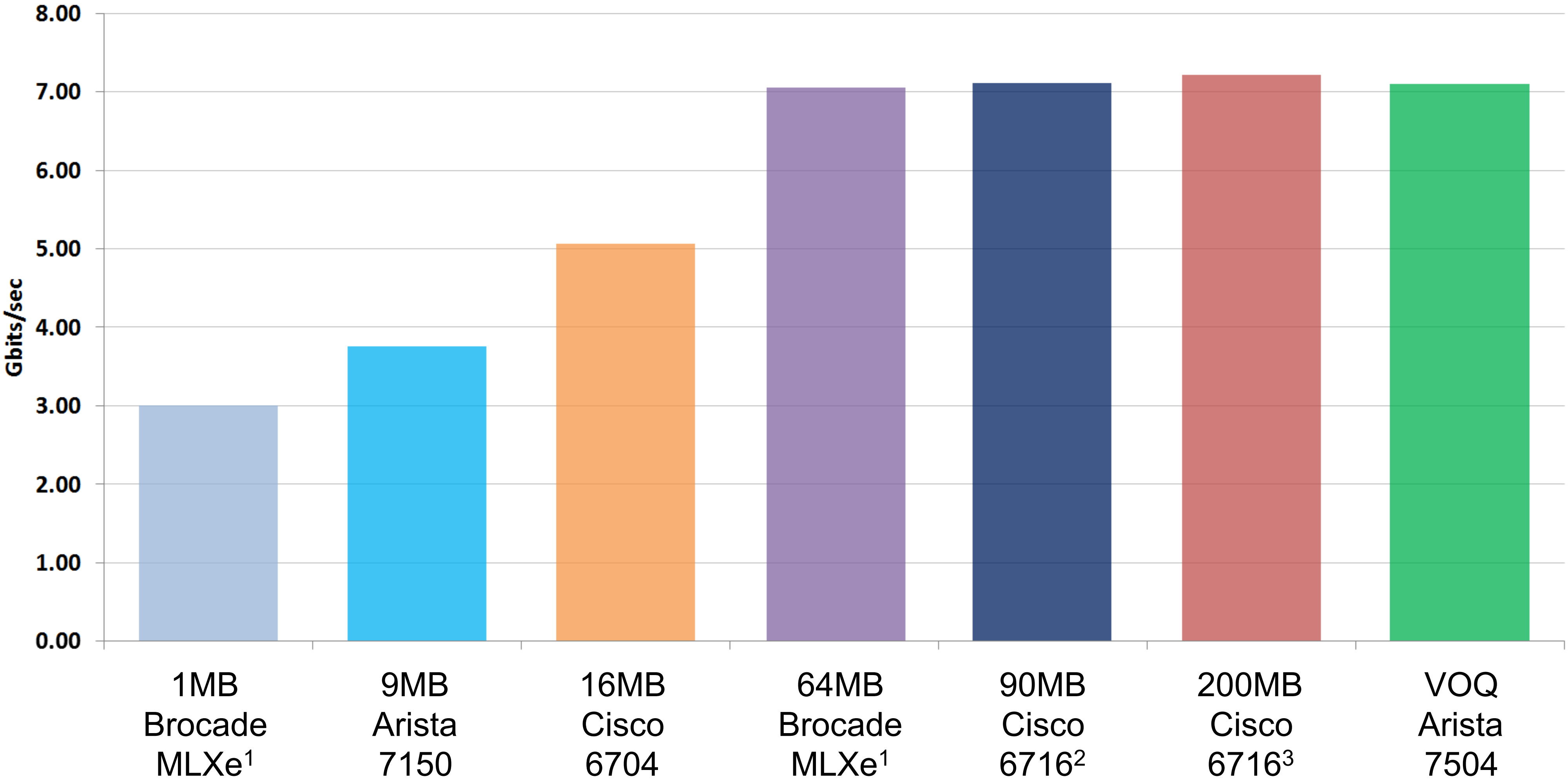
```
host1# iperf3 -c host2 -P2 -t30 -O5
```

# Test Results (example):

```
[  4]  27.00-28.00  sec   276 MBytes  2.32 Gbits/sec    0   15.4 MBytes
[  6]  27.00-28.00  sec   145 MBytes  1.22 Gbits/sec    0   8.66 MBytes
[SUM]  27.00-28.00  sec   421 MBytes  3.53 Gbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
[  4]  28.00-29.00  sec   324 MBytes  2.72 Gbits/sec    5   12.5 MBytes
[  6]  28.00-29.00  sec   195 MBytes  1.64 Gbits/sec    7   9.61 MBytes
[SUM]  28.00-29.00  sec   519 MBytes  4.35 Gbits/sec   12
- - - - - - - - - - - - - - - - - - - - - - - - -
[  4]  29.00-30.00  sec   201 MBytes  1.69 Gbits/sec    0   9.54 MBytes
[  6]  29.00-30.00  sec   126 MBytes  1.06 Gbits/sec    0   6.05 MBytes
[SUM]  29.00-30.00  sec   328 MBytes  2.75 Gbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-30.00  sec  5.85 GBytes  1.68 Gbits/sec   40             sender
[  4]   0.00-30.00  sec  5.83 GBytes  1.67 Gbits/sec                  receiver
[  6]   0.00-30.00  sec  4.04 GBytes  1.16 Gbits/sec   39             sender
[  6]   0.00-30.00  sec  4.01 GBytes  1.15 Gbits/sec                  receiver
[SUM]   0.00-30.00  sec  9.89 GBytes  2.83 Gbits/sec   79             sender
[SUM]   0.00-30.00  sec  9.85 GBytes  2.82 Gbits/sec                  receiver
```

# Average TCP results, various switches

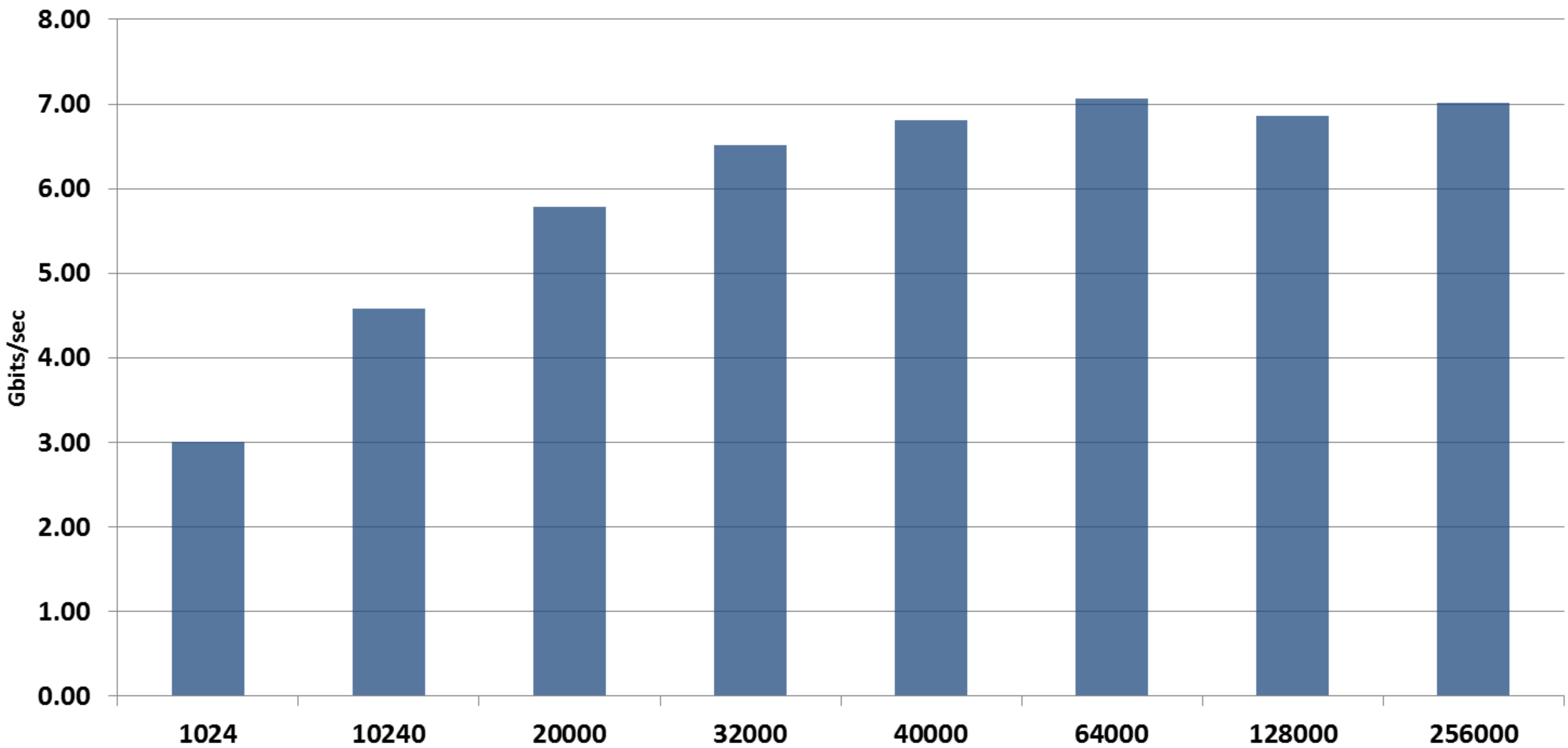
## Buffers per 10G egress port, 2x parallel TCP streams, 50ms simulated RTT, 2Gbps UDP background traffic




[1] NI-MLX-10Gx8-M
[2] Over-subscription Mode
[3] Performance Mode

# Tunable Buffers with a Brocade MLXe[1]

## Buffers per 10G egress port, 2x parallel TCP streams, 50ms simulated RTT, 2Gbps UDP background traffic



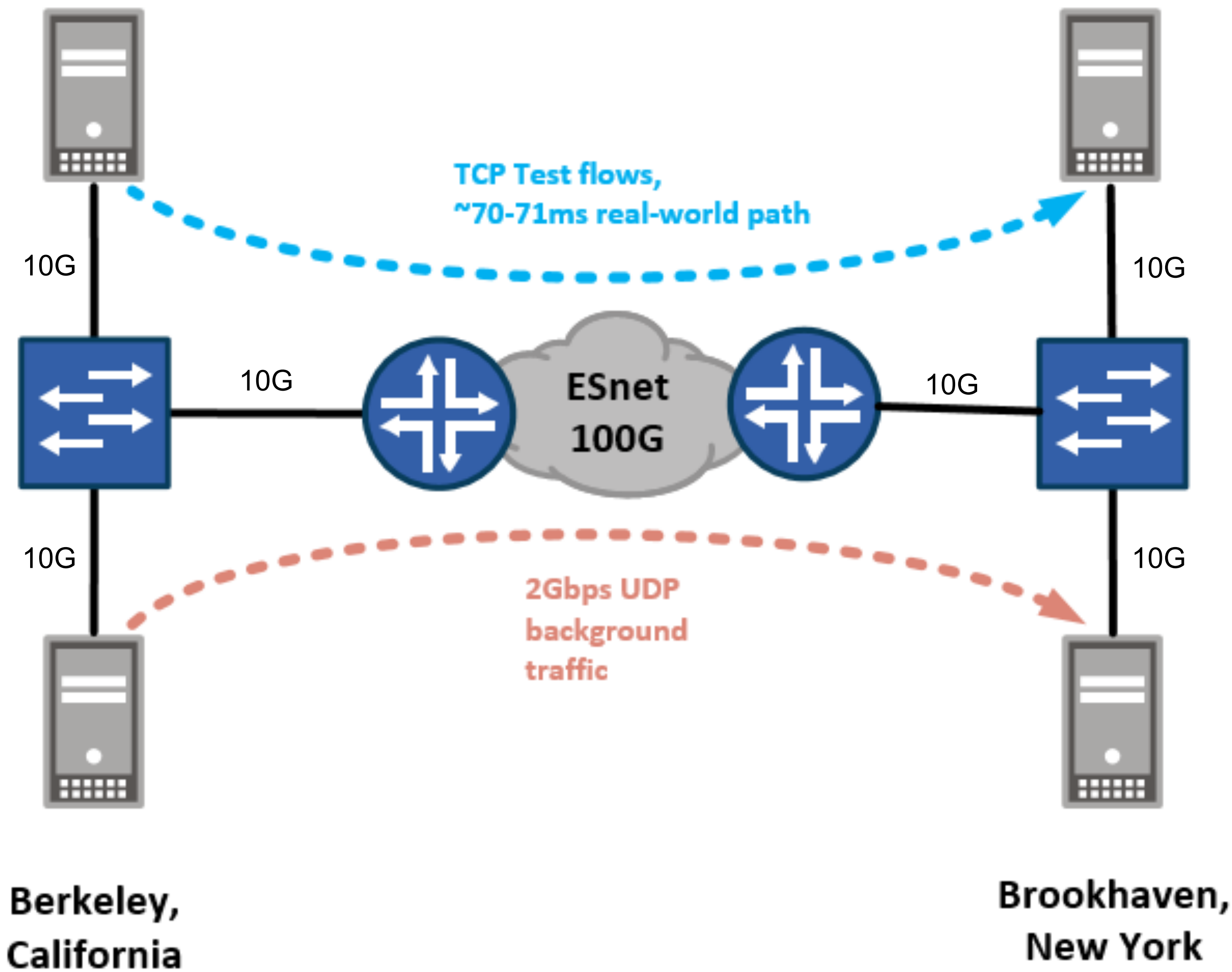

[1] NI-MLX-10Gx8-M Linecard



ESnet ENERGY SCIENCES NETWORK

BERKELEY LAB

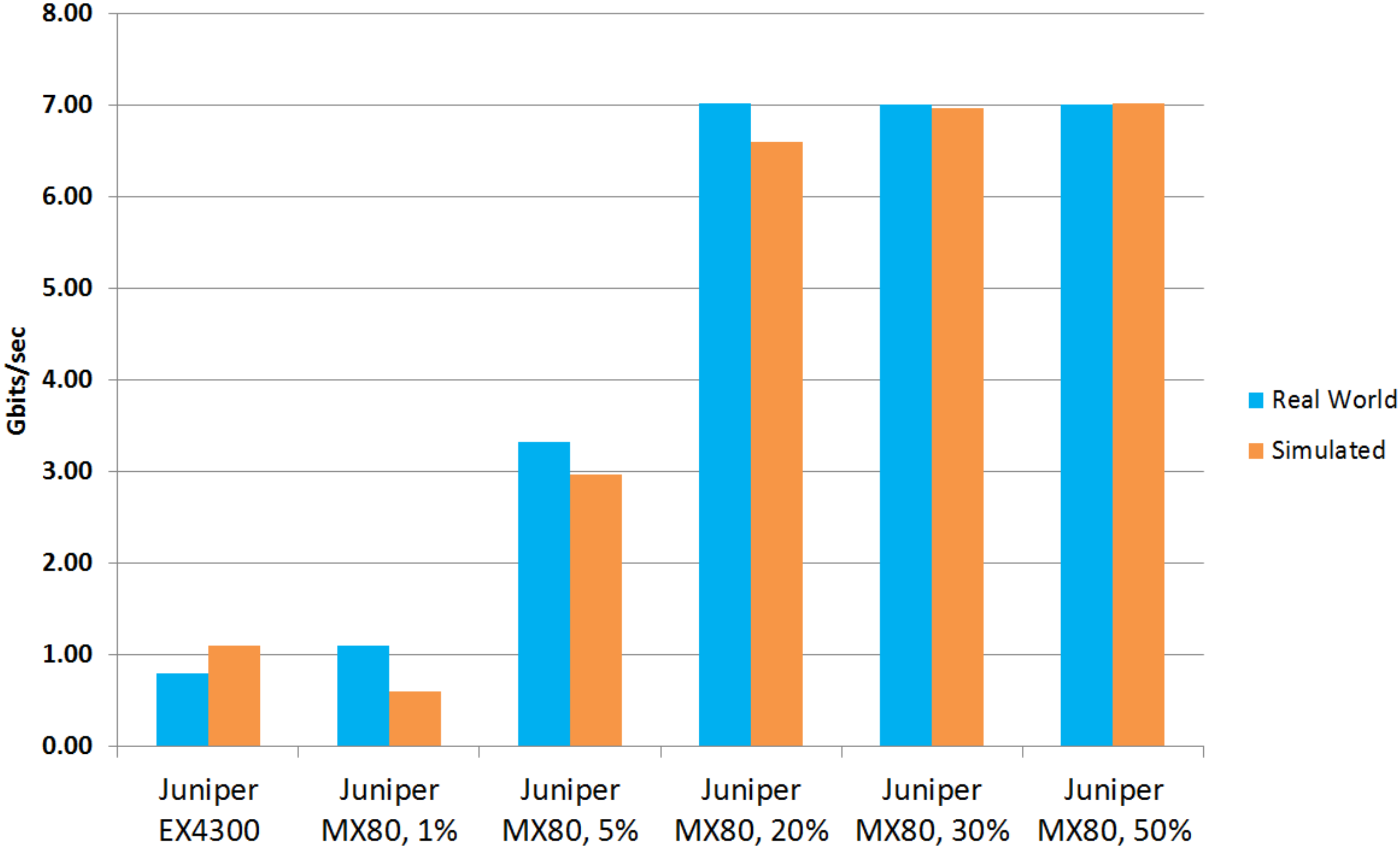
# In the Real World @ 70ms RTT



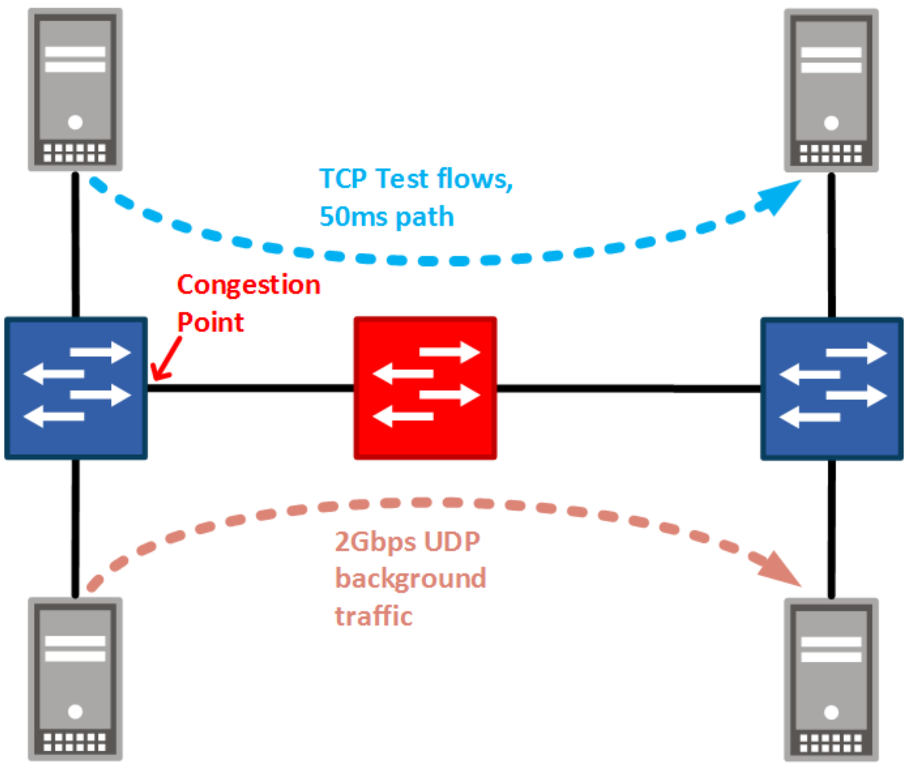

Special thanks to Mark Lukasczyk at Brookhaven National Laboratory for providing far-end test servers

# Real World vs Simulated

70ms RTT, 2x parallel TCP streams,
2Gbps UDP background traffic

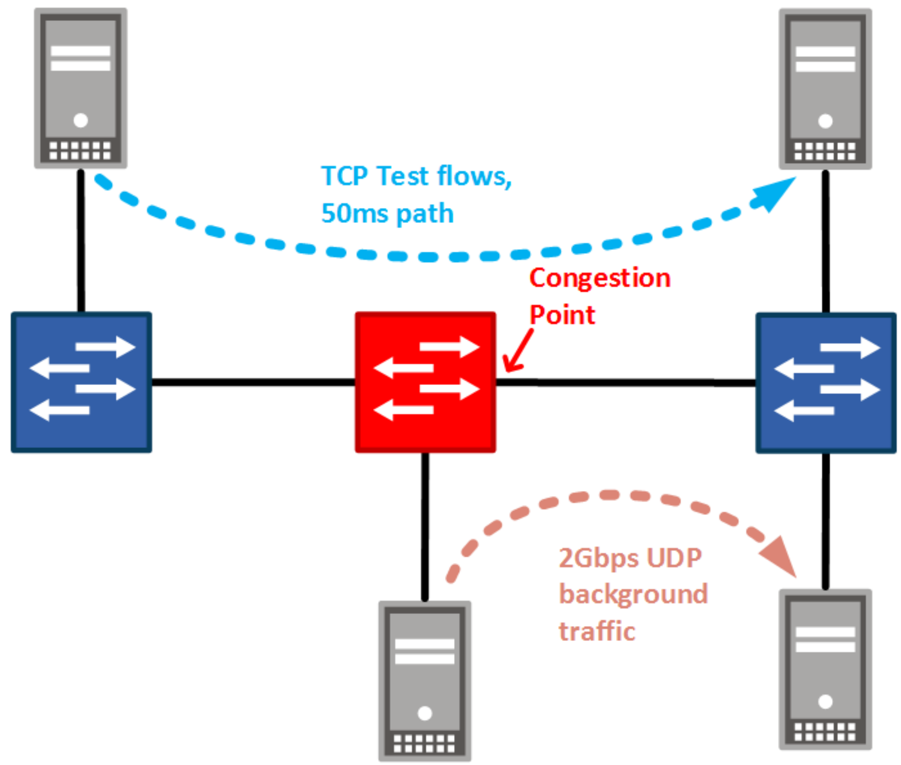


# Can we detect insufficient buffers?




Congestion at

first hop




Congestion at

second hop

# nuttcp test procedures

Simulate WAN connectivity by adding 25ms delay to each

```
host1# tc qdisc add dev eth1 root netem delay 25ms
host2# tc qdisc add dev eth1 root netem delay 25ms
```

Add 2Gbps UDP background traffic on link:

```
host4# iperf3 –s
host3# iperf3 -c host4 -u -b2G -t3000
```

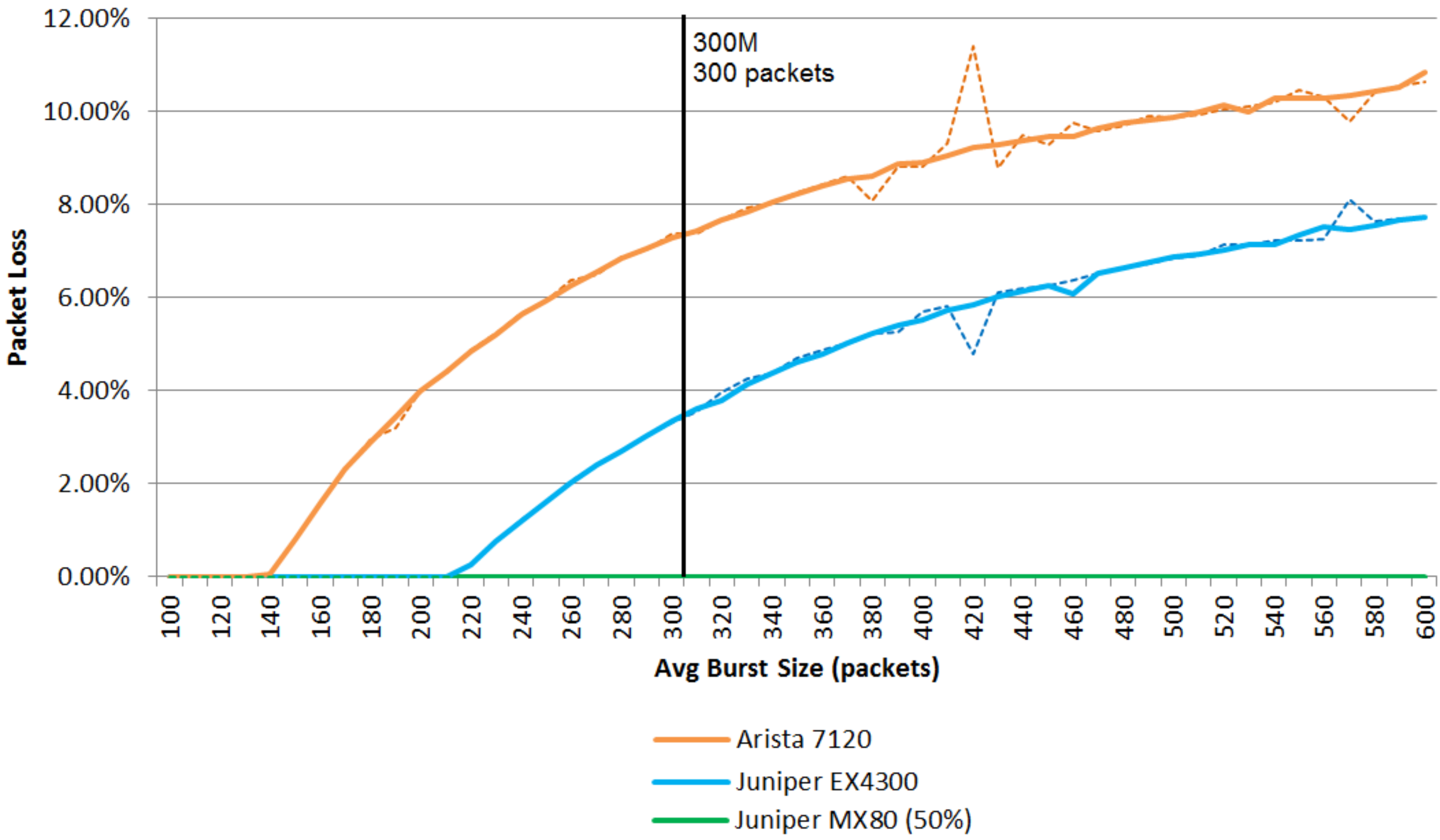
Basic test parameters[1]:

```
host2# nuttcp -S
host1# nuttcp -l8972 -T30 -u -w4m –Ri300m/X –i1 host2
```

X= Burst Size (# of packets)

[1] https://fasterdata.es.net/performance-testing/network-troubleshooting-tools/nuttcp/

# nuttcp results over various burst sizes




Deviations likely due to network emulation.



ESnet ENERGY SCIENCES NETWORK BERKELEY LAB

# nuttcp conclusion

**This will probably have no packet loss on smaller buffer switches:**

```
nuttcp -l8972 -T30 -u -w4m -Ri300m/65 -i1
```

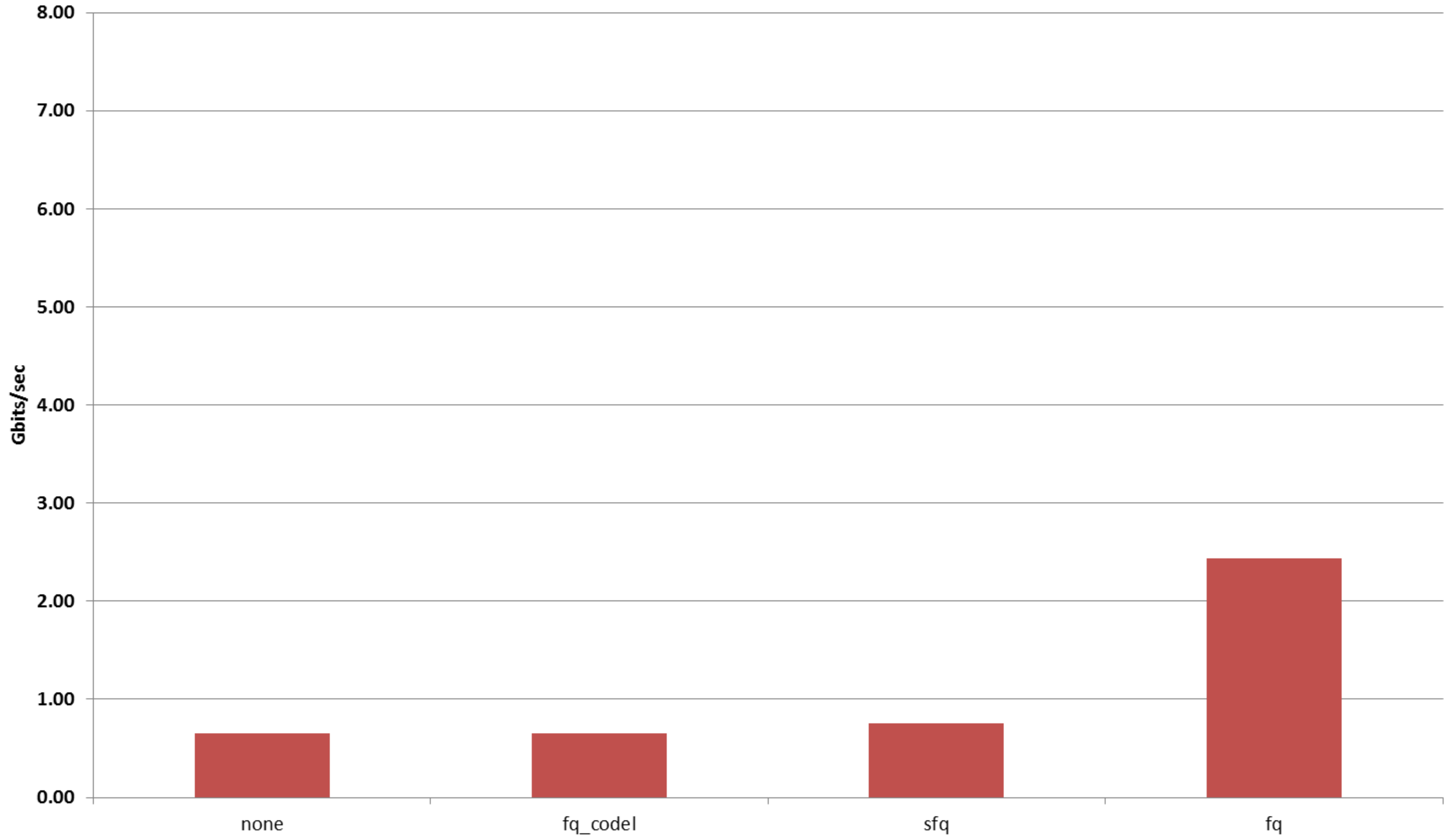
**While this will probably have some:**

```
nuttcp -l8972 -T30 -u -w4m -Ri300m/300 -i1
```

**BUT** only applies to where there is congestion. A small buffer switch that isn't congested won't be detectable with this method.

# Host Queuing Alternatives in Linux kernel 3.11+[1]
# Real World ~70ms RTT, ~9-12MB buffers

```
tc qdisc add dev EthN root [ fq_codel | sfq | fq ]
```




[1] These tests performed in Fedora, FC21/3.19.5-200
No tweaks or tuning to FQ_CoDel, SFQ or FQ

# Additional Information

- **A History of Buffer Sizing**
  http://people.ucsc.edu/~warner/Bufs/buffer-requirements
- **Jim Warner's Packet Buffer Page**
  http://people.ucsc.edu/~warner/buffer.html
- **Faster Data @ ESnet**
  http://fasterdata.es.net
- **Cisco Buffers, Queues & Thresholds on Cat 6500 Ethernet Modules**
  http://goo.gl/gTyryX

**Michael Smitasin**
mnsmitasin@lbl.gov

**Brian Tierney**
bltierney@es.net