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
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

Energy Sciences Network: https://my.es.net
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
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.

http://fasterdata.es.net/network-tuning/tcp-issues-explained/
How can we tell what’s sufficient?

Test with tools that are:

• Readily Available
• Open Source
• Easy to Use
• Free
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 -05
```
## Test Results (example):

<table>
<thead>
<tr>
<th>ID</th>
<th>Interval</th>
<th>Transfer</th>
<th>Bandwidth</th>
<th>Retr</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>27.00-28.00</td>
<td>276 MBytes</td>
<td>2.32 Gbits/sec</td>
<td>0</td>
<td>15.4 MBytes</td>
</tr>
<tr>
<td>6</td>
<td>27.00-28.00</td>
<td>145 MBytes</td>
<td>1.22 Gbits/sec</td>
<td>0</td>
<td>8.66 MBytes</td>
</tr>
<tr>
<td>[SUM]</td>
<td>27.00-28.00</td>
<td>421 MBytes</td>
<td>3.53 Gbits/sec</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>28.00-29.00</td>
<td>324 MBytes</td>
<td>2.72 Gbits/sec</td>
<td>5</td>
<td>12.5 MBytes</td>
</tr>
<tr>
<td>6</td>
<td>28.00-29.00</td>
<td>195 MBytes</td>
<td>1.64 Gbits/sec</td>
<td>7</td>
<td>9.61 MBytes</td>
</tr>
<tr>
<td>[SUM]</td>
<td>28.00-29.00</td>
<td>519 MBytes</td>
<td>4.35 Gbits/sec</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>29.00-30.00</td>
<td>201 MBytes</td>
<td>1.69 Gbits/sec</td>
<td>0</td>
<td>9.54 MBytes</td>
</tr>
<tr>
<td>6</td>
<td>29.00-30.00</td>
<td>126 MBytes</td>
<td>1.06 Gbits/sec</td>
<td>0</td>
<td>6.05 MBytes</td>
</tr>
<tr>
<td>[SUM]</td>
<td>29.00-30.00</td>
<td>328 MBytes</td>
<td>2.75 Gbits/sec</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Interval</th>
<th>Transfer</th>
<th>Bandwidth</th>
<th>Retr</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.00-30.00</td>
<td>5.85 GBytes</td>
<td>1.68 Gbits/sec</td>
<td>40</td>
<td>sender</td>
</tr>
<tr>
<td>4</td>
<td>0.00-30.00</td>
<td>5.83 GBytes</td>
<td>1.67 Gbits/sec</td>
<td></td>
<td>receiver</td>
</tr>
<tr>
<td>6</td>
<td>0.00-30.00</td>
<td>4.04 GBytes</td>
<td>1.16 Gbits/sec</td>
<td>39</td>
<td>sender</td>
</tr>
<tr>
<td>6</td>
<td>0.00-30.00</td>
<td>4.01 GBytes</td>
<td>1.15 Gbits/sec</td>
<td></td>
<td>receiver</td>
</tr>
<tr>
<td>[SUM]</td>
<td>0.00-30.00</td>
<td>9.89 GBytes</td>
<td>2.83 Gbits/sec</td>
<td>79</td>
<td>sender</td>
</tr>
<tr>
<td>[SUM]</td>
<td>0.00-30.00</td>
<td>9.85 GBytes</td>
<td>2.82 Gbits/sec</td>
<td></td>
<td>receiver</td>
</tr>
</tbody>
</table>
Average TCP results, various switches

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

- 1MB Brocade MLXe¹
- 9MB Arista 7150
- 16MB Cisco 6704
- 64MB Brocade MLXe¹
- 90MB Cisco 6716²
- 200MB Cisco 6716³
- VOQ Arista 7504

[1] NI-MLX-10Gx8-M
[2] Over-subscription Mode
[3] Performance Mode
Tunable Buffers with a Brocade MLXe

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

[1] NI-MLX-10Gx8-M Linecard
In the Real World @ 70ms RTT

TCP Test flows, ~70-71ms real-world path

2Gbps UDP background traffic

Berkeley, California

Brookhaven, New York

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 -l18972 -T30 -u -w4m -Ri300m/X -i1 host2

X= Burst Size (# of packets)

nuttcp results over various burst sizes

Deviations likely due to network emulation.
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+
Real World ~70ms RTT, ~9-12MB buffers

tc qdisc add dev EthN root [ fq_codel | sfq | 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/qTyryX

Michael Smitasin
mnsmitasin@lbl.gov

Brian Tierney
bltierney@es.net