

Why Big Data Needs Big Buffer Switches

ANDREAS BECHTOLSHEIM, LINCOLN DALE, HUGH HOLBROOK, AND ANG LI

Today's cloud data applications, including Hadoop, Big Data, Search or Storage, are distributed applications running on server clusters with many-to-many communication patterns. The key to achieving predictable performance for these distributed applications is to provide consistent network bandwidth and latency to the various traffic flows since in most cases it is the slowest flow or query completing last that determines the overall performance.

In this paper, we demonstrate that without sufficient packet buffer memory in the switches, network bandwidth is allocated grossly unfairly among different flows, resulting in unpredictable completion times for distributed applications. This is the result of packets on certain flows getting dropped more often than on other flows, the so-called TCP/IP Bandwidth Capture effect. We present simulation data that show that in heavily loaded networks, query completion times are dramatically shorter with big buffer switches compared to small buffer switches.

Packet Buffers In Cloud Network Switches

The performance of distributed Big Data applications depends critically on the packet buffer size of the datacenter switches in the network fabric. In this paper we compare the performance of both individual switches and network fabrics built with a leaf-spine architecture with switches of various packet buffer sizes under various levels of network load. The results show a dramatic improvement at the application level under load when using big buffer switches.

There are two types of switch chips commonly used in datacenter switches. The first type uses on-chip shared SRAM buffers, which today are typically 12 MBytes of packet buffer memory for 128 10G ports, or approximately 100 KBytes per 10GigE port. The second type of switch chip uses external DRAM buffer, which typically provides 100 MBytes of packet buffer per 10GigE port, or 1000X more than the SRAM switch. At Arista, we build datacenter switches with both types of silicon. The small buffer switch silicon is used in the Arista 7050X, 7250X, 7060X, 7260X and 7300X/7320X switches, while the large buffer switch silicon is used in the Arista 7048T, 7280E/7280R, and 7500E/7500R switches.

Both types of switch product lines have been widely deployed. In this paper we will discuss how the difference in buffer size affects application performance in distributed cloud applications.



A Simple Rack-Top Switch Model

To understand the interaction of multiple TCP/IP flows and switch buffer size it helps to start with devices attached to a single switch. For this, we assumed 20 servers connected with 10 Gigabit Ethernet to a rack top switch with one 40 Gigabit Ethernet uplink. Each server has 10 threads, resulting in a total of 200 flows (20 servers x 10 flows/server) sharing the 40G uplink (5:1 over subscription). The question is what bandwidth is seen by each flow.

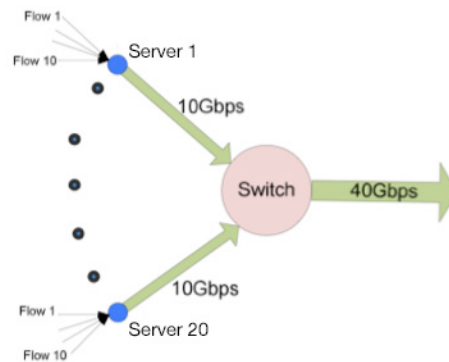


Figure 1: Single switch with 40 Gigabit Ethernet uplink connected with 10Gig to 20 servers with 10 flows each

We modeled the network shown in Figure 1 with the NS-2 network simulator using standard TCP/IP settings using two types of switches: (1) a large buffer switch with 256 MBytes shared packet buffer and perfect per-flow queuing behavior, and (2) a small buffer switch with 4 MBytes of shared packet buffer. The resulting bandwidth per flow is shown in Figure 2 below.

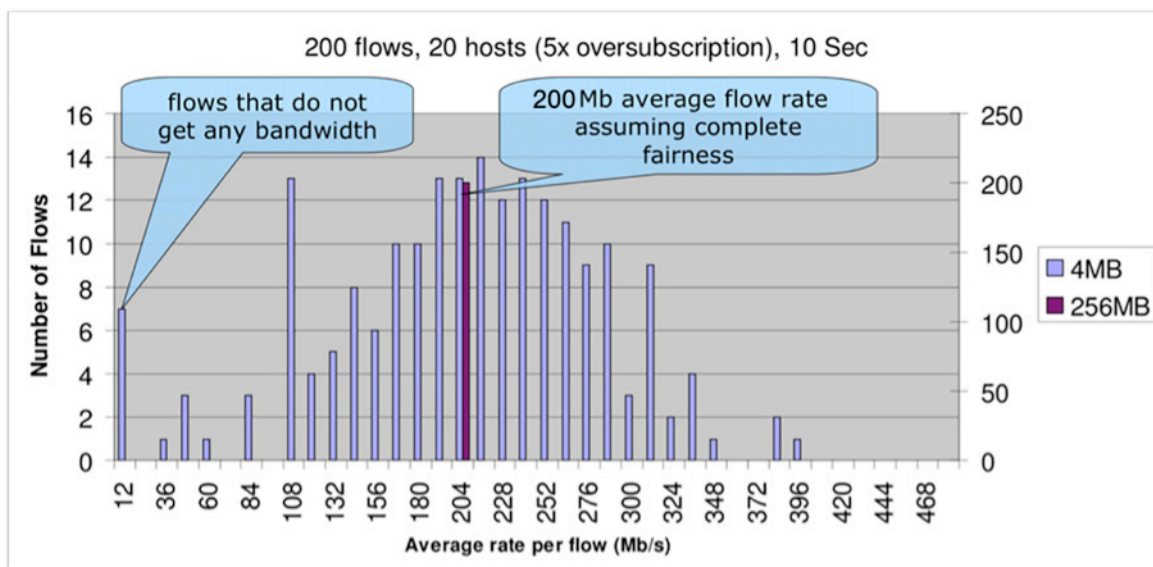


Figure 2: Distribution of bandwidth across different flows in a small buffer versus big buffer switch.

As can be seen, the ideal large buffer switch delivers 200 Mbps per flow. With the small buffer switch however, the bandwidth per flow is distributed along a Gaussian curve. Roughly half the flows receive more bandwidth than the mean, while the other flows receive less bandwidth than what would have been fair, and some flows receive barely any bandwidth at all. The result of this unfair bandwidth allocation between different flows is that some flows have an awfully long tail and have a substantially longer completion time.

The TCP/IP Bandwidth Capture Effect

Why would small buffer switches create such a wide range of bandwidth for different flows? The answer to this is inherent in the way TCP/IP works and TCP/IP flows interact when packets are dropped. The TCP/IP protocol relies on ACK packets from the receiver to pace the speed of transmission of packets by adjusting the sender bandwidth to the available network bandwidth. If there are insufficient buffers in the network, packets are dropped to signal the sender to reduce its rate of packet transmission.

When many flows pass through a congested switch with limited buffer resources, which packet is dropped and which flow is impacted is a function of whether a packet buffer was available in the precise moment when that packet arrived, and is therefore a function of chance. TCP flows with packets that are dropped will back off and get less share of the overall network bandwidth, with some really “unlucky” flows getting their packets dropped all the time and receiving barely any bandwidth at all. In the meantime, the “lucky” flows that by chance have packets arriving when packet buffer space is available do not drop packets and instead of slowing down will increase their share of bandwidth. The result is a Poisson-like distribution of bandwidth per flow that can vary by more than an order of magnitude between the top 5% and the bottom 5% of flows. We call this behavior the “TCP/IP Bandwidth Capture Effect”, meaning in a congested network with limited buffer resources certain flows will capture more bandwidth than other flows.

The TCP/IP Bandwidth Capture Effect is conceptually similar to the Ethernet CSMA/CD bandwidth capture effect in shared Ethernet, where stations that collide with other stations on the shared LAN keep backing off and as a result receive less bandwidth than other stations that were not colliding [7][8]. The Ethernet CSMA/CD bandwidth capture effect was solved with the introduction of full-duplex Ethernet and Ethernet switches that eliminated the CSMA/CD access method. The TCP/IP Bandwidth Capture Effect can be solved by switches that have sufficient buffering such that they don’t cause TCP retransmission timeouts.

Note that the overall throughput of the network is not impacted by the TCP bandwidth capture effect since when certain flows time out, other flows will pick up the slack. Thus one does not need large packet buffers to saturate a bottleneck link, assuming sufficient number of flows, however that does not say anything about how the bandwidth of the bottleneck link is allocated to the various flows. Without sufficient buffering, the allocation of bandwidth will be very much uneven, which can have a very significant impact on distributed applications that depend on all flows completing.

In summary, the TCP/IP Bandwidth Capture Effect is inherent in the way the TCP/IP protocol interacts with networks that have insufficient buffer resources and drop packets under load. In contrast, large buffer switches drop virtually no packets, enabling the network to provide predictable and fair bandwidth across all flows.

Leaf-Spine Cloud Network Simulation

We next expanded our simulation model to a leaf-spine network consisting of five racks and 20 servers per rack, for a total of 100 servers. Each server in a rack is connected with 10 GigE to the rack leaf switch (20 per leaf switch, or 200 Gbps maximum bandwidth), and each leaf switch connects to the spine layer with two 40 GigE ports (2.5:1 oversubscribed).

We studied the query and flow completion times for distributed applications for various levels of network loading and for different buffer sizes in the leaf and spine switches. The detailed results are presented in [1]. Our server and network fabric topology was identical to the one used in [2], except we did not stop at 70% network loading but simulated all the way to 95% network load.

We used the NS3 simulator [3] with TCP NewReno [4] (the official implementation in NS3) with the initial RTO set to 200 msec (the standard Linux value). Flow control was disabled.

For the workload, we simulated a dynamic workload for foreground query traffic consisting of each server sending requests to n other servers. The background traffic was file transfers between randomly chosen pairs of servers. The exact traffic model is described in [1].

Flow Completion Times At Various Network Loads

We found that flow completion times (FCT) increased dramatically for small-buffer switches as the load on the spine increased. At 95% network loading and 95% flow completion the FCT with small-buffer switches increased to 600 milliseconds. With big buffer switches under the same conditions, FCT remained at less than 10 milliseconds, a 60:1 performance advantage.

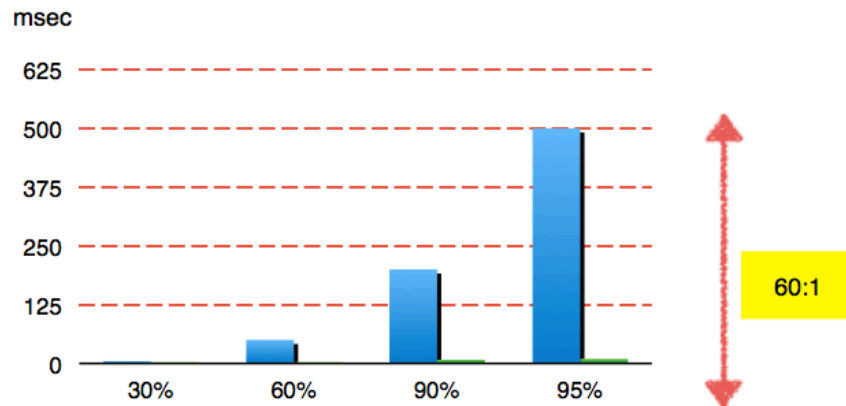


Figure 3: Flow Completion Times with small and large buffer Switches under various network loads

Query Completion Times As A Function Of Network Load

We found similarly dramatic results for Query Completion Times (QCT). Under 90% network load, and for 90% query completion, the small buffer network took more than 1 second. In contrast, under the same loading conditions, the QCT with big buffer switches best performance was a mere 20 msec, a 50:1 performance advantage.

In contrast to [2], we found that for spine traffic loads of 90% packet loss at the spine layer is equally disruptive to packet loss at the leaf layer. This was not found in [2] because their study did not simulate high traffic load at the spine layer, which did not stress the spine switch.

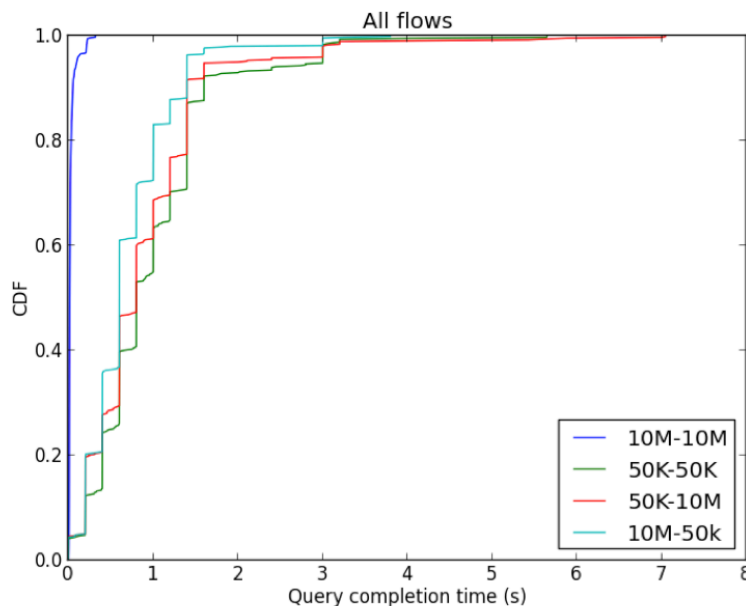


Figure 4: Query Completion Times with small and large buffer Switches under 90% network load

Impact On Real World Applications

The simulations we have run use traffic flows that are common in many Big Data applications. The exact impact of the network on application performance of course depends on the actual network communication patterns, network loading, and so on.

A real-life example of the impact of big packet buffers on application performance is Figure 5 below from DataSift, which replaced small buffer legacy GigE switches with the large buffer Arista 7048 switch, dropping application query completion times from 15 msec to 2.5 msec, a 6X improvement.



Figure 5: Average latency seen in a Big Data cluster before and after introducing big buffer switches [5].
<http://architects.dzone.com/articles/platform-performance-gains>

The effect that variable network performance can have on large scale web services can be found in [6] which shows the impact of the “long tail” of variances in a large cloud compute cluster on the overall task completion time, stretching it to 140 ms (see Table 1 below).

Table 1: Individual-leaf-request finishing times for a large fan-out service tree (measured from root node of the tree)

	50%ile Latency	95%ile Latency	99%ile Latency
One random leaf finishes	1ms	5ms	10ms
95% of all leaf requests finish	12ms	32ms	70ms
100% of all leaf requests finish	40ms	87ms	140ms

Summary

We presented simulation data that demonstrates that the way TCP/IP interacts with insufficient packet buffers in network switches under load leads to gross unfairness in bandwidth allocation among different flows in the network. Flows that experience packet drops reduce their data rate and bandwidth consumed, with some flows getting very little if any bandwidth, while other flows that experience fewer packet drops receive more than their fair share of the available network bandwidth. We call this the TCP/IP Bandwidth Capture Effect.

With a single small-buffer switch, this effect can lead to differences of 10:1 in bandwidth seen per flow. In a network with multiple switching tiers, the bandwidth allocation per flow can vary by 100:1. This wide range of bandwidth as seen by different flows can lead to highly variable completion times for distributed cloud applications that depend on the all flows or queries to complete.

In our simulations of a leaf-spine network under heavy load, we have seen query and flow completion times of more than one second with small buffer switches, while under the same conditions with large buffer switches the maximum query completion time was 20 millisecond, a 50:1 performance advantage with big buffer switches over small buffer switches.

The gap between deep buffer and shallow buffer switches will only continue to widen over time as higher performance storage (Flash/SSD), higher performance servers and increase use of TCP offloading occurs within servers/storage over time.

In summary, large buffer switches dramatically improve application performance under high loads. Customers that have deployed Hadoop and other Big Data applications, High Performance storage and distributed applications with big buffer switches have seen significant performance improvements compared to small buffer switches.

References

- [1] Simulation Study of Deep vs Shallow Buffer Switches in Datacenter Environments Ang Li and Hugh Holbrook, Arista Networks Inc.
- [2] On the Data Path Performance of Leaf-Spine Datacenter Fabrics, Mohammad Alizadeh and Tom Edsall, 2013 IEEE 21st Annual Symposium on High-Performance Interconnects
- [3] The NS-3 Network Simulator, Tom Henderson, George Riley, Sally Floyd, and Sumit Roy <http://www.nsnam.org/overview/what-is-ns-3/>
- [4] The TCP NewReno Implementation, RFC6582, Tom Henderson, Sally Floyd, et al <http://tools.ietf.org/html/rfc6582>
- [5] Platform Performance Gains with Arista Switches <http://architects.dzone.com/articles/platform-performance-gains>
- [6] The Tail at Scale, Jeffrey Dean and Luiz Andre Barroso, Google Inc Communications of the ACM, February 2013
- [7] The Ethernet Capture Effect: Analysis and Solution. K.K Ramakrishnan and Henry Yang. Proceedings of IEEE 19th Conference on Local Computer Networks, MN, USA, October 1994
- [8] Impact of the Ethernet capture effect on bandwidth measurements, Mats Bjorkmann and Bob Melander, Networking 2000, Volume 1815, Page 156-167

Santa Clara—Corporate Headquarters

5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500

Fax: +1-408-538-8920

Email: info@arista.com

Ireland—International Headquarters

3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office

Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office

9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office

10 Tara Boulevard
Nashua, NH 03062



Copyright © 2016 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. Sep 2016 02-0037-01