RoCE Rocks without PFC: Detailed Evaluation

Alexander Shpiner, Eitan Zahavi, Omar Dahley, Aviv Barnea, Rotem Damsker, Gennady Yekelis,
Michael Zus, Eitan Kuta and Dean Baram
Mellanox Technologies
Yokneam, Israel
alexshp,eitan,omard,avivb,rotemd,gennadyy,michaelz,eitank,deanb@mellanox.com

ABSTRACT

In recent years, the usage of RDMA in data center networks has increased significantly, with RDMA over Converged Ethernet (RoCE) emerging as the canonical approach for deploying RDMA in Ethernet-based data centers. Initial implementations of RoCE required a lossless fabric for optimal performance. This is typically achieved by enabling Priority Flow Control (PFC) on Ethernet NICs and switches. The RoCEv2 specification introduced RoCE congestion control, which allows throttling the transmission rate in response to congestion. Consequently, packet loss is minimized and performance is maintained, even if the underlying Ethernet network is lossy.

In this paper, we discuss the latest developments in RoCE congestion control. Hardware congestion control reduces the latency of the congestion control loop; it reacts promptly in the face of congestion by throttling the transmission rate quickly and accurately. The short control loop also prevents network buffers from overfilling under various congestion scenarios. In addition, fast hardware retransmission complements congestion control in severe congestion scenarios, by significantly reducing the performance penalty of packet drops. We survey architectural features that allow deployment of RoCE over lossy networks and present real lab test results.

CCS CONCEPTS

• **Networks → Transport protocols**; **Network performance analysis**; **Data center networks**;

KEYWORDS

RoCE, Congestion Control, Performance

ACM Reference format:
Alexander Shpiner, Eitan Zahavi, Omar Dahley, Aviv Barnea, Rotem Damsker, Gennady Yekelis, Michael Zus, Eitan Kuta and Dean Baram. 2017. RoCE Rocks without PFC: Detailed Evaluation. In *Proceedings of KBNets '17, Los Angeles, CA, USA, August 21, 2017,* 6 pages.
https://doi.org/10.1145/3098583.3098588

1 INTRODUCTION

Several factors define RDMA as a fundamental technology for the data center. First, the relatively high cost of CPU cycles and the associated power demands call for offloading the network protocol processing to the NIC such that the CPU is dedicated for application related computation. Second, data centers regularly host distributed, real-time or interactive applications that require low and deterministic latency for consistent response. Finally, the move to SSDs has required lower network latency in storage environments.

These three requirements are met by a hardware-based RDMA reliable transport. RDMA reduces CPU load, as it bypasses the kernel and the TCP/IP stack and avoids data copy between the user space and the kernel. By offloading the networking stack and eliminating kernel/user context switches, it provides low latency and high throughput. Consequently, RDMA, which was once mainly used for high-performance computing and storage back-end networks, has become a critical technology for hyperscale data centers.

Contrary to lossless networks, used by HPC clusters, where RDMA was first used, most data centers use an Ethernet network, which started as a lossy network. Even though in recent years Data Center Bridging (DCB) extension allows for lossless Ethernet, operators often avoid lossless networks for several reasons, including the need to deal with credit loops created by BGP or PIM, or their perceived vulnerability to hardware faults.

RDMA transport was planned assuming that losses are rare. This is one of the reasons RoCE was first used over lossless networks [6] like the large deployment utilizing Mellanox ConnectX-3 Pro NICs [11] reported by [3] . For these networks, lossless mode is favorable to lossy, since these NICs perform complex packet drop handling in firmware, wasting bandwidth due to the go-back-N scheme of the RoCE transport.

Recently, Mellanox announced Resilient RoCE [9], starting from its ConnectX-4 devices [13], offering the ability to run RoCE over a lossy network. DCQCN, a congestion control algorithm for RoCE [16] that was introduced in ConnectX-3 Pro devices, is implemented by the ConnectX-4 hardware, along with various hardware accelerations for packet retransmission. Thus RoCE over lossy networks rely on an efficient implementation of congestion control and recovery scheme that reduces packet drops. Initial tests of RoCE traffic under congestion and comparison to competing technology was demonstrated in [15].

In this paper, we present measurements of RoCE over ConnectX-4 in various congestion scenarios over lossless and lossy networks. We compare it to TCP and report on its coexistence with TCP. In addition, we survey the required and recommended network configuration for resilient RoCE installation in NICs and switches. Our results show that resilient RoCE can work well in lossy networks.

KBNets '17, August 21, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5053-2/17/08…$15.00
https://doi.org/10.1145/3098583.3098588

2 RESILIENT ROCE BASICS

2.1 Reducing and Handling Packet Drops

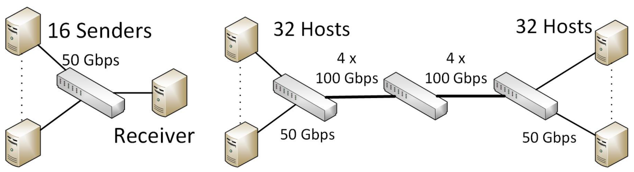
Network congestion happens when the incoming rate of traffic is higher than the switch output link rate, causing packet accumulation in the switch, that eventually overflows the switch buffer.

Lossless networks avoid dropping packets when the buffer is full by using Priority Flow Control (PFC) [1][1] to pause the incoming packet transmission before the buffers that receive the packets overfill. However, the losslessness property also has some disadvantages, such as deployment complexity due to the necessity of additional configuration of network devices.

In lossy networks, when the buffer overflows, packets are dropped. Therefore, congestion causes performance degradation due to the need to identify lost packets and retransmit them, which introduces both algorithmic and implementation challenges.

Congestion control schemes are used to throttle the traffic injection rate in reaction to congestion, reducing the possibility of switch buffer overflow. RoCE NICs implement the DCQCN algorithm for congestion control. DCQCN relies on ECN marking in the congested switch and on a congestion notification packet returned by the receiver, upon arrival of an ECN-marked RDMA packet, as depicted in Figure 2. Full documentation of the algorithm was published in [16]. ConnectX-4 hardware-based implementation of DCQCN allows RoCE resiliency through a fast and accurate response to congestion. While software-based packet processing latency can take hundreds of microseconds, a hardware-based mechanism can react to notification in tens of nanoseconds.

Sometimes congestion control cannot prevent buffer overfilling, and packet loss may still occur. Handling packet loss relies on the InfiniBand transport specification [5], as depicted in Figure 3. Packets are stamped with a packet sequence number (PSN). The responder in the operation accepts packets in order and sends out-of-sequence (OOS) NACK upon receipt of the first packet in a sequence that arrived out of order. OOS NACK includes the PSN of the expected packet. The requestor handles the OOS NACK by retransmitting all packets beginning from the expected PSN using the go-back-N style scheme. The lost packet is fetched again from the host memory. OOS NACK handling is a relatively complex flow in the NIC combining hardware and firmware. In order to minimize the impact of packet loss, retransmissions must be fast and efficient.



(a) Small (b) Large

Figure 1: Evaluation networks.

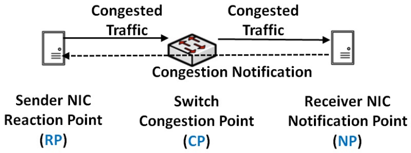


Figure 2: DCQCN Overview.

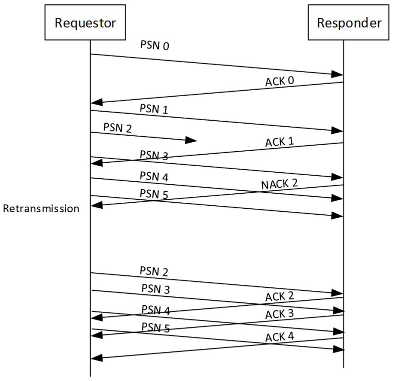


Figure 3: Handling packet loss.

2.2 Network Quality-of-Service Configuration

RoCE works out of the box. However, similarly to other networking protocols, its performance can get additional boost by complementing it with quality of service tools provided by most of standard networking devices.

Two standards are used to classify traffic for quality of service: PCP and DSCP. PCP uses a 3-bit Priority Code Point (PCP) field in the VLAN header. Newer devices support DSCP-based classification, by relying on an 8-bit Differentiated Service Code Point (DSCP) field in the IP header. This field exists in IP-based packet, and its value is preserved along the route. Ref. [3] describes deployment of RoCE using DSCP-based priority flow control.

Traffic classification can boost RoCE performance in several manners. The first is Priority Flow Control (PFC) [1]. PFC relies on traffic classification to store the traffic of each priority in separate ingress buffers. Transmission is paused independently for each priority when buffers are about to overflow.

The second is guaranteed service for Congestion Notification Packets (CNP). ConnectX devices are able to set configurable PCP and DSCP values on CNP. Setting a unique class value to CNPs, allows configuring the network devices to deliver CNPs with guaranteed buffering and high priority scheduling. This enables the CNPs to reach from the notification point to the reaction point faster without suffering network congestion, thus shortening the congestion control loop delay. This is an important benefit compared to the TCP implementation of congestion control, in which congestion information is piggybacked with the data, and thus incurs network queuing delays.

Finally, the presence of different traffic types in the network can affect the performance of RoCE, especially if the non-RoCE traffic is

[1] The InfiniBand specification defines a slightly different approach for flow control.

not congestion controlled. This also occurs with co-existence with other congestion control methods that apply different schemes and implementations. Therefore, non-RoCE traffic may react differently, for example less aggressively, to congestion occurrences, and behave unfairly to RoCE-based flows. Unfair coexistence of different TCP flavors in the data center has been studied before [2, 4, 8, 10]. Its conclusion predicts unfairness between different flavors of TCP. Furthermore, the same conclusion also applies to the co-existence of RoCE and TCP. Therefore, we recommend that RoCE be isolated from other traffic in terms of guaranteed buffering and scheduling.

3 EVALUATION

3.1 Overview

We evaluated resilient RoCE using ConnectX-4 NICs connected to a Mellanox Spectrum switch [14]. We used default congestion control parameters of WinOF driver version 1.60.16219.0 and Linux OFED version 4.0-1.6.1.0.

Two levels of network configuration were tested. The first level aimed to achieve the peak performance configuration with network quality-of-service support as surveyed in Section 2.2, including enabling flow control. This is therefore referred to as 'lossless network'. The second level is the default configuration without network quality-of-service support and is referred to as 'lossy network'. All the tests were run with the switch's low and high ECN thresholds set to 150KB and 1500KB, respectively. In performance measures, we aimed to maximize and stabilize throughput, while maintaining inter-flow fairness and co-existence with non-RoCE traffic. The hosts created multiple RDMA connections and continuously posted $1KB$ WRITE requests to the receiver using an RDMA bandwidth testing tools [12].

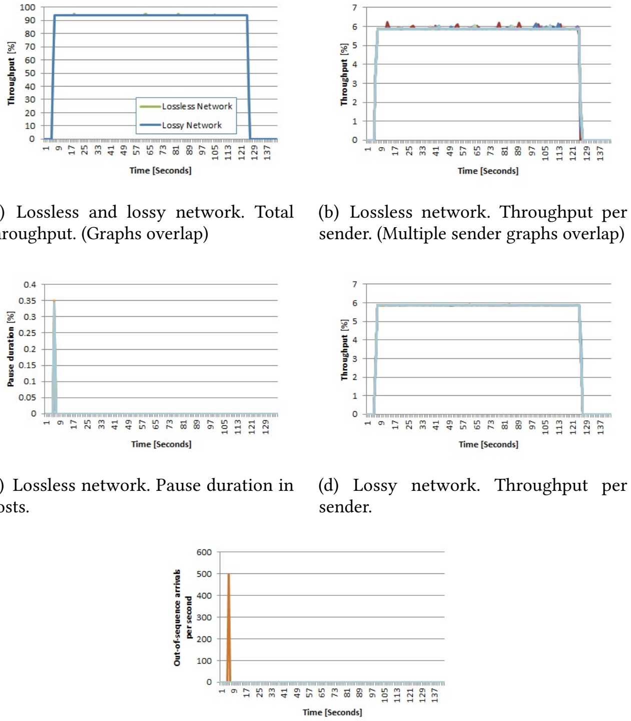
We used two lab setups for the tests: 17-host 'small-scale' to mimic single-rack traffic, and 64-host 'large-scale' scale to mimic inter-rack traffic, as presented on Figure 1. In the small-scale tests we were able to run more connections per host than in large-scale tests, due to testing tools limitations.

For convenience, in the graphs of throughput measurements, we present throughput as a percentage of the link rate.

3.2 Small-Scale 17-Host Single Switch Topology

We began evaluating RoCE resiliency by using many-to-one incast style tests. The hosts were connected to the switch in a star topology. 16 hosts sent traffic to a common receiving host, congesting the incoming link to the receiver, as shown in Figure 1(a).

3.2.1 RoCE over Lossless Network. First we ran experiments of RoCE traffic over a lossless network. That served as a reference of comparison to experiments over a lossy network. In the test, each of the 16 hosts sent traffic over 64 connections to the common receiver. The results of the test are presented in Figure 4. Figure 4(a) shows the total incoming throughput to the receiver over time. The throughput is shown to be stable and roughly close to the link rate. The gap between the throughput and the link rate is attributed to the packet headers overhead. Figure 4(b) shows the throughput per sending host. Every host achieves fair and stable throughput. In addition, Figure 4(c) shows the duration that the incoming link to the congested switch was paused. At the start of the test there was a short period when pauses were sent, until the network converged. That is because every connection started transmitting at the full rate, and the rate was throttled gradually as long it received CNPs. Even then, the maximal sampled pause duration is below 0.35% of the time in a single second. There were no pauses at all during the rest of the test, so we conclude that congestion control works well.



(a) Lossless and lossy network. Total throughput. (Graphs overlap)

(b) Lossless network. Throughput per sender. (Multiple sender graphs overlap)

(c) Lossless network. Pause duration in hosts.

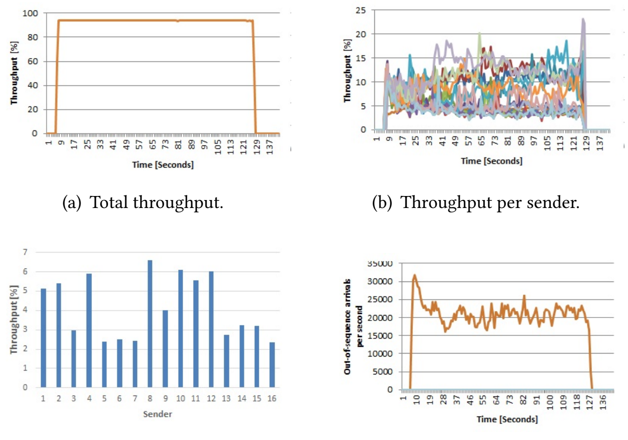
(d) Lossy network. Throughput per sender.

(e) Lossy network. Out of sequence arrivals.

Figure 4: 16 to 1. 64 flows per sender.

3.2.2 RoCE over Lossy Network. In the next test, we evaluated RoCE's behavior over a lossy network. We ran the previous scenario without flow control, such that the packets might be dropped. The test results are shown in Figure 4. The total throughput graph and per-sender throughput in Figures 4(a) and 4(d) show similar results to the experiment over a lossless network. Figure 4(e) shows the out of sequence events sampled by the sender based on received out of sequence NACK (OOS NACK) messages. An out of sequence event indicates a packet drop. Similar to the lossless network case, buffer overflow occurs at the beginning of the test, causing packets to be dropped and triggering OOS NACK messages. However, the system recovers rapidly from congestion, and buffer overflow is not observed in the rest of the test.

Next, we stressed RoCE over the lossy network test with severe incast load to evaluate the system's ability to cope with continuous packet drops. To do that, we increased the number of connections initiated from each sender host to 512. With such a load, the congestion control was not able to prevent drops entirely. Figure 5 shows the results of the test. In Figure 5(d) it is evident that packet



(a) Total throughput.

(b) Throughput per sender.

(c) Average throughput per sender.

(d) Out of sequence arrivals.

Figure 5: 16 to 1. 512 flows per sender. Lossy network.

drops lasted continuously throughout the test, which is reflected by out of sequence NACKs. Despite the multiple drops, as Figure 5(a) shows, the total throughput stays stable and at the maximum level, meaning that the congested link is very close to full utilization. Unfortunately, in Figure 5(d) we see that packet losses caused unfair throughput distribution (Figures 5(b) and 5(c)).

3.2.3 *Coexistence with TCP.* Our evaluation would not be complete without testing the coexistence of RoCE with non-RoCE traffic. Section 2.2 discussed network configuration that supports the coexistence of RoCE traffic with non-RoCE traffic. First, we evaluated the configuration of traffic isolation using multiple priorities. We combined two many-to-one traffic patterns, one over RoCE, the other over TCP, on the same network. RoCE and TCP traffic were separated into different priorities in the NICs and switches, as explained in Section 2.2. Figure 6 shows the results of the test. Figures 6(a) and 6(b) show that RoCE traffic is not suppressed by the existence of TCP traffic. In addition, Figure 6(c) shows that there are no pauses from the switch during the stable stage of the test, meaning that congestion control succeeded in preventing switch buffer overflow. Similarly, Figures 6(a) and 6(d) show that TCP traffic is not significantly harmed, as well. Specifically, the throughput is roughly equally divided between the two types of traffic. We surmise that the dips in TCP throughput were caused by TCP buffer underutilization.

Next, we evaluated the performance without priority configuration. In this test, RoCE traffic, CNPs, and TCP traffic were sent on the same priority. 16 hosts sent 64 RoCE connections and 32 TCP connections to the same destination. Test results are presented in Figure 7. The total goodput of RoCE traffic and of TCP traffic is presented in Figure 7(a). As expected, the fair bandwidth sharing is not maintained as it was with priority separation. Surprisingly, however, the traffic was divided in a ratio similar to the number of respective connections (two-thirds to one-third in RoCE’s favor).

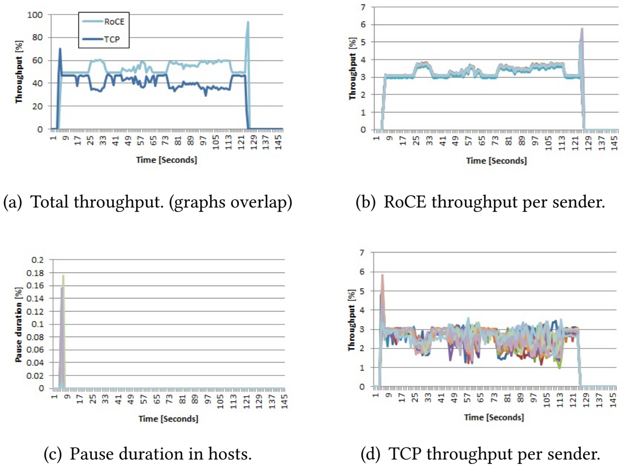
3.2.4 *All-to-All Traffic.* We then evaluated a different network scenario in which we ran an all-to-all traffic pattern with 16 hosts. In this traffic pattern, every host sent traffic to every other host over 16 outgoing and 16 incoming RoCE connections between every pair of hosts. Although theoretically this traffic pattern is congestion-less, due to the bursty traffic injections of the NICs and de-synchronization, queues may build up in the switch, invoking the congestion control. The aim of the benchmark is to verify that congestion control does not hurt performance. Figure 8 shows the outgoing (and therefore also the incoming) throughput of every host. The performance over a lossless network (Figure 8(a)) shows high and stable throughput, while the performance over a lossy network (Figure 8(b)) shows slightly reduced and less stable throughput, but still plausible.

3.2.5 *Dynamic Traffic.* Next, we tested the resiliency of RoCE under dynamically changing traffic scenario. Our dynamic test randomly connects and disconnect some of the flows. Every 1 second, a subset of hosts is chosen to start new connections that last a random time span. We measure the egress ports throughput. The dynamic traffic is expected to decrease the throughput due to the time it takes for the congestion control to prevent buffer to overflow when new flow starts. It also expected to be reduced during the time from the moment a flow stops to the recovery of its bandwidth by the remaining flows.

Figure 9 shows the results of the test. The total throughput is stable in Figure 9(a) and fairness is kept as shown in Figure 9(b). All to all test results of optimal throughput are provided in Figure 9(c).

3.3 Large-scale Tests

Next, we increased the scale of the tests to a 64-host cluster connected in a 2-level fat-tree style topology as depicted in Figure 1(b). Figure 10 shows the results of the large-scale 63-to-one incast tests with 16 flows per sender over a lossless network and a lossy network. The throughput in lossless network was stable and fair without pauses.



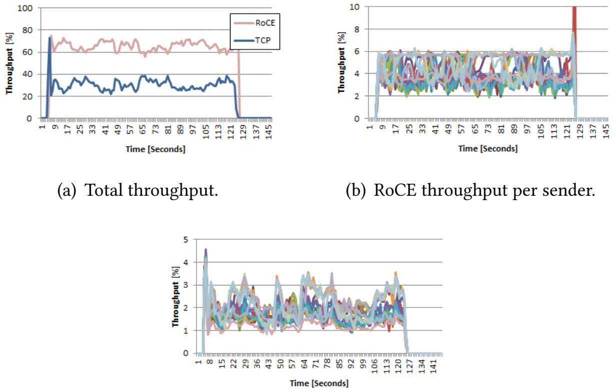
(a) Total throughput. (graphs overlap)

(b) RoCE throughput per sender.

(c) Pause duration in hosts.

(d) TCP throughput per sender.

Figure 6: RoCE over lossless with TCP over lossy with isolated priorities. 16 to 1. 32 RoCE flows and 32 TCP flows.

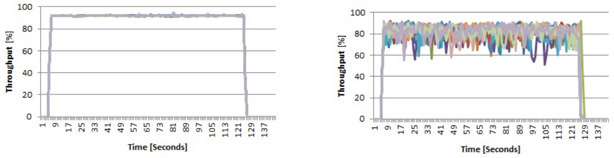


(a) Total throughput.

(b) RoCE throughput per sender.

(c) TCP throughput per sender.

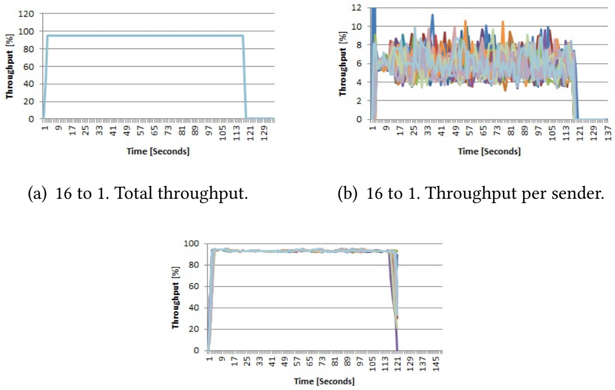
Figure 7: RoCE and TCP over lossy without isolated priorities. 16 to 1. 64 RoCE flows and 32 TCP flows.



(a) Lossless Network.

(b) Lossy Network.

Figure 8: 16 hosts all to all. 16 flows between each pair.

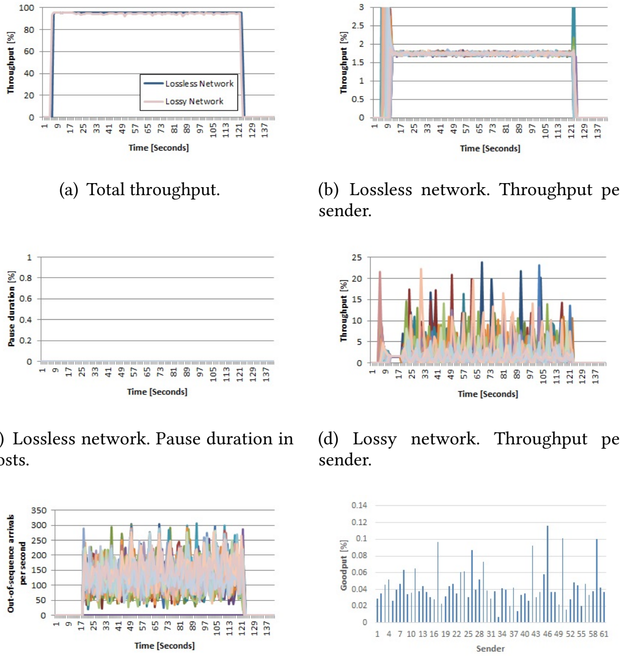


(a) 16 to 1. Total throughput.

(b) 16 to 1. Throughput per sender.

(c) 16 hosts all to all. Total Throughput.

Figure 9: Dynamic traffic.



(a) Total throughput.

(b) Lossless network. Throughput per sender.

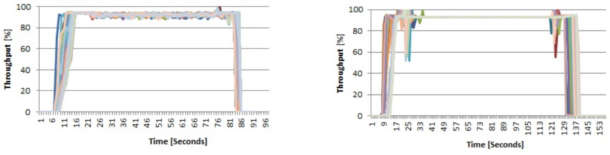
(c) Lossless network. Pause duration in hosts.

(d) Lossy network. Throughput per sender.

(e) Lossy network. Out of sequence arrivals.

(f) Lossy network. Average throughput per sender.

Figure 10: 63 to 1. 16 flows per sender.



(a) Lossless Network.

(b) Lossy Network.

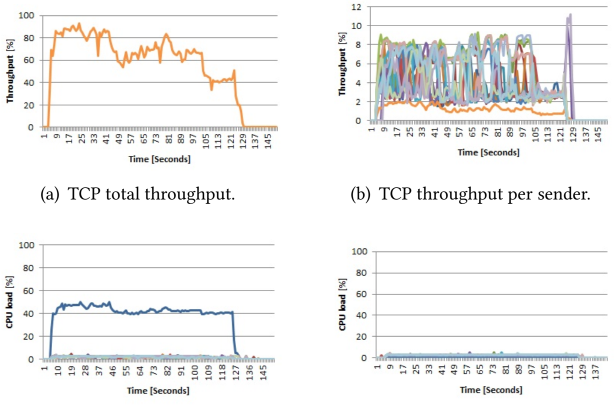
Figure 11: 64 hosts all to all. 4 flows between each pair.

In lossy network tests, despite some losses (Figure 10(e)), the total throughput was stable near the line rate (Figure 10(a)). There was slight unfairness between the senders (Figures 10(d) and 10(f)).

We then ran all-to-all traffic between the 64 hosts with 4 flows from each host to every other host. The results of the total throughput for lossless and lossy networks are presented in Figures 11(a) and 11(b), respectively. In both experiments the throughput was close to line rate with fair sharing between the hosts.

3.4 Comparison to TCP

Next, we compared the performance of RoCE to that of TCP. We repeated the same experiments as before, only instead of sending traffic over RoCE connections, we sent it over TCP using the iperf tool [7]. The results for TCP are presented in Figure 12. Figure 12(a) shows the total throughput achieved with TCP connections. TCP could not achieve full stable throughput and could not utilize the congested link as RoCE did (Figure 4(a)). Figure 12(b) shows the goodput per sending host. TCP achieved worse fairness with more fluctuations compared to RoCE (Figure 4(d)). Figure 12(c) shows the CPU usage of the hosts running TCP connections. We see that the CPU of the receiver was nearly 50% utilized from handling the connections, compared to negligible CPU usage on the hosts



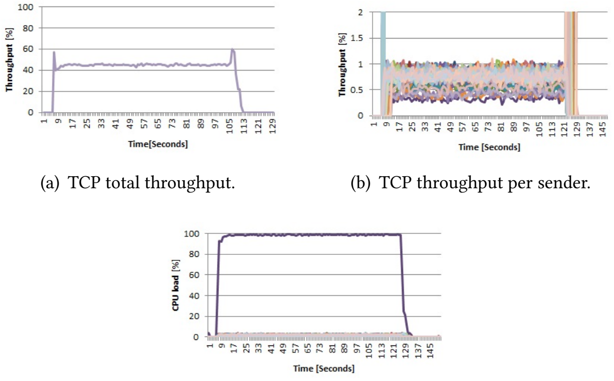
(a) TCP total throughput.

(b) TCP throughput per sender.

(c) TCP CPU load.

(d) RoCE CPU load for the same traffic pattern

Figure 12: TCP. 16 to 1. 64 flows per sender. Lossy network.



(a) TCP total throughput.

(b) TCP throughput per sender.

(c) TCP CPU load.

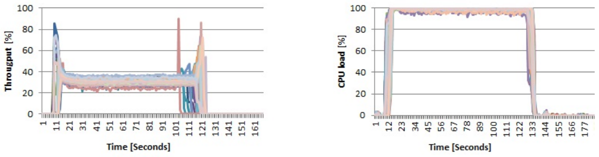
Figure 13: TCP. 63 to 1. 16 flows per sender. Lossy network.

running RoCE connections (Figure 12(d)). In fact, CPU was not used at all by the receiver of RDMA WRITE requests.

Finally, we ran TCP on a large-scale cluster of 64 hosts. Figure 13 shows the results of many-to-one traffic and Figure 14 shows the results of all-to-all traffic. In both cases, the CPUs of the receiving hosts were overloaded, and the throughput was only half of the line rate, while the CPU load reached 100%.

4 SUMMARY

Following the progress of the novel congestion control implementation in the recent Mellanox ConnectX NICs, Mellanox announced its Resilient RoCE capability, which omits the requirement of a lossless network to achieve acceptable performance, and is fully compliant with RoCEv2 specifications. In this paper, we provided evaluation of RoCE resiliency at scale. The results prove that RoCE provides solid performance even given packet drops, and significantly outperforms TCP. In addition, we provide suggestions of



(a) TCP total throughput.

(b) TCP CPU load.

Figure 14: TCP. 64 hosts all to all. 4 flows per sender. Lossy network.

network configuration to bring RoCE performance to the peak, which is still achieved over lossless network. That is intuitively true, as well; given buffer overflow, dropping packets usually has a more destructive effect than pausing the arriving traffic.

Mellanox continues to improve congestion and transport control logic and implementation in its upcoming devices, specifically, selective repeat scheme transport control to improve application throughput given the drops, and packet drop-based congestion control to remove the requirement of ECN configuration.

REFERENCES

[1] IEEE Draft Standard, 2010.. P802.1Qbb/D1.3 Virtual Bridged Local Area Networks - Amendment: Priority-based Flow Control. www.ieee802.org/1/pages/802.1bb.html. (IEEE Draft Standard, 2010.).

[2] Bryce Cronkite-Ratcliff, Aran Bergman, Shay Vargaftik, Madhusudhan Ravi, Nick McKeown, Ittai Abraham, and Isaac Keslassy. 2016. Virtualized congestion control. In *ACM SIGCOMM*. 230–243.

[3] Chuanxiong Guo, Haitao Wu, Zhong Deng, Gaurav Soni, Jianxi Ye, Jitu Padhye, and Marina Lipshteyn. 2016. RDMA over commodity ethernet at scale. In *ACM SIGCOMM 2016*. ACM, 202–215.

[4] Keqiang He, Eric Rozner, Kanak Agarwal, Yu Jason Gu, Wes Felter, John Carter, and Aditya Akella. 2016. AC/DC TCP: Virtual congestion control enforcement for datacenter networks. In *ACM SIGCOMM*. 244–257.

[5] Infiniband Trade Association 2008. *InfiniBand architecture volume 1, general specifications, release 1.2.1*. Infiniband Trade Association. www.infinibandta.org/specs.

[6] Infiniband Trade Association 2008. *Supplement to InfiniBand architecture specification volume 1 release 1.2.2 annex A17: RoCEv2 (IP routable RoCE),*. Infiniband Trade Association. www.infinibandta.org/specs.

[7] Bruce A. Mah Jeff Poskanzer Jon Dugan, Seth Elliott and Kaustubh Prabhu. 2016. *iPerf - The ultimate speed test tool for TCP, UDP and SCTP*. iperf.fr.

[8] Glenn Judd. 2015. Attaining the Promise and Avoiding the Pitfalls of TCP in the Datacenter.. In *NSDI*. 145–157.

[9] John F. Kim. 2016. *Resilient RoCE Relaxes RDMA Requirements*. www.mellanox.com/blog/2016/07/resilient-roce-relaxes-rdma-requirements/.

[10] Mirja Kühlewind, David P Wagner, Juan Manuel Reyes Espinosa, and Bob Briscoe. 2014. Using data center TCP (DCTCP) in the Internet. In *Globecom Workshops*. IEEE, 583–588.

[11] Mellanox Technologies 2013. *Mellanox ConnectX-3 Pro Product Brief*. Mellanox Technologies. www.mellanox.com/page/products_dyn?product_family=162&mtag=connectx_3_pro_en_card.

[12] Mellanox Technologies 2016. *ib write bw*. Mellanox Technologies. https://community.mellanox.com/docs/DOC-2804.

[13] Mellanox Technologies 2016. *Mellanox ConnectX-4 Product Brief*. Mellanox Technologies. www.mellanox.com/related-docs/prod_adapter_cards/PB_ConnectX-4_EN_Card.pdf.

[14] Mellanox Technologies 2016. *Mellanox SN2700 Switch System Product Brief*. Mellanox Technologies. www.mellanox.com/related-docs/prod_eth_switches/PB_SN2700.pdf.

[15] Mellanox Technologies 2017. *RoCE vs. iWARP Competitive Analysis*. Mellanox Technologies. www.mellanox.com/related-docs/whitepapers/WP_RoCE_vs_iWARP.pdf.

[16] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. 2015. Congestion control for large-scale RDMA deployments. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 523–536.