SIP (2019), vol. 8, e25, page 1 of 9 © The Authors, 2019.

doi:10.1017/ATSIP.2019.17

INDUSTRIAL TECHNOLOGY ADVANCES

Reliable multicast using remote direct memory access (RDMA) over a passive optical cross-connect fabric enhanced with wavelength division multiplexing (WDM)

KIN-WAI LEONG,[1,2] ZHILONG LI[1] AND YUNQU LEON LIU[1]

It has been well studied that reliable multicast enables consistency protocols, including Byzantine Fault Tolerant protocols, for distributed systems. However, no transport-layer reliable multicast is used today due to limitations with existing switch fabrics and transport-layer protocols. In this paper, we introduce a layer-4 (L4) transport based on remote direct memory access (RDMA) datagram to achieve reliable multicast over a shared optical medium. By connecting a cluster of networking nodes using a passive optical cross-connect fabric enhanced with wavelength division multiplexing, all messages are broadcast to all nodes. This mechanism enables consistency in a distributed system to be maintained at a low latency cost. By further utilizing RDMA datagram as the L4 protocol, we have achieved a low-enough message loss-ratio (better than one in 68 billion) to make a simple Negative Acknowledge (NACK)-based L4 multicast practical to deploy. To our knowledge, it is the first multicast architecture able to demonstrate such low message loss-ratio. Furthermore, with this reliable multicast transport, end-to-end latencies of eight microseconds or less (< 8us) have been routinely achieved using an enhanced software RDMA implementation on a variety of commodity 10G Ethernet network adapters.

Keywords: Passive optical cross-connect, Optical switch, RDMA over Converged Ethernet, Reliable multicast, Fault tolerance

Received 14 March 2019; Revised 16 September 2019

I. INTRODUCTION

Reliable multicast is an important communication primitive and building block in the architecture of scalable distributed systems. However, implementing reliable multicast at scale to-date is challenging due to limitations with existing switch fabrics and transport-layer protocols. These switch fabrics and transport-layer protocols are primarily designed for point-to-point (unicast) communications, which have insufficient permutations to support low loss-ratio multicast. So, in practice, reliable multicast communications are better off to be implemented as a software overlay on top of the unicast network.

Multicast communications consume significant resources which scale nonlinearly with the number of endpoint nodes, often requiring implementations to make trade-offs between latency, reliability guarantees, and scalability. For example, multicast applications can range diversely from live multimedia events which are broadcast to many (even millions of) subscribers in which strict reliability is not a requirement, but timeliness is, to distributed file systems in which real-time performance is not critical but data integrity is. Nonetheless, time-sensitive applications in cloud computing and other distributed systems requiring both high availability, strong consistency, and low latency at the same time are emerging, fueled by new technologies like Network Virtualization, 5G, Internet of Things (IoT), high-performance computing (HPC), and AI clusters.

We believe a reliable multicast technique with low intrinsic latency and the ability to scale could become an important building block to address the challenges posed by these time-sensitive applications. Furthermore, it could also play an important role in Byzantine Fault Tolerant protocols, which are becoming more appealing as users of data and applications are increasingly more susceptible to malicious behavior.

Even if we assume the switch fabric itself can somehow be made lossless, the networking interface and protocol stack at each of the node's memory and Central Processing Unit (CPU) still introduce packet drops. This can be due to many reasons, ranging from insufficient allocation of buffers to the processor's inability to keep up with the rate

[1]Viscore Technologies Inc, 15, Fitzgerald Road, Ottawa, K2H 9G1, Canada
[2]Rockport Networks Inc., 515 Legget Drive, Ottawa, ON K2K 3G4, Canada

Corresponding author:
Yunqu Leon Liu
Email: leon.liu@viscore.com

of packet arrival and transmission. Multicast traffic would only exacerbate these issues, as we shall explain later.

The rest of the paper is organized as follows. In Section II, we discuss the multicast challenges faced by existing switch fabrics and propose a solution based on an optical cross-connection network configured as an Optical Distributed Broadcast-Select Switch (ODBSS). In Section III, we discuss a low-loss and low latency implementation by combining ODBSS with the use of remote direct memory access (RDMA) datagram. We highlight our contributions by comparing with relevant existing works in Section IV. We conclude the paper in Section V.

II. PACKET LOSS CHALLENGES OF MULTICAST AND PROPOSED SCALABLE SOLUTIONS

In a cluster of networking nodes, packets sent out from the sender's CPU go through the transmitting protocol stack layers, traverse the switch fabric, and finally move up the receiving protocol stack layers before it reaches the receiving side's CPU. Along this path, packets could be dropped due to reasons such as traffic congestion, insufficient allocation of buffers, or blocking in the switch fabric. This could happen at many points within the sender's stack, the switch fabric, as well as the receiver's layer-2, 3, and 4 (L2, L3, and L4) buffers.

Most switch fabrics (especially for Ethernet) are not designed to be lossless even for unicast traffic. In addition, the Ethernet/IP/TCP&UDP stack was designed as best-effort, so that it does not guarantee delivery of packets. For us, to achieve a reliable multicast at a line rate of 10 Gb/s and beyond, we need the loss ratio to be lower than one in a billion. We employed a combination of an optical switch fabric and the RDMA stack to achieve this.

A) Tackling packet loss in the L1 switch fabric

Multicast communication transmits information from a single source to multiple destinations. Although it is a fundamental communication pattern in telecommunication networks as well as in scalable parallel and distributed computing systems, it is often difficult to implement efficiently using hardware at the physical layer (L1).

Building large-scale switch fabrics is challenging even for unicast (point-to-point) connections. Let us first consider an $N \times N$ switch to represent the switch fabric and consider the permutations of connections needed among inputs and outputs. For a non-blocking switch (also called perfect switch), the number of permutation assignments (maximal set of concurrent one-to-one connections) needs to be $N!$ (N factorial), with the number of cross-points scaling as N^2 (N square). When N becomes large, this crossbar switch is difficult and expensive to scale, so the switch fabric is usually implemented in a multistage switching configuration using a Clos-switch or some variation thereof.

The interconnections between the internal switch stages further increase in the number of potential congestion points that can lead to package drops. Furthermore, even though the full Clos configuration is strictly non-blocking for unicast traffic, oversubscription is often introduced in some of the switching stages for cost reasons, further increasing the probability for congestion and package loss within the switch fabric.

When used in a packet-switched context for point-to-point (unicast) traffic, a perfect switch will ensure that no packet is lost within the switch itself. Packets can still be lost outside the switch if there is congestion before or after the switch which can cause the ingress and egress buffers to overrun.

In the presence of multicast traffic, things get more challenging. In this case, the crossbar switch is no longer internally non-blocking, since the number of multicast assignments needed to support arbitrary multicast is N^N [1], which is significantly larger than $N!$ (N factorial). Furthermore, multicast traffic can exacerbate congestion issues, especially at the switch egress buffers, since packets from many sources can be directed to the same destination output port (incast).

It is not difficult to see that the number of multicast assignments needed rapidly outgrow the number of available permutation assignments, even for a relatively small port count. For example, as seen in Fig. 1, even at $N = 16$, we would need almost 900 000 times more assignments than what is available on the perfect switch.

This implies that performing multicast directly using existing switch hardware will quickly lead to blocking and loss of information, making low-loss-ratio multicast challenging, and practically impossible. It is therefore not surprising why multicast in today's distributed systems is often implemented using the software as an overlay on top of the unicast switch hardware.

To overcome the aforementioned hardware limitation, we have successfully implemented a key physical-layer (L1) building block device based on a passive optical cross-connection network (POXN) [2] by using an $N \times N$ optical coupler fabric. Optical power from each input is divided equally among the N outputs so that no reconfiguration is needed to set up a circuit between an input and an output. Since this architecture supports multicast, it can also support unicast too. However, if used primarily for unicast traffic, this architecture could be expensive.

The original POXN design was combined with a time division multiple access (TDMA) protocol. However, the POXN architecture could also be used as an optical distributed broadcast-select switch (ODBSS) as well when enhanced by wavelength division multiplexing (WDM), as shown in Fig. 2. To do so, we assign each port a dedicated optical transmitter wavelength. At each destination port end, an optical demultiplexer followed by an array of photodetectors can be used to implement the receiver function. In this way, the POXN fabric works in a distributed broadcast-and-select mode, with every port being able to

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

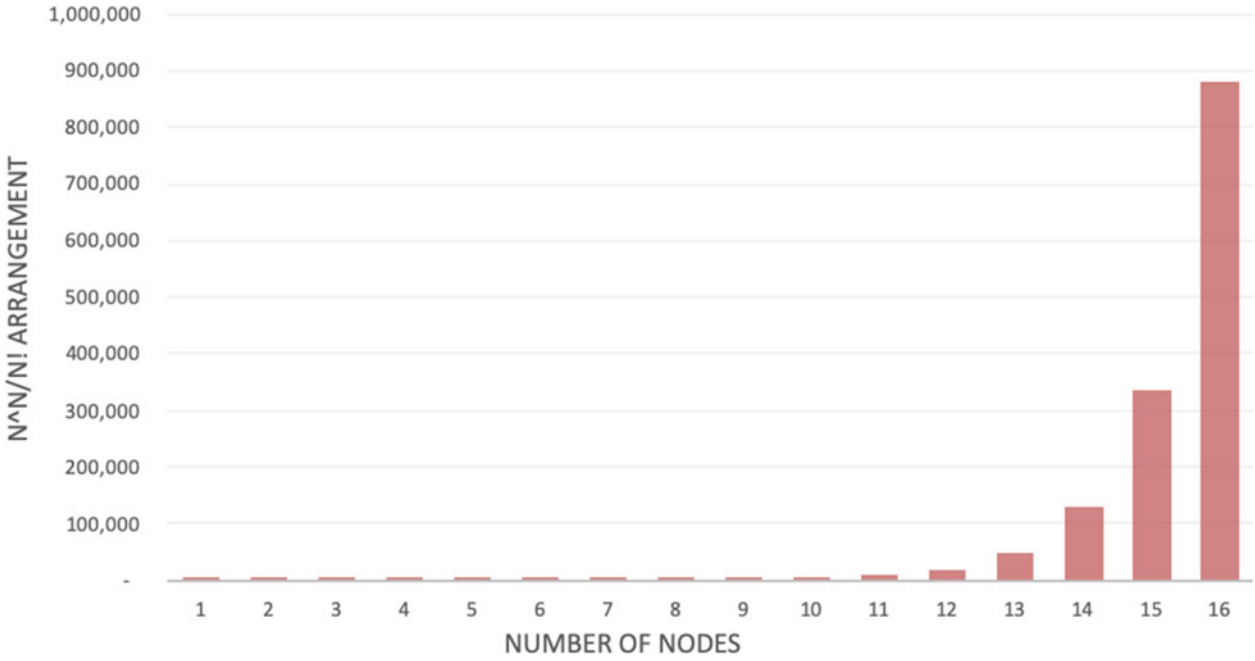

Fig. 1. The ratio of $N^{\wedge}N$ to $N!$ as a function of N.

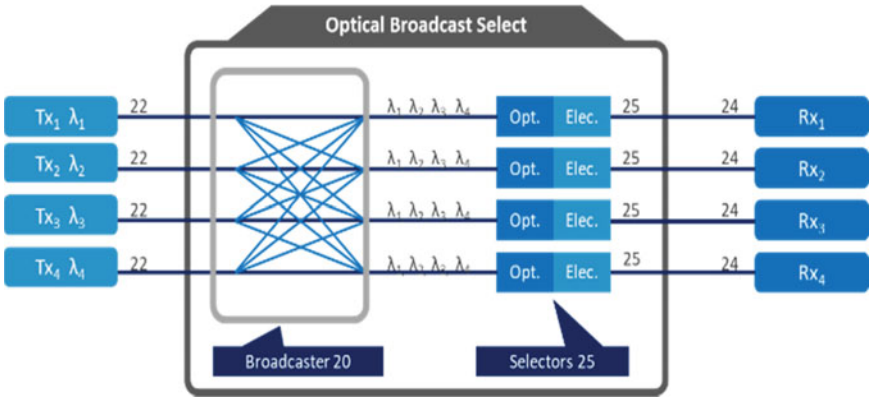

Fig. 2. Optical distributed broadcast-select switch.

broadcast to any port, and the receiving port can select the wavelength it would like to pick up.

Thanks to the wide and inexpensive bandwidth available in the optical fiber medium, this optical-based architecture can work in a distributed manner. Unlike the old-fashioned electronics-based design which has to complete the selection job within a centric switch chip, channel selection in an optical-based design can be delayed to the end-points, making it much easier to align with end-point subscription policies. This architecture has $N^{\wedge}3$ interconnections inside which can support $N^{\wedge}N$ permutations.

One familiar with switch fabric architectures would notice the similarity between an ODBSS and a crossbar with fan-out. In fact, the ODBSS design could be considered as a crossbar with full 1:N fan-out which has $N^{\wedge}N$ permutation as shown in Fig. 3. Such a switch architecture was known to offer better multicast. By being able to achieve a full fan-out, the ODBSS is capable to offer arbitrary multicast with NN permutations within.

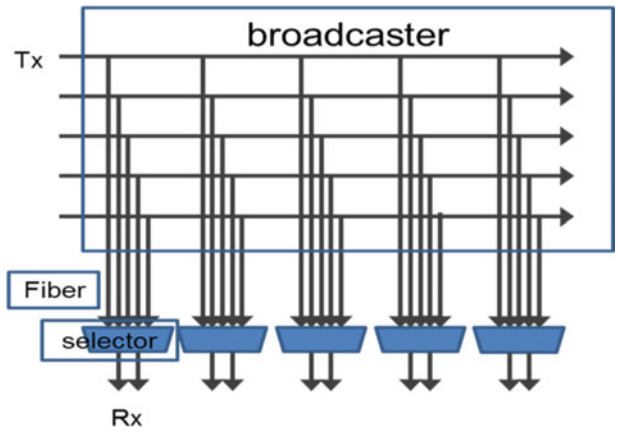

Fig. 3. 1:N fan-out crossbar view of ODBSS.

In today's widely deployed commercial optical modules, an 80 wavelength-channel system based on dense wavelength division multiplexing (DWDM) is already practical.

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

Hence, our device can support up to 80 ports using the ODBSS fabric; with a larger port count, optical amplifiers can be used within the fabric to compensate for the higher losses and maintain a suitable link budget.

The maturity of the optical component and module industry has led to dramatic cost reduction over the last two decades. Therefore, our device can be built out of cost-effective, off-the-shelf optical modules and components.

B) Tackling packet loss in receiving buffers

Buffer mis-alignment in communication stacks is another major factor for failure to achieve low loss-ratio multicast. It could happen in different layers that refer to memory buffer allocation actions.

To deliver the message to processes (CPU), a reliable receiving mechanism is required. In standard transmission control protocol/internet protocol (TCP/IP) architecture, reliable delivery is guaranteed by layer 4 protocol TCP. Despite its ability to ensure lossless delivery for unicast traffic, TCP cannot be used as an L4 protocol for multicast because as a connection-based protocol, TCP has no mechanism to handle one-to-many connections. On the other hand, with user datagram protocol (UDP), a multicast over IP (L3) is practical, but the delivery reliability is never guaranteed. Furthermore, due to the standard protocol stack implementation on the Linux platform, the kernel would allocate socket buffer for each ethernet frame received and copy payload from kernel space to user space applications after. This could amplify buffer mis-alignment problems and trigger a high loss rate in the upper layer protocols. When we measured UDP loss over a good one-to-one physical connection, the loss-ratio we obtained was as high as 20% initially. With careful fine-tuning of the kernel buffer and traffic load, the loss ratio can be improved but is still often beyond 1%.

Ideally, a message-based L4 protocol with pre-allocated buffers for receiving messages and working in tandem with a lossless ODBSS architecture in L1 would be appropriate for a low-loss multicast system. Based on this understanding, we explored RDMA, which is a protocol developed for HPC. In RDMA specifications, two datagram-based queue pair types, namely reliable datagram (RD) and unreliable datagram (UD), could potentially be used for multicast. However, among all the known RDMA implementations today, none of them supports RD and some of them do not support multicast at all. This is not surprising and is likely due to the lack of a powerful switch that can support low loss-ratio multicast.

InfiniBand, RDMA over converged ethernet (RoCE) and internet wide area RDMA protocol (iWARP) are the three major implementations of RDMA commonly used in the industry. Among them the best-known implementation is InfiniBand. RoCE, which leverage the low-cost and ubiquitous IP/Ethernet ecosystem, is now being deployed in datacenters.

We employ RDMA datagram (UD) transport, which has a pre-allocated resource on both the sender and receiver sides. In our proof-of-concept work, we experimented with RoCE hardware-based network interface card (NICs) from different vendors. Using these, we were able to achieve a multicast loss ratio level of the order of one per million in our laboratory, which was much better than what is possible with UDP. However, without access to the internal hardware/firmware, we were not able to determine if this could be further improved. Therefore, we turned to Soft-RoCE (http://www.roceinitiative.org/software-based-roce-a-new-way-to-experience-rdma/), which is an open-source software implementation of RoCE. With some debugging and improvement of the software, we were able to get the multicast datagram feature to work successfully; in doing so, we succeeded in sending over 68 billion multicast packages through our prototype POXN fabric without any packet loss.

Using a Perftest (https://github.com/linux-rdma/perftest/tree/master/src) package, we performed message latency benchmarking tests using two different RoCE hardware NICs (Mellanox and Emulex), comparing the hardware RoCE performance with Viscore-improved Soft-RoCE, as well as the open-source Soft-RoCE. We carried out latency testing using both RDMA datagram and RDMA reliable connection (RC). Since the RDMA datagram size is limited by the Maximum Transmission Unit (MTU) (which is 4096 bytes), we used RDMA RC to extend the testing to larger messages, as shown in Fig. 4. With some bug fixes and software optimization, we were able to achieve better performance than open-source Soft-RoCE, by improving latency and throughput performance of Soft-RoCE by 2X and 3X, respectively.

C) Scaling the multicast in multidimensions

For larger port counts, one can leverage a multidimensional approach, as shown in Fig. 5, to scale the network to $N^{\wedge}D$ ports, in which D is the number of dimensions, and N is the number of nodes within a dimension. When data packets move from one dimension to another, they go through an optical-to-electrical-to-optical (OEO) conversion. This enables optical wavelengths to be re-used in different dimensions, facilitating the ability to scale. For example, a three-dimensional system based on 40 wavelengths can support up to $40 \times 40 \times 40 = 64$K ports. Similarly, an 80-port ODBSS can potentially scale up to 512K ports. The multidimensional approach to network scaling and its routing and control are well studied in direct networks such as Torus, Hypercube, and B-Cube [3].

It should be noted that, in the multidimension scaling method, the nodes in between dimensions are filtering the multicast packets to its sub-nodes. If over-subscription happens, then these nodes will be exposed to the risk of higher ratio packet loss. Therefore, when designing upper-layer protocols, one should bear this in mind to carefully control the over-subscription policy.

Nevertheless, since the ODBSS works in a distributed manner, any over-subscription only affects the end-nodes, not the fabric in between, thus limiting the loss risk to

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

	Viscore SoftRoCE			Open Source SoftRoCE			Hardware RoCE	
#bytes	Emulex_VT	Ether_VT	Mlnx_VT	Emulex_Open	Ether_Open	Mlnx_Open	Emulex_HW	Mlnx_HW
2	10.47	4.31	4.68	15.01	10.43	7.48	6.45	1.55
4	10.46	4.32	4.59	15.04	9.78	7.46	6.43	1.54
8	10.33	4.68	4.6	15.22	9.96	7.44	6.43	1.54
16	10.29	4.31	4.6	15.15	9.87	7.44	6.43	1.56
32	10.22	4.33	4.59	15.18	9.8	7.45	6.44	1.57
64	10.55	4.90	4.94	15.21	9.43	7.4	6.45	1.66
128	10.54	4.40	5.45	15.36	9.85	7.56	6.47	1.82
256	10.73	4.49	5.66	15.39	9.97	8.05	7.09	2.56
512	11.71	5.21	6.22	15.47	16.26	7.69	7.38	2.89
1024	13.17	10.92	6.87	15.48	13.44	8.22	7.98	3.59
2048	14.98	11.68	7.77	18.31	16.07	11.08	10.29	4.97
4096	20.89	14.18	10.02	23.79	21.43	14.1	12.38	6.71
8192	35.75	15.75	13.79	34.27	40.37	22.13	16.56	10.16
16384	69.32	21.30	20.21	57.39	67.07	37.45	22.77	17.03
32768	110.83	42.14	37.2	104.25	124.44	68.35	37.05	30.74
65536	164.49	65.75	68.25	198.04	238.96	142.51	65.51	58.18
8388608	19019.35	7352.47	8136.81	27304.29	29230.87	15959.76	7317.24	7029.42

Fig. 4. Software RoCE latency measurement.

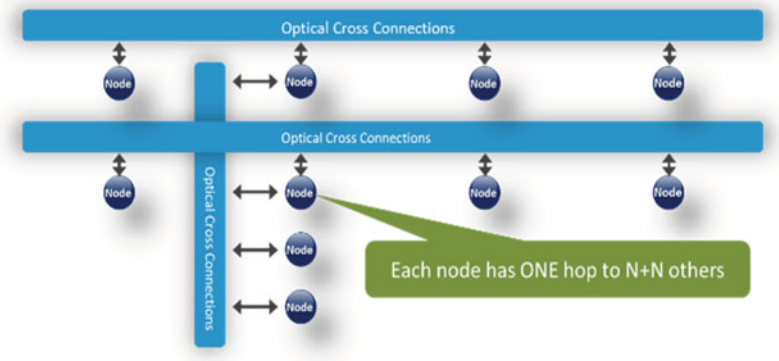

Fig. 5. Scale-out in multidimensions.

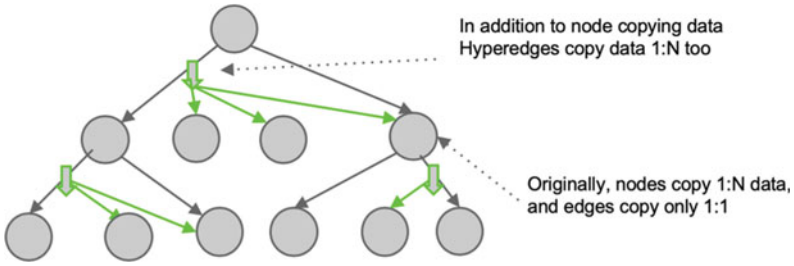

Fig. 6. The logical view of multicast enhancement.

within a subnet or the end-nodes alone. This is in contrast to a centric switch-based architecture, in which there is a well-known risk of broadcast storms that affect the entire network [4].

D) Enhancing multicast in arbitrary network topology

More generally, the ODBSS can be used to enhance multicast in an arbitrary network topology, e.g. multistage network and torus.

Without dwelling into any specific network topology for the moment and treating it as a black-box, multicast in the network abstraction can be described logically as a 1:*N* tree, per Fig. 6. In this view, the ODBSS could be added into the multicast tree arbitrarily at any point to enhance the multicast function, by helping to reduce the depth of the tree, which in turn reduce the overall load and latency of the entire multicast process.

As a specific example, let us consider the physical deployment view using a Spine-Leaf ToR network topology since it is commonly used in datacenters. In this case, the ODBSS could be added either within a rack or in between racks to enhance multicast. Figure 7 shows the case in which ODBSS is added in every rack to offload multicast traffic from the Spine-Leaf-ToR. This enables all 32 servers in a rack to receive the multicast simultaneously, hence reducing overall load and latency. In a cluster of 1024 (32×32) nodes, up to

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

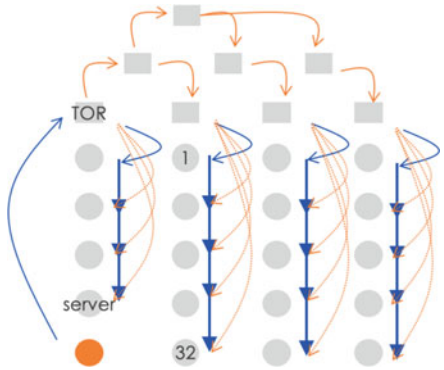

Fig. 7. The physical implementation view of multicast enhancement.

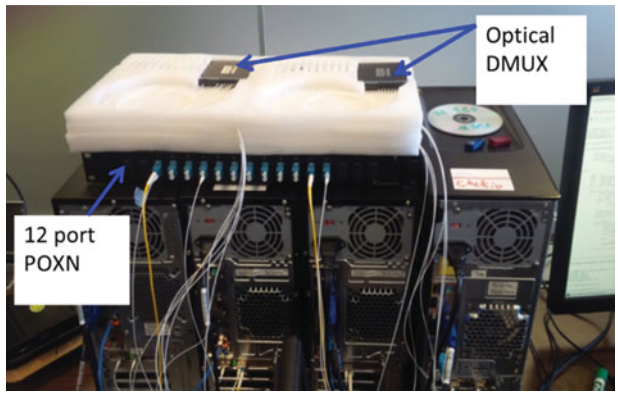

Fig. 8. Proof-of-concept test-bed setup.

96.8% multicast traffic is offloaded from the network itself to the deployed ODBSS.

Interestingly, one could consider this hybrid deployment approach as an alternative scaling method for ODBSS multicast too, in addition to the multidimensional approach described above. It should be emphasized that the process of constructing the multicast tree and implementing the associated routing is a significant challenge for any scaling method employed. We will address this topic in further details in a future paper.

III. LOW LATENCY AND LOW LOSS IMPLEMENTATION

A) Implementation and proof-of-concept test-bed setup

We built our proof-of-concept test-bed (as shown in Fig. 8) using four computer nodes connected together by a 12-port POXN module. Off-the-shelf DWDM 10Gb/s SFP+ transceivers and optical de-multiplexers were used to complete an ODBSS implementation for the four nodes. With this setup, we then tested RDMA UD multicast over IP/Ethernet multicast addressing with several RoCE hardware implementations and software RoCE implementation.

It is interesting to note that our setup actually provided several unique advantages when it comes to being able to push the loss ratio as low as possible. First of all, if one has already reached a loss ratio that is lower than one in a million using a setup involving an electronic switch, it would be hard to determine if the loss is happening in the switch or in the NIC itself. With our ODBSS architecture, we are confident that if a packet is lost, it could only happen in the transmitting or receiving ports or the buffers which are aligned with them. Since we have more than one receiving port, if the transmitting side loses the packet, all receiving sides should lose that packet. This rather simple feature was of great help in de-bugging and identifying the root cause of packet loss (Fig. 8).

Second, using a software RoCE implementation actually enabled us to debug more effectively for several reasons: (a) the implementation is more transparent to us since we have access to the source code; (b) we can tag the packets and messages as needed for de-bugging purposes, and (c) we can easily fix bugs when we identify them. We started out testing with hardware RoCE implementations, but when we encountered packet loss, we could not make further progress until we switched to a software implementation. The packet loss observed with the hardware RoCE NICs does not necessarily imply that there are bugs in the hardware implementation itself, but we could not pursue its root cause given the proprietary nature of the hardware implementation.

After we pushed the loss ratio to less than one in a hundred million, some unexpected bugs started to show up that could only be identified and fixed in the test-bed described above. For instance, after such a large number of packets are continuously sent out, the Packet Serial Number (PSN) can become larger than its range and needs to be reset. Although this procedure is well defined and documented, it turned out that the related algorithm in the Soft-RoCE C code was not completed to cover this rare case, which does not happen often unless a very large number of UD packets is sent. We do not know if the hardware implementations cover such rare cases with a very large number of UD packets, but it will not be a big surprise if they did not.

Last but not least, the practical know-how of building passive optical cross-connects with inexpensive optical components made this implementation economically feasible for us. The interdisciplinary nature of this work can be seen in how this optical hardware played a key role in helping to improve the low-loss performance of RoCE, which in turn helps bring out the full multicast potential of this optical hardware.

B) Low latency and low loss ratio

It is instructive to do a quick comparison of the achievable latency performance with ODBSS-based RDMA multicast versus that of overlay multicast and other hardware (i.e. switch-based) multicast.

A good example of a high-performance overlay multicast is based on Binomial tree implementation [5]. A classic binomial multicast tree is shown in Fig. 9.

The overlay binomial multicast latency can be thought of as Latency $= K(\mathrm{Log2}(N)) \times L$ in which L is the unicast

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

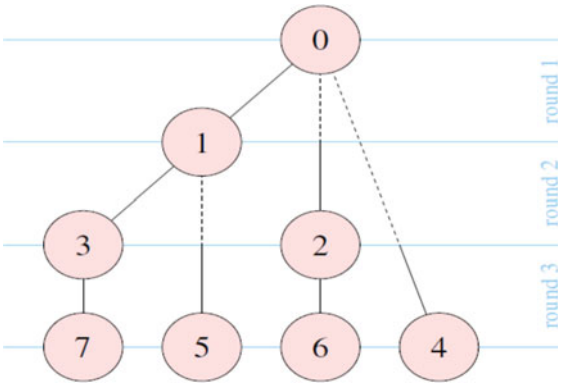

Fig. 9. Classic binomial multicast [5].

latency, N is the node count, and K is a weighting factor which is dependent on how long each node has to wait to complete its task (and can, therefore, increase nonlinearly with N). At first glance, the latency of binomial overlay multicast does not grow that fast with the node count because the binomial algorithm builds chains with length of $\log_2(N)$.

However, measurements [5] showed that the latency of binomial multicast grows nonlinearly with node count. This is due to two factors in the overlay implementation; one is related to the long tail in unicast latency being much larger (35 us versus 3 us) than that of the average latency; the other is related to nodes on the chain needing to wait for the previous one to send them a packet before they can send. Therefore, the latency of chain(s) in the binomial tree is vulnerable to the statistical nature of traffic in a network; these statistical fluctuations only worsen with extra traffic burden introduced by the binomial algorithm.

Hardware (i.e. switch-based) multicast, e.g. IP multicast or InfiniBand multicast, in principle, should have better latency than overlay multicast. For example, the latency of hardware-multicast-based algorithms was shown to outperform that of binomial overlay multicast in [5]. However, InfiniBand multicast (as well as IP multicast) is lossy, which limits its potential use.

Unlike InfiniBand hardware multicast, the loss ratio of RDMA multicast over ODBSS is very low. In our test-bed demonstration, we have pushed the loss ratio to be as low as one in 68 billion packets. With ODBSS, if we stay within one dimension, the multicast latency is comparable to the unicast latency. When we scale using multidimensions, the increase in multicast latency is weighted by the number of dimensions, rather than by N (the number of nodes). As N increases, the multicast latency advantage grows nonlinearly when compared to overlay multicast latency.

It is worthwhile to note that incast and the oversubscription management is always a challenge for all multicast. However, the proposed ODBSS architecture has advantages for incast traffic because the selection happens at the endpoint. Even if one node is over-subscribed, it only affects that one particular node, but neither the ODBSS fabric, the sender, nor the other receiving nodes are impacted.

C) Enabling low latency reliable multicast

The low-latency low-loss-ratio optical multicast we have described could become an important toolset for protocol designers who need a low-latency reliable multicast to implement consistency protocols. Given the very low loss ratio we observed in our laboratory for optical multicast, we believe it is practical to build a simple NACK-based reliable multicast transport over ODBSS and RDMA Datagram.

As an example, Byzantine fault tolerance consistency protocols, e.g. Rampart [6] are built using reliable multicast, so it is conceivable that such protocols could potentially benefit from an intrinsically low-latency reliable multicast. A low latency consistency protocol could shorten the time window available for traitorous processes to attack by enabling a distributed system to achieve consistency faster. Furthermore, traitorous processes would have their own consistency challenge if they need to collaborate among themselves using a secret communication channel, especially if their channel lacks this low latency advantage.

IV. RELATED WORK

In this section, we highlight our contributions by comparing our work with other existing ones. Our main contribution is proposing a scalable low-latency, low loss-ratio transport-layer multicast solution by combining the benefits of an optical cross-connect fabric (ODBSS) with RDMA. This combination, in turn, simplifies low-latency reliable multicast implementation.

Several previous works have investigated the use of optical couplers (1:N or $N \times N$) to build an optical switch fabric [2], handle multicast traffic [7], and to demonstrate reliable optical multicast [8].

In [2], a passive optical cross-connection network was used as a switch fabric through a TDMA implementation. We extended this initial passive optical cross-connection idea by introducing WDM and proposing an ODBSS architecture.

In [7] the authors described integrating $1 \times N$ passive optical splitters in a hybrid network architecture (combining optical circuit switching with electronic packet switching) to simplify the delivery of multicast traffic flows. Performance comparisons were made with other types of multicast (IP Multicast, Overlay Multicast, and Peer-to-Peer Multicast) using Unreliable Datagram Protocol (UDP) datagrams. RDMA was not used or considered in their approach. The authors also studied Ring Paxos over IP multicast with their architecture. However, IP-multicast is subject to message losses, mostly due to network congestion and buffer overflow (i.e. when the receiver is not able to consume messages at the rate they are transmitted). Ring Paxos can cope with message loss and therefore benefit from the throughput that IP-multicast can provide without falling prey to its shortcomings. It would be interesting to

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

investigate if such atomic multicast protocols can leverage a low-latency reliable multicast as described in our work.

In [8] an implementation of reliable optical multicast is described using a hybrid scheme in which an optical circuit switching network directs multicast traffic to a $1 \times N$ optical splitter, while a separate electronic packet switching network is used for NACK control. In our approach, a separate electronic packet switching network is not needed, and the very low loss ratio achievable can simplify NACK control and improve latency.

Software overlay multicast over InfiniBand and RoCE have been studied, e.g. RDMC [9], but these are performed over existing unicast switch fabrics. Since our proposed multicast has lower latency but not necessarily bandwidth or throughput advantage compared to such overlay techniques, it may be interesting to consider combining these multicast techniques to achieve optimal overall performance.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new optical architecture working in tandem with RDMA to offer an intrinsically low-latency and low loss-ratio multicast channel. Building upon this, a reliable multicast protocol is proposed to deliver a reliable, low-latency, and scalable multicast service to distributed systems. By offloading multicast traffic, these reliable low-latency multicast service also improve the unicast performance of existing switch fabrics.

Within a subnet, this optical hardware offers intrinsic ordering in the physical layer. Also, RDMA maintains ordering within a message. These features would be an interesting research topic for researchers of distributed consistency to explore further. Such low-latency reliable multicast services may enable new ways to optimize the design of distributed systems.

In the next step, the team will study deployment scenarios to investigate how the potential value offered by ODBSS-based RDMA could be best demonstrated in Datacentre, High Performance Computing (HPC), and Artificial Intelligence (AI) clusters. As part of this investigation, we shall address the challenge of constructing a multicast tree and implementing routing as the size of cluster scales. We will also investigate the use of low-latency reliable multicast enabled by our technology for fast data replication services, including Pub/Sub services and distributed lock services, especially in use cases with fast NVMeOF (Non-Volatile Memory Express Over Fabric) storage. Additionally, as mentioned above, RD) is currently not supported by the RDMA implementations we have tested, primarily because of the $N\hat{}N$ to $N!$ issue alluded to earlier. This makes it extremely hard to perform non-blocking broadcast in modern electrical packet switching systems. However, the proposed ODBSS gives us a chance to overcome this obstacle. Hence, exploring the feasibility and benefits of implementing RD over the ODBSS architecture becomes a very attractive and worthy topic in our future work.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Wenda Ni of Microsoft Corporation, Professor Ken Birman of Cornell University, and Professor Changcheng Huang of Carleton University for useful discussions in the initial phase of this work. They also thank the reviewers for their valuable feedback and suggestions.

FINANCIAL SUPPORT

This work is funded by Viscore Technologies Inc, with additional financial support from Ontario Centre of Excellence and NRC-IRAP. All IPR are owned by Viscore and patent pending.

STATEMENT OF INTEREST

None.

REFERENCES

[1] Yuanyuan Y; Gerald M.M.: Nonblocking broadcast switching networks. *IEEE Trans. Comput.*, **40** (9) (1991), 1005–1015.

[2] Wenda N.; Changcheng H.; Yunqu L.L.; Weiwei L.; Kin-Wai L.; Jing W.: POXN: a new passive optical cross-connection network for low-cost power-efficient datacenters. *J. Lightwave Technol.*, **32** (8) (2014), 1482–1500.

[3] Chuanxiong G. *et al.* BCube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.* **39** (4) (2009), 63–74.

[4] Krzysztof O.; Ken B.; Danny D.: Quicksilver scalable multicast (qsm), in *Seventh IEEE International Symposium on Network Computing and Applications*, 2008, 9–18.

[5] Torsten H.; Christian S.; Wolfgang R.: A practically constant-time MPI Broadcast Algorithm for large-scale InfiniBand Clusters with Multicast, in *21th International Parallel and Distributed Processing Symposium (IPDPS 2007)*, March 2007, 26–30.

[6] Michael K.R.: Secure agreement protocols: reliable and atomic group multicast in Rampart, in *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, 1994, 68–80.

[7] Payman S.; Varun G.; Junjie X.; Howard W., Gil; Z.; Keren B.: Optical multicast system for data center networks. *Opt. Express*, **23** (17) (2015), 22162–22180.

[8] Payman S.; Junjie X.; Keren B.: Experimental demonstration of one-to-many virtual machine migration by reliable optical multicast, in *European Conference on Optical Communication (ECOC)*, 2015, 1–3.

[9] Jonathan B.; Sagar J.; Ken B.; Edward T.: RDMC: a reliable RDMA multicast for large objects, in *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, 71–82.

Kin-Wai Leong received the combined S.B. and S.M. degrees in electrical engineering and computer science, and the Ph.D. degree in quantum electronics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1985 and 1990, respectively. He is a Co-Founder of Viscore Technologies Inc., Ottawa, ON, Canada and has led its technology development in his role as Senior Vice

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press

President of Research and Development since joining in 2008. He was previously a Senior Manager at JDS Uniphase, where he led the development of advanced passive optical products, and Manager of Product Development at Bell-Northern Research, where he managed the development of several generations of DFB lasers deployed in optical systems by Nortel Networks.

Zhilong Li received his B. Eng. and M.A.Sc. degree from Northwestern Polytechnical University, Xi'an, China in 2012 and 2015, respectively. And he received his M.Eng degree in 2017 from the department system and computer engineering at Carleton University, Ottawa, Canada. He is currently a Senior Software Engineer in Viscore Technologies Inc., Ottawa, ON, Canada.

Yunqu Leon Liu received the B.E. degree in fluid dynamic control from the Huazhong University of Science and Technology, Wuhan, China, in 1993, and the MBA degree from Cornell University and Queen's University in 2011. He is currently a Founder and President of Viscore Technologies Inc. Ottawa, ON, Canada. Before that, he was a Senior Engineer and Senior Manager in Nortel Networks and Huawei Technologies, respectively. He has invented several patents in software fault detective, optical modules and networking. His research interests include optical networking for distributed systems, e.g. Cloud, HPC and AI clusters, especially the asymmetric communication, e.g. multicast and incast, in these systems.

https://doi.org/10.1017/ATSIP.2019.17 Published online by Cambridge University Press