Optical multicast system for data center networks

Payman Samadi,*[*] Varun Gupta, Junjie Xu, Howard Wang, Gil Zussman, and Keren Bergman

Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

[*]*ps2805@columbia.edu*

Abstract: We present the design and experimental evaluation of an Optical Multicast System for Data Center Networks, a hardware-software system architecture that uniquely integrates passive optical splitters in a hybrid network architecture for faster and simpler delivery of multicast traffic flows. An application-driven control plane manages the integrated optical and electronic switched traffic routing in the data plane layer. The control plane includes a resource allocation algorithm to optimally assign optical splitters to the flows. The hardware architecture is built on a hybrid network with both Electronic Packet Switching (EPS) and Optical Circuit Switching (OCS) networks to aggregate Top-of-Rack switches. The OCS is also the connectivity substrate of splitters to the optical network. The optical multicast system implementation requires only commodity optical components. We built a prototype and developed a simulation environment to evaluate the performance of the system for bulk multicasting. Experimental and numerical results show simultaneous delivery of multicast flows to all receivers with steady throughput. Compared to IP multicast that is the electronic counterpart, optical multicast performs with less protocol complexity and reduced energy consumption. Compared to peer-to-peer multicast methods, it achieves at minimum an order of magnitude higher throughput for flows under 250 MB with significantly less connection overheads. Furthermore, for delivering 20 TB of data containing only 15% multicast flows, it reduces the total delivery energy consumption by 50% and improves latency by 55% compared to a data center with a sole non-blocking EPS network.

OCIS codes: (060.4250) Networks; (060.4255) Networks, multicast; (060.4510) Optical communications; (060.4253) Networks, circuit-switched; (200.4650) Optical interconnects.

References and links

1. D. Kilper, K. Bergman, V. W. Chan, I. Monga, G. Porter, and K. Rauschenbach, "Optical networks come of age," Opt. Photon. News **25**(9), 50–57 (2014).
2. "Cisco global cloud index: Forecast and methodology: 2013–2018," `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf`.
3. J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM **51**(1), 107–113 (2008).
4. L. Lamport, "The part-time parliament," ACM Trans. Comput. Syst. **16**(2), 133–169 (1998).
5. S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google File System," SIGOPS Oper. Syst. Rev. **37**(5), 29–43 (2003).

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015

6. M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Proceedings of the ACM Symposium on Operating Systems Design and Implementation* (ACM, 2006), pp. 335–350.
7. S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proceedings of the ACM Symposium on Operating Systems Design and Implementation* (ACM, 2006), pp. 307–320.
8. D. Borthakur, "The Hadoop Distributed File System: Architecture and design," (2007), `http://hadoop.apache.org/core/docs/current/hdfs design.pdf`.
9. W. Mach and E. Schikuta, "Parallel database join operations in heterogeneous grids," in *Proceedings of IEEE International Conference on Parallel and Distributed Computing* (IEEE, 2007), pp. 236–243.
10. M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Managing data transfers in computer clusters with orchestra," SIGCOMM Comput. Commun. Rev. **41**(4), 98–109 (2009).
11. "Apache spark: Lightning-fast cluster computing," `https://spark.apache.org`.
12. R. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments," in *Proceedings of IEEE International Conference on Parallel Processing* (IEEE, 2011), pp. 295–304.
13. "Twitter murder: Fast data center code deploy using bittorrent," `http://engineering.twitter.com/2010/07/murder-fast-datacenter-code-deploys.html`.
14. B. Quinn and K. Almeroth, "IP multicast applications: Challenges and solutions," IETF, Internet Draft,(2001).
15. H. McBride and H. Liu, "Multicast in the data center overview," IETF, Internet Draft (2012).
16. C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," IEEE Netw. **14**(1), 78–88 (2000).
17. X. Li and M. J. Freedman, "Scaling IP multicast on datacenter topologies," in *Proceedings of the ACM Conference on Emerging Networking Experiments and Technologies* (ACM, 2013), pp. 61–72.
18. D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE **103**(1), 14–76 (2015).
19. P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," SIGCOMM Comput. Commun. Rev. **39**(4), 195–206 (2009).
20. D. Li, Y. Li, J. Wu, S. Su, and J. Yu, "ESM: efficient and scalable data center multicast routing," ACM Trans. Netw. **20**(3), 944–955 (2012).
21. J. Cao, C. Guo, G. Lu, Y. Xiong, Y. Zheng, Y. Zhang, Y. Zhu, and C. Chen, "Datacast: a scalable and efficient reliable group data delivery service for data centers," IEEE J. Sel. Areas Commun. **31**(12), 2632–2645 (2012).
22. B. Cohen, "The bittorrent protocol specification," (2008), `http://bittorrent.org/index.html`.
23. A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," Commun. ACM **54**(3), 95–104 (2011).
24. M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," SIGCOMM Comput. Commun. Rev. **38**(4), 63–74 (2008).
25. J. Bowers, "Low power 3D MEMS optical switches," in *Proceedings of IEEE Conference on Optical MEMS and Nanophotonics* (IEEE, 2009), pp. 152–153.
26. N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," SIGCOMM Comput. Commun. Rev. **41**(4), 339–350 (2011).
27. G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," SIGCOMM Comput. Commun. Rev., **41**(4), 327–338 (2011).
28. H. Wang, Y. Xia, K. Bergman, T. E. Ng, S. Sahu, and K. Sripanidkulchai, "Rethinking the physical layer of data center networks of the next decade: Using optics to enable efficient *-cast connectivity," SIGCOMM Comput. Commun. Rev., **43**(3), 52–58 (2013).
29. P. Samadi, V. Gupta, B. Birand, H. Wang, G. Zussman, and K. Bergman. "Accelerating incast and multicast traffic delivery for data-intensive applications using physical layer optics," in *Proceedings of the ACM Conference on SIGCOMM* (ACM, 2014), pp. 373-374.
30. G.N. Rouskas. "Optical layer multicast: rationale, building blocks, and challenges," IEEE Network **17**(1), 60–65 (2013).
31. L.H. Sahasrabuddhe, B. Mukherjee. "Light trees: optical multicasting for improved performance in wavelength routed networks," IEEE Commun. Mag. **37**(2), 67–73 (1999).
32. N. Sambo, G. Meloni, G. Berrettini, F. Paolucci, A. Malacarne, A. Bogoni, F. Cugini, L. Poti, P. Castoldi. "Demonstration of data and control plane for optical multicast at 100 and 200 Gb/s with and without frequency conversion," IEEE J. Opt. Commun. Netw. **5**(7), 667-676 (2013).
33. B. Birand, H. Wang, K. Bergman, and G. Zussman, "Measurements-based power control - a cross-layered framework," in *Optical Fiber Communications Conference*, OSA Technical Digest (CD) (Optical Society of America, 2013), paper JTh2A.66.
34. C. Lai, D. Brunina, B. Buckley, C. Ware, W. Zhang, A. Garg, B. Jalali, and K. Bergman, "First demonstration of a cross-layer enabled network node," J. Lightwave Technol. **31**(9), 1512–1525 (2013).

35. “Calient S320 optical circuit switching,” http://www.calient.com.
36. “Polatis series 6000,” http://www.polatis.com.
37. “PhotonDesign: A 1×8 planar MMI coupler,” http://www.photond.com/products/fimmprop/fimmprop_applications_03.htm.
38. A. Sugita, A. Kaneko, and M. Itoh, “Planar lightwave circuit,” United States Patent 6304706 (October 16, 2001).
39. “PLC optical splitter,” http://www.fiberstore.com/1x64-fiber-plc-splitter-with-fan-out-kits-p-11606.html.
40. Y. Luo, X. Zhou, F. Effenberger, X. Yan, G. Peng, Y. Qian, and Y. Ma, “Time- and wavelength-division multiplexed passive optical network (TWDM-PON) for next-generation PON stage 2 (NG-PON2),” J. Lightwave Technol. **31**(4), 587–593 (2013).
41. G. Keiser, *FTTX concepts and applications* (John Wiley and Sons, 2006).
42. X. Ma and G.-S. Kuo, “Optical switching technology comparison: optical MEMS vs. other technologies,” IEEE Commun. Mag. **41**(11), 16–23 (2003).
43. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: Enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., **38**(2), 69–74 2008.
44. F. Balus, M. Pisica, N. Bitar, W. Henderickx, and D. Stiliadis, “Software driven networks: Use cases and framework,” IETF, Internet Draft (2011).
45. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, “B4: Experience with a globally-deployed software defined WAN,” SIGCOMM Comput. Commun. Rev. **43**(4), 3–14 (2013).
46. C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. “Achieving high utilization with software-driven WAN,” SIGCOMM Comput. Commun. Rev. **43**(4), 15–26 (2013).
47. T. White, *Hadoop: the definitive guide* (O’Reilly Media Inc., 2009).
48. A. Fréville, “The multidimensional 0–1 knapsack problem: An overview,” Eur. J. Oper. Res. **155**(1), 1–21 (2004).
49. S. Sanfilippo and P. Noordhuis, “Redis,” http://redis.io.
50. M. J. Connelly, *Semiconductor Optical Amplifiers* (Springer, 2002).
51. Y. Li and L. Tong, “Mach-Zehnder interferometers assembled with optical microfibers or nanofibers,” Opt. Lett. **33**(4), 303–305 (2008).
52. J. Kim, C. J. Nuzman, B. Kumar, D. F. Lieuwen, J. S. Kraus, A. Weiss, C. P. Lichtenwalner, A. R. Papazian, R. E. Frahm, and J. V. Gates, “1100 × 1100 port MEMS-based optical crossconnect with 4-dB maximum loss,” IEEE Photon. Technol. Let. **15**(11), 1537–1539 (2003).
53. “Iperf,” https://iperf.fr/.
54. “JGroups,” http://www.jgroups.org/.
55. “NACK-Oriented Reliable Multicast (NORM),” http://www.nrl.navy.mil/itd/ncs/products/norm.
56. P. Samadi, J. Xu, K. Bergman, “Experimental demonstration of one-to-many virtual machine migration by reliable optical multicast,” in *European Conference and Exhibition on Optical Communication*, OSA Technical Digest Series (Optical Society of America, 2015), paper 689.
57. “Herd,” https://github.com/russss/Herd.
58. “D-ITG: distributed internet traffic generator,” http://traffic.comics.unina.it/software/ITG/.
59. “NS3,” http://www.nsnam.org/.
60. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Parveen, “The cost of a cloud: research problems in data center networks,” SIGCOMM Comput. Commun. Rev., **39**(1), 68–73 (2009).
61. P. Marandi, M. Primi, N. Schiper, and F. Pedone, “Ring paxos: A high-throughput atomic broadcast protocol,” in *Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks* (IEEE, 2010), pp. 527–536.
62. R. Levy, *The complexity of reliable distributed storage* (PhD Thesis, EPFL 2008).
63. Y. Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarfella, “The totem single-ring ordering and membership protocol,” ACM Trans. Comput. Syst. **13**(4), 311–342 (1995).
64. H. Takahashi, S. Suzuki, K. Kato, and I. Nishi, “Arrayed-waveguide grating for wavelength division multi/demultiplexer with nanometre resolution,” IEEE Electron. Lett., **26**(2), 87–88 (1990).
65. G. Baxter, S. Frisken, D. Abakoumov, H. Zhou, I. Clarke, A. Bartos, and S. Poole, “Highly programmable wavelength selective switch based on liquid crystal on silicon switching elements,” in *Optical Fiber Communication Conference*, OSA Technical Digest Series (Optical Society of America, 2006), paper OTuF2.
66. P. Samadi, J. Xu, and K. Bergman. “Virtual machine migration over optical circuit switching network in a converged inter/intra data center architecture,” in *Optical Fiber Communication Conference*, OSA Technical Digest Series (Optical Society of America, 2015), paper Th4G.6.

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015

1. Introduction

Traffic in cloud computing data centers has shifted in recent years from predominantly (80%) outbound (north-to-south) to mostly (70%) rack-to-rack (east-to-west) pattern [1, 2]. This increase in rack-to-rack traffic that is also the case for High Performance Computing (HPC) has introduced complex patterns involving several nodes with large flow sizes such as multicast i.e., transmitting identical data from one-to-many nodes. Many data center applications that use distributed file systems for storage and MapReduce [3] type of algorithms to process data, inherently require multicast traffic delivery. Distributed file systems use state-machine replication as a fundamental approach to build fault tolerant systems. Many of these systems use Paxos [4] algorithm or its variations to provide strong data consistency guarantees. Paxos-type algorithms entail group communication primitives that are mainly multicast.

For example, Google File System (GFS) [5] uses Chubby [6] that is a Paxos-based system. Ceph [7] is also a distributed file system that relies on Paxos. Similar traffic patterns exist in Hadoop Distributed File System (HDFS) [8] and in the shuffle stage of the MapReduce algorithm. Parallel database join operation [9] includes multicast of several hundred megabytes, and the broadcast phase of Orchestra [10] controlled by Spark [11] involves 300 MB of multicast on 100 iterations. Multicast traffic is also frequent in other data center applications such as Virtual Machine (VM) provisioning [12] and in-cluster software updating [13] where 300–800 MB of data are transmitted among hundreds of nodes. Additionally, multicast traffic delivery will facilitate one-to-many VM migrations.

Current data center networks do not natively support multicast traffic delivery. Internet Protocol multicast (IP multicast) [14] is the most established protocol for one-to-many transmission in electronic networks. Supporting IP multicast requires complex configurations on all the switches and routers of the data center network [15]. Scaling IP multicast is also a challenge in multi-tier networks with several IP subnets. Due to the scaling and stability issues with IP multicast [16] and the importance of multicast in data centers, there is a growing interest in improving the performance of IP multicast. To improve scalability, [17] proposes a Software Defined Networking (SDN) [18] solution to manage multicast groups. LIPSIN [19] and ESM [20] rely on encoding forwarding states in in-packet Bloom Filters. Datacast [21] introduces an algorithm to calculate multiple edge-disjoint Steiner trees, and then distributes data among them. Despite these efforts, IP multicast is not supported in the majority of current data center networks and multicast traffic is transmitted either through sequence of unicast transmissions or through application layer solutions such as peer-to-peer methods [22]. These methods are inherently inefficient since they send multiple copies of the same data. Furthermore, such multicast traffic typically suffers from excessive latency that increases with the number of receivers as well as large connection overheads.

In conventional data centers, the interconnection network is an Electronic Packet Switching (EPS) network [23, 24]. Due to the switching cost and cabling complexity, providing a non-blocking EPS network in data centers is a challenge and networks are often forced to rely on over-subscription. Optical Circuit Switching (OCS) is data rate transparent and consumes less switching energy [25]. A hybrid architecture as shown in Fig. 1, providing OCS and EPS along with an SDN control plane to manage the traffic between them, can deliver a viable co-optimized solution [26, 27]. Performing optical switching in data centers would make an immediate improvement in energy efficiency since it eliminates the optical-to-electrical conversions at the switches. Moreover, transmitting larger flows by optical links decreases the traffic load in the aggregation and core tiers and reduces the total switching cost by allowing higher over-subscription on the EPS network.

In [28, 29], the authors proposed the concept of using optics' advanced functionalities for faster delivery of complex traffic patterns such as multicast, incast and all-to-all-cast over an

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015

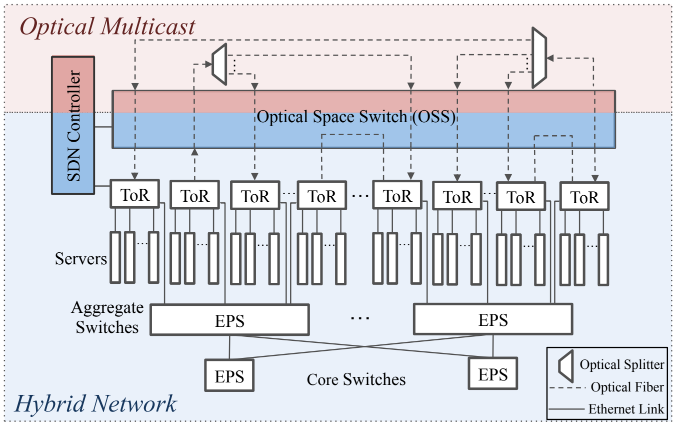


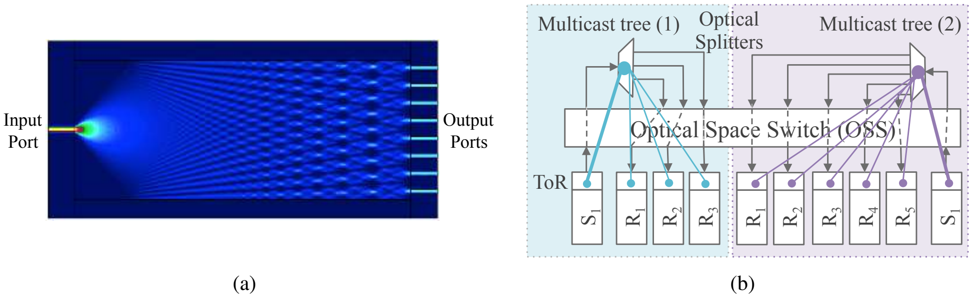
Fig. 1. Optical multicast system network architecture built over a hybrid network, enabling optical multicast by passive optical splitters and an SDN control plane, (ToR: Top-of-Rack).

OCS substrate in data center networks. For example, leveraging passive optical splitters for multicast and time and wavelength multiplexing for incast. Optical layer multicast [30] in the context of transport networks, has been used by researchers to increase the logical connectivity of the network and decrease hop distances at the routing nodes [31]. It is achieved either by passive splitting or frequency conversion in a periodically poled lithium niobate (PPLN) [32].

In contrast to transport networks, the main challenge in implementing an end-to-end system containing optical modules in data center networks, is the control and management integration with conventional data center EPS networks. Moreover, larger multicast groups and faster reconfiguration is necessary. SDN along with cross-layer designs [33, 34] can overcome this critical challenge and provide the optical modules functionalities seamlessly to the higher layers.

In this paper, we present the design and experimental evaluation of an Optical Multicast System for Data Center Networks; an integrated hardware-software system architecture that enables native physical layer optical multicast in data center networks. We designed an application-driven control plane architecture to i) receive multicast connection requests from the application layer, ii) control the routing of the electronic packet switches, optical circuit switches, and connectivity of optical splitters in the data plane, and iii) optimally assign optical splitters to the flows with a resource allocation algorithm. The hardware architecture (Fig. 1) is built on a hybrid network, i.e. the Top-of-Rack switches are simultaneously aggregated by a L2/L3 EPS network and an OCS network provided by an Optical Space Switch (OSS) (OSS is a switching substrate that provides an optical circuit between any idle input and output ports, without optical to electronic conversion [35, 36]). The OSS is also the substrate to connect passive optical splitters to the optical network. The control plane software runs on the SDN controller and communicates with the hosts through the EPS network. We built a prototype to experimentally evaluate the performance of the system and also developed a simulation platform to numerically assess its performance at scale.

Experimental and numerical results show that optical multicast transmits multicast flows simultaneously to all the receivers. It provides similar throughput for delivering multicast flows as IP multicast but i) does not require applying complex configurations on all the switches/routers of the data center to enable IP multicast since multicast trees are directly created by the SDN controller, ii) has superior energy efficiency since it is built on an OCS network that consumes less energy than an EPS network, iii) is future-proof due to the data rate transparency of the sys-



(a) (b)

Fig. 2. (a) Intensity profile of an integrated optical splitter [37], that supports 1:8 optical multicast by dividing the input power, $(P_{out}(i) = \frac{P_{in}}{8}, \forall i = 1, \ldots, 8)$, (b) An example of multicast trees constructed by using passive optical splitters and configuring the OSS ports' connectivity of the senders and receivers ToRs.

tem. In addition, optical multicast can be a compliment service to IP multicast for bulk traffic delivery in real-life scenarios that the Ethernet network is highly over-subscribed. Compared to unicast transmissions where the throughput is inversely proportional to the number of receivers, optical multicast have steady performance irrespective to the multicast group size. Compared to peer-to-peer multicast, it provides at minimum an order of magnitude higher throughput for flows with sizes under 250 MB. Also, it results in shorter and fixed connection overhead (OSS reconfiguration time) that is independent of the number of receivers. Furthermore, the optical multicast architecture is designed to enable direct integration with additional optical modules for optical incast and all-to-all cast functions in data center networks.

Adding the optical multicast system to a data center with a sole non-blocking EPS network decreases the total energy consumption by 50% while delivering 20 TB of data containing 15% multicast flows. The latency also drops by 55%. The improvements are more significant in the case of over-subscribed EPS networks and larger volumes of multicast flows. We also evaluated the resource allocation algorithm with an optimal and greedy heuristic solutions. Our numerical results show that the greedy algorithm is a practical and efficient approach for the control plane. Furthermore, the architecture is designed to enable direct integration with additional optical modules for optical incast and all-to-all cast functions in data center networks.

The rest of this paper is organized as follows. In Section 2, we explain the details of optical multicast, the software and hardware architecture, and the prototype testbed. Section 3 shows the evaluation of different components of the control plane including the resource allocation algorithm. Section 4 is devoted to experimental and numerical evaluations for multicast traffic delivery as well as the cost and energy efficiency analysis of the architecture. Section 5 presents an end-to-end implementation of Ring Paxos on the optical multicast prototype. Section 6 explains the potential design to address incast using optical multicast architecture and we conclude in Section 7.

2. Architecture and implementation

The optical multicast system architecture consists of a 3-layered software component that runs on an SDN controller, and a hardware component built upon an optical circuit switching network. Passive optical splitters are connected to the ports of the OSS and a resource allocation algorithm assigns the splitters to the flows. In this section, we demonstrate the physical layer optical multicast enabled by passive optical splitters. We present the hardware and software architectures, demonstrate the prototype implementation, and discuss its scalability.

Table 1. Insertion loss and cost of the commodity passive optical splitters [39].

Splitter Size	Insertion Loss (dB)	Cost ($)
1:4	7.3	14.00
1:8	10.5	18.70
1:16	13.8	36.10
1:32	16.8	62.00
1:64	20.5	132.00

2.1. Hardware architecture

Physical Layer Optical Multicast: Physical layer optical multicast is performed by optical splitters. As illustrated in Fig. 2(a), these are passive modules that divide the input optical signal to several optical signals by splitting the signal power at predetermined ratios. Optical splitters are manufactured by the Planar Lightwave Circuit (PLC) technology [38] and are commercially available up to 1:128 ratio with the footprint of 2 cm^2 [39]. These splitters are widely used in Passive Optical Networks (PON) [40] for Fiber-To-The-x (FTTx) [41] applications. Table 1 shows the insertion loss and the cost of commodity optical splitters.

As shown in Fig. 1, the hardware architecture is built on a hybrid network. ToR switches are simultaneously aggregated by an optical circuit switching network provided by an OSS and an electronic packet switching network provided by a L2/L3 EPS. MEMS-based OSSs provide high port count optical switching substrates without optical to electronic conversions. Integrated OSS also exist with lower port counts but faster switching speed [42].

ToRs are connected to the OSS by point-to-point optical links and copper cables provide connectivity to the electronic packet switch. Integrated optical splitters have fiber connections and are connected to the ports of the OSS. The controller configures the routing of the OSS and ToRs via OpenFlow [43]. Optical splitters are passive and do not require any configuration.

Figure 2(b) demonstrates physical layer optical multicast between the racks in a data center network using passive optical splitters. Multicast trees are created between the senders and the receivers by configuring the OSS ports. The sender S_1, is connected to the input port of the splitter and receivers $R_1,\ldots,R_n$ are connected to the output ports. The upper bound for the multicast group size is set by the optical link power budget.

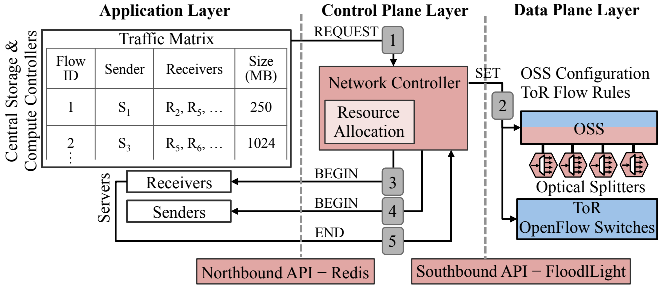
2.2. Software architecture

Application-driven Networks: Application-driven networking [44–46] and SDN are increasingly used for designing cloud-based data center networks. Big-data applications are also moving in this direction. For example, in Hadoop [47], NameNode and JobTracker are the compute and storage controllers that manage the HDFS and MapReduce tasks, respectively, over the nodes. Global knowledge of application processes and the storage systems, as well as the central management of the network, can be intelligently used to improve the performance of big data applications. While designing the software architecture, we take the application-driven approach, as it seems to be the emerging direction.

Figure 3 demonstrates the software architecture consisting of the application, control plane, and the data plane layers. The network controller (including the resource allocation component) is in the control plane layer. It receives multicast traffic requests from the application layer via the northbound API. It configures ToRs, OSS and optical splitters connectivity in the data plane accordingly through the southbound API.

The central storage and compute controllers provide the traffic matrix of multicast flows to the network controller. For each flow request, the traffic matrix provides the flow ID, the sender and receivers servers IDs and the flow size. The resource allocation algorithm processes the traffic matrix and assigns flows to the optical splitters with the goal of maximizing the traffic

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015



Flow ID	Sender	Receivers	Size (MB)
1	S_1	$R_2, R_5, \ldots$	250
2 ⋮	S_3	$R_5, R_6, \ldots$	1024

Fig. 3. The 3-layered software architecture performs: i) Configuration of the OSS and ToRs, and connectivity of the optical splitters in the data plane layer, ii) Receipt of multicast traffic matrix at the application layer from the central compute and storage controllers, iii) Assignment of optical splitters to the flows by a resource allocation algorithm.

offloaded to the optical multicast system (see Section 2.3). The output of the resource allocation algorithm is a set of configurations corresponding to the selected flows and the splitters. The network controller sets the configurations to the ToRs and OSS through the Southbound API. Once the configurations are finished, the network controller notifies the servers involved in the scheduled flows to begin the transmission through the northbound API. Upon completing the transmission, the receivers notify the network controller and it releases the splitters and servers involved in the scheduled flows. The traffic matrix is updated with flows from the applications and the resource allocation algorithm selects the next set of flows. Servers can also request for multicast connection via Northbound API. The network controller aggregates multicast flows between the servers that share the same rack and inserts them in the traffic matrix.

2.3. *Resource allocation algorithm*

The objective of the resource allocation algorithm is to maximize the multicast traffic offloaded to the optical multicast system under the constraint of limited number of splitters. Depending upon the reallocation strategy, the resource allocation algorithm allocates flows to the splitters when a certain number of splitters are available. The reallocation strategy can be myopic (immediate reallocation of free splitters) or far-sighted (wait for a large number of free splitters before reallocation).

We model the resource allocation problem as an Integer Program. We denote by F the number of multicast flows and by R the number of racks. Multiple multicast flows can have the same sender and receiver racks. To simplify presentation, we refer to a flow as an aggregate of all the flows with the same set of senders and receivers. Each flow i has a size f_i and the number of optical splitters required s_i (if $s_i > 1$, splitters are cascaded). The binary variable a_i indicates if flow i is scheduled in the current computation. r_j is 1 if the rack j is available at the current iteration. The constant m_{ij} is 1 if the i^{th} flow requires rack j as a sender or a receiver. S denotes the number of splitters available and we assume that all modules have an identical number of ports (our model can be extended to support different number of ports). The problem can then be formulated as follows:

$$\max \sum_i a_i \cdot f_i \tag{1}$$

$$\sum_{i=1}^{F} a_i \cdot m_{ij} \leq r_j \quad \forall j = 1 \cdots R \tag{2}$$

$$\sum_{i=1}^{F} a_i \cdot s_i \leq S \tag{3}$$

At every iteration, the resource allocation algorithm selects the set flows from the traffic matrix that maximizes the objective (1). Each node has a single optical port and can serve only one flow at any instant, which is modeled by the constraint (2). Finally, the limit on the number of optical splitters is modeled by (3).

Optimal Solution: The Integer Program above, is a variant of the NP complete multidimensional knapsack problem [48]. The Integer Program can be optimally solved by branch-and-bound methods. The optimal branch-and-bound methods can be potentially time consuming and lead to wasted optical resources (In Section 3.2, we show that optimal calculation for a large number of racks and flows can exceed the typical OSS reconfiguration time of 20-30 ms). Thus, we also employ a heuristic to efficiently solve the resource allocation problem.

Greedy Solution: The greedy algorithm iteratively selects the flows with the maximum values of traffic, $\sum_{j=1}^{H} f_i \cdot m_{ij}$ and checks if the flow can be scheduled (optical ports of all associated racks are free and required number of optical splitters are available). If yes, then the flow is scheduled and the racks and optical modules are marked as occupied. The greedy approach is faster but sub-optimal.

2.4. *Prototype and testbed*

Figure 4(a) shows the optical multicast system prototype configuration used in the experimental evaluations (see Section 4). It consists of 8 racks, each consisting of one server. Each server is equipped with a dual-port 10G Network Interface Controller (NIC), an Intel Xeon E5-2430 6-core processor and 24 GB of RAM. A Pica8 P-3920 10G OpenFlow switch is divided into 8 bridges that operate as 8 separate ToR switches. The EPS network among the ToRs is provided by a Juniper EX4500 switch. The Juniper EX4500 is a 40-port non-blocking 10G Ethernet switch that consumes 350 W and has a 2.7 μs latency. 10G Small Form-factor Pluggable (SFP+) Direct-Attached (DA) cables are used to connect ToRs to the Juniper switch.

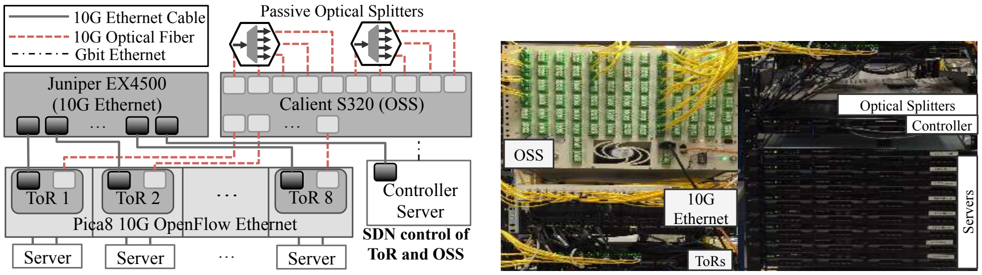
The optical network is an OCS constructed by a Calient S320 OSS. The Calient S320 is a 320-port MEMS OSS with the connection setup time of 25 ms, 20 ns port-to-port latency, 45 W operation power, and a typical 2.0 dB insertion loss. 18 ports of the Calient switch are used to connect 8 ToRs and two 1:4 passive optical splitters. 1:4 optical splitters have 7.3 dB insertion loss and are connected to the Calient S320 by single-mode fibers. Optical to electronic conversions at ToRs are carried out by 10GBASE-ZR SFP+ transceivers and single-mode fibers provide optical links to the Calient S320. The controller server is also equipped with a dual-port 10G NIC, two Intel Xeon E5-2403 4-core processors and 24 GB of RAM.

We used Floodlight as the southbound API since the majority of commercial ToRs and OSSs are now OpenFlow enabled. For the OSSs that do not support OpenFlow, we developed a python-based API that controls the switch using TL1 commands. For the northbound API, we implemented a fast pub/sub messaging system using open-source libraries of Redis [49]. Byte size messages are transmitted through the EPS network from the network controller to the servers. The messaging system is much faster than the REST API, conventionally used as the northbound API.

2.5. *Scalability*

The scalability of the hardware architecture is determined by i) multicast group size, and ii) OSS port count. Every 1:2 optical multicast reduces the signal power by 3 dB. A 1:64 optical multicast requires 18 dB (20.5 dB in a manufactured device) link power budget that can be provided by SFP+ ZR transceivers. Cascading sixteen 1:32 passive optical splitters to the output ports of one 1:16 active optical splitter scales optical multicast group size to 512 racks using 545 optical ports. Active optical splitters provide lossless splitting using semiconductor optical

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015



(a) (b)

Fig. 4. Optical multicast system prototype: (a) Configuration, and (b) Picture. The prototype consists of an Ethernet switch, an OSS, 8 ToR emulated by an OpenFlow switch, 8 servers, two 1:4 optical splitters, and an SDN server that runs the control plane software.

amplifier (SOA) [50]. They can also provide tunability in the splitting ratio using Mach-Zehnder Interferometers (MZI) [51] to create asymmetric multicast trees.

MEMS OSS with 320 ports are commercially available [35] and 1100 port implementation was presented in [52]. Considering same number of splitter ports as number of racks, a 1100 port OSS can support 512 racks and maximum multicast group size of 512 (broadcast). Multiple OSSs (with an SDN controller to manage the traffic among them) can be placed in ring or tree topologies to support larger number of racks. Commodity OSSs are open-flow enabled and compatible to integrate with SDN data centers.

Software architecture scalability is imposed by the control plane delay. The control plane achieves an average end-to-end delay of 30–50 ms for processing 320 multicast jobs in a 320 rack data center (Section 3.1). The control plane delay grows slowly with increasing traffic matrix and rack sizes. This makes the system scalable to support hundreds of racks and numerous multicast flows.

3. Control plane evaluation

In this section, we evaluate the implementation of the prototype control plane. We measured the execution delay of different control plane components. The results indicate that the control plane incurs a very low overhead apart from the fixed OSS reconfiguration time. Moreover, we numerically study the optimal and greedy resource allocation algorithms, the reallocation strategies, and the impact of number of splitters. Our results show that: (i) the greedy heuristic is a practical and efficient solution to the resource allocation problem, (ii) a myopic reallocation strategy, can be more efficient than a far-sighted strategy, and (iii) deploying a large number of small splitters improves the performance of flows with small multicast group sizes.

3.1. Control plane

The total delay of the control plane consists of communicating via the northbound (Redis Messaging System) and southbound APIs, the resource allocation algorithm, and the network controller software. For Redis, we measured the average latency for transmitting 100 messages of 20 bytes between the controller and the servers. In our measurements, we did not include the ToR setup time (flow rule entry), since it is much faster (5–10 ms) than the OSS reconfiguration and is executed in parallel. Table 2 shows the average delay of different components for a configuration of 320 ToRs, processing traffic matrix of 320 multicast flows with ten 1:32 optical splitters on the prototype controller server. The total delay is 52.8 ms and 32.3 ms for the optimal and the greedy algorithms respectively.

Table 2. Average control plane delays measured on the prototype.

Control Plane Component	**Delay (ms)**
Northbound API (Redis)	0.65
Computing an optimal solution	22.40
Computing a greedy solution	1.90
Network Controller software	4.8
OSS reconfiguration	25
Total (Optimal):	**52.8**
Total (Greedy):	**32.3**

3.2. Algorithm evaluation

We study the performance of the resource allocation algorithm through simulations. In our evaluations, the resource allocation algorithm runs on a traffic matrix of a given size, which is periodically repopulated with random flows. The flow sizes are uniformly distributed between 250 MB–2.5 GB and the multicast group is selected uniformly randomly among all the racks subject to a maximum group size. We modeled the OCS network with link speed and reconfiguration delays as measured on the prototype. The simulation time includes the OSS reconfiguration delay, the resource allocation algorithm computation delay, and the transmission time on the optical links. The results are averaged over several runs of 200 seconds simulation time. We define following metrics for our evaluations:
i) **Achievable Throughput**: Average throughput over the optical network excluding the throughput loss due to the idle time during resource allocation algorithm computation: $\frac{\text{Traffic Transmitted}}{\text{(Total Time - Total Algorithm Time)}}$. ii) **Effective Throughput**: Overall average throughput over the optical network: $\frac{\text{Traffic Transmitted}}{\text{Total Time}}$.

The achievable throughput is the theoretical maximum throughput that can be achieved by providing more computation power to the controller and using advanced optical switching technologies.

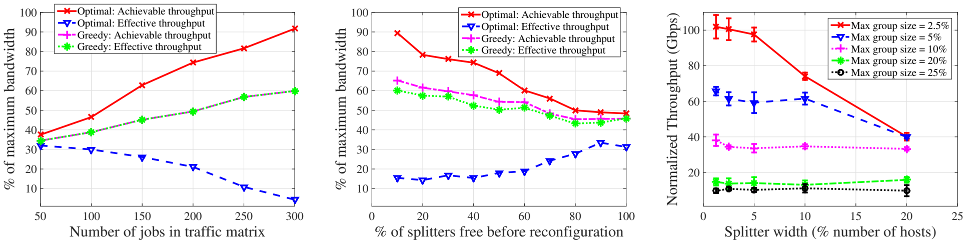
3.2.1. Optimal vs. greedy allocation

We computed the achievable and effective throughput for different traffic matrix sizes in a 320 rack network with twenty 1:16 splitters and maximum multicast group size of 32. Figure 5(a) shows the achievable and effective throughput as a percentage of the maximum bandwidth of the optical network. The achievable throughput for the both optimal and greedy solutions increases as the traffic matrix size increases. A larger traffic matrix increases the probability of scheduling larger flows, thus, increasing the achievable throughput. The optimal algorithm incurs large computation delay in processing large traffic matrix sizes and consequently, the effective throughput reduces with increasing matrix sizes. The difference between the achievable and effective throughput for the greedy algorithm is small due to fast algorithm computation. In summary, the greedy algorithm is an efficient heuristic in practice as it trade-offs the sub-optimal solution with faster performance. The optimal solution's computation time can be reduced by a more efficient implementation of the branch-and-bound method.

3.2.2. Reallocation strategy

We evaluate the reallocation strategies by measuring the achievable and effective throughput vs. the number of free splitters prior to reallocation. Figure 5(b) shows the results for an architecture of 320 racks and twenty 1:16 splitters. Myopic reallocation (waiting for a small number of splitters before reconfiguration) of free optical splitters, leads to a higher achievable throughput for both optimal and greedy solutions. Reconfiguring the switch as soon as a few splitters

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015



(a)

(b)

(c)

Fig. 5. (a) Achievable and effective throughput of the optimal and greedy algorithms vs. the traffic matrix size, (b) Impact of the reallocation strategy on the maximum achievable and effective throughput, and (c) Effect of optical splitter size on the maximum achievable throughput for 160-640 racks.

are available prevents wastage of optical resources (e.g. when a large flow and a small flow are scheduled together, a small flow will finish much faster). The far-sighted strategy (waiting for a large number of splitters before reconfiguration) leads to less frequent reconfigurations and consequently less overhead due to OSS reconfiguration and control algorithm delay. Since the optimal algorithm has a large control overhead, the effective throughput of the optimal algorithm is higher for the far-sighted strategy. However, far-sighted strategy results in a lower achievable throughput for the greedy algorithm. For the greedy algorithm, the control overhead is relatively low and the losses due to the idle time dominate.

3.2.3. Optical splitter sizing

Due to the limited number of ports on the OSS, only a fixed number of splitters can be used. Thus, it is important to determine the optimal number and size of the splitters. We evaluate the achievable throughput for the optimal algorithm for a traffic matrix of 100 multicast flows with maximum group size varying between 2.5%–25% of the number of racks. We consider the total number of racks ranging from 160 to 640 with an equal number of OSS ports in each scenario.

Figure 5(c) shows the achievable throughput vs. the optical splitter widths (as a percentage of the number of racks) for different maximum multicast group sizes. The average throughput values are normalized to the same scale after accounting for increased capacity due to number of ports in each scenario. We observe that for smaller multicast group sizes, the achievable throughput is higher with narrower but larger number of splitters. The achievable throughput decreases as the multicast group size increases as several modules need to be cascaded to serve one single flow. As a rule of thumb, we will use splitter sizes between 5–10% of the total number of racks. This range of splitter size limits excessive cascading for larger multicast group sizes and provides higher throughput.

4. System evaluation

In this section, we evaluate the performance of Optical Multicast System for transmitting bulk multicast flows and compare it with IP multicast, unicast, and peer-to-peer methods. In all evaluations, optical multicast refers to physical layer optical multicast using passive optical splitters. We start by presenting experimental results measured on the prototype testbed. Next, we present numerical evaluations at scale, computed on our custom simulation platform and conclude with the cost and energy efficiency analysis. We use following metrics:
i) **Transmission Time:** Time to deliver a flow excluding the connection overheads.
ii) **Latency:** Total time to deliver a set of flows including all connection and control plane

overheads.
iii) **Throughput:** Link throughput while transmitting one flow: $\frac{\text{Flow Size}}{\text{Transmission Time}}$.
iv) **Connection Overheads:** The total delay from a request to begin the flow transmission.
v) **Energy Consumption:** Energy (Joules) = Power (Watt)×Transmission Time (Seconds).

We compare the performance for delivering a traffic matrix of multicast flows with i) IP multicast that creates a multicast tree (star for the 1-hop topology in our prototype) using spanning-tree algorithm, manages the multicast group memberships, and replicates packets at the switch/router, ii) sequence of unicast transmissions on the EPS network, iii) peer-to-peer method that imitates multicast transmission by creating many-to-many connections using bittorrent [22], and iv) an optical circuit switching network not equipped with the optical multicast system.

We perform comparison with both non-blocking and over-subscribed EPS networks as real-world data center networks are typically over-subscribed. Moreover, since the extra optical switching capacity in the optical multicast system allows further over-subscription of the EPS network, this comparison helps us to evaluate the benefits of adding this system to a data center network. Following are the network configurations in our evaluations:

- Physical layer optical multicast
- Transport layer IP multicast
  - Non-blocking EPS network
  - EPS network + background traffic
- Multicast through sequence of unicasts
  - OCS point-to-point network
  - Non-blocking EPS network
  - EPS network + background traffic
- Peer-to-peer multicast using Twitter Murder
  - Non-blocking EPS network
  - EPS network + background traffic

Implementation Details: We used Iperf [53] to generate data and measure the link characteristics. Iperf is a common network performance measurement tool that generates UDP and TCP datagrams and measures the network throughput. UDP is an unreliable but fast and efficient transmission protocol For a fair comparison against peer-to-peer multicast that guarantees data delivery, we optimized the UDP buffer size and service type to achieve an average 0.35% packet error rate for 4.2 Gbps throughput. The multicast transmission can be improved by using reliable multicast protocols [54,55] in which, data is transmitted on the optical network and the NACKs on the electronic network [56]. Flows are read/written in the host memory on transmission/reception to make maximal use of the optical link bandwidth. The Juniper EX4500 switch in the testbed provides advanced Layer 3 functionality that allows IP multicast implementation. For peer-to-peer multicast, we implemented Murder [13] that is a bittorrent-based fast data distribution platform developed by Twitter. We implemented Murder using the open source Herd libraries [57]. To emulate over-subscription, we generated background traffic using Distributed Internet Traffic Generator (D-ITG) [58] on a spare NIC of the servers.

4.1. Experimental results

In the first experiment, we transmit a traffic matrix of 50 multicast flows, with uniform distribution of flow size (250 MB–2.5 GB) and multicast group size (1–4). These flow sizes are chosen based on the data center applications. As discussed in the introduction, parallel database join operations include multicast flows of several hundred megabytes. For software updates (e.g. OS updates) and VM migrations, the flow sizes are in the range of gigabytes. Figure 6(a) shows the

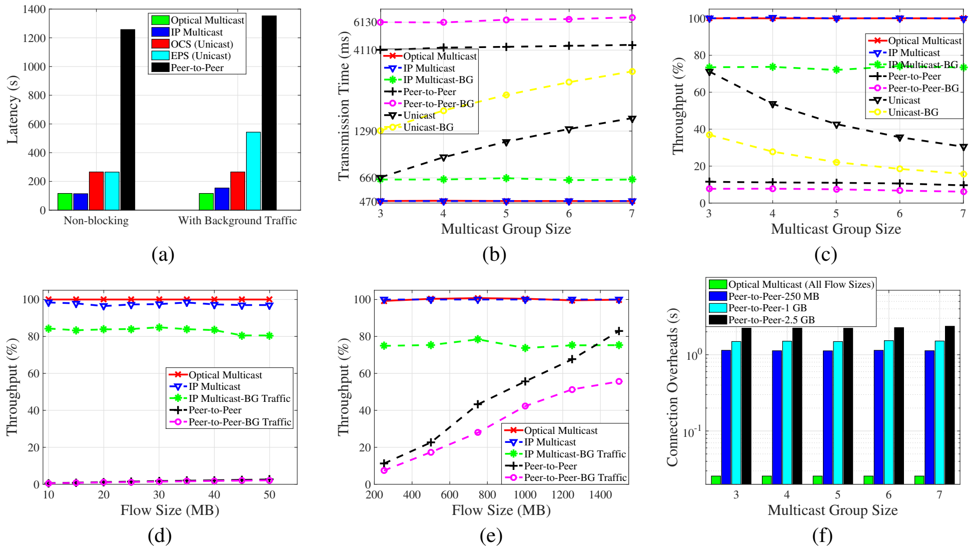


Fig. 6. Experimental results: (a) Latency to deliver 50 multicast flows in a configuration of 8 racks and two 1:4 optical splitters. (b, c) Evaluating the effect of increasing number of multicast receivers on the transmission time and the throughput of a 250 MB flow, (d, e) Evaluating the effect of flow size on throughput for mice and elephant flows, (f) Evaluating the effect of flow and multicast group size on peer-to-peer multicast connection overheads, (BG: Background Traffic).

latencies over different network configurations. Optical multicast and IP multicast have comparable latencies. IP multicast with background traffic leads to 32% higher latency than optical multicast. Transmitting multicast flows with sequence of unicast flows is twice as slower than optical multicast. In this case, adding background traffic increases the latency by over 3x. Peer-to-peer multicast on the EPS network is an order of magnitude slower than optical multicast with and without background traffic. Transmitting the same traffic matrix in a hybrid network not equipped with the optical multicast system takes twice as long. We observe that the OSS reconfiguration time does not noticeably affect optical multicast latency for bulk multicasting. Equipping an OCS network with optical multicast has significant impact on the multicast traffic delivery. Furthermore, peer-to-peer is a time-consuming multicast data delivery method.

In the next experiment, we evaluate the effect of multicast group size on transmitting one multicast flow. We measured the transmission time excluding the connection overheads. Figure 6(b) shows the results for a 250 MB flow. Optical multicast, IP multicast and peer-to-peer have constant transmission time regardless of multicast group size. Unicast method transmission time increases linearly with the number of receivers.

We also measured the throughput vs. multicast group size as shown in Fig. 6(c). Optical multicast and IP multicast achieve the highest throughput regardless of multicast group size and we compare all subsequent results with this value. Introducing background traffic decreases the throughput of IP multicast to 73% of its original throughput. Unicast method's performance decreases with increasing group size. For multicast group of 7 receivers, unicast method's throughput is almost 30% of optical multicast. Background traffic reduces the throughput of unicast further close to 16% of the optical multicast. Peer-to-peer multicast has a very low throughput for a 250 MB flow size close to 10% of optical multicast. This result confirms that multicast group size does not change the performance of optical and IP multicast. However, unicast transmission performance is highly dependent on the number of receivers, as expected.

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015

Peer-to-peer method also has constant throughput over different multicast group sizes but it is an order of magnitude lower than optical multicast.

We also measured the impact of the flow size on the throughput for optical multicast, IP multicast, and peer-to-peer methods. For each measurement point, we performed 3 multicast transmissions with 2–4 multicast receivers and average the throughput. However, based on the previous experiment, the multicast group size has no impact on the performance of these methods. We plotted all the measurements compared to the highest throughput that was for optical and IP multicast.

Figures 6(d) and 6(e) show the results for mice and elephant flows, respectively. Optical and IP multicast achieve similar performance irrespective of the flow size. Introducing background traffic has lower impact on the throughput of mice than elephant flows. Peer-to-peer multicast has an average throughput of 1–2% for mice flows. Its performance improves as the flow size increases and it reaches similar performance as optical multicast for flows larger than 1.5 GB. We conclude that peer-to-peer multicast is not efficient for transmission of mice multicast flows. However, for very large flows, it achieves comparable performance as optical and IP multicast. For peer-to-peer multicast, the impact of background traffic is more notable on transmission of elephant flows.

We further compared the efficiency of optical multicast and peer-to-peer multicast by measuring the connection overheads on transmitting 250 MB, 1 GB and 2.5 GB flows. Optical multicast connection overhead is the OSS configuration time. For peer-to-peer multicast, it is the bittorrent file (metadata of the flow) generation and peer-to-seed connection times. Figure 6(f) shows the results for 3–7 multicast receivers. Peer-to-peer multicast has significantly higher and variable connection overhead that increases with the flow size.

We also measured the share of connection overheads in the latencies of the first experiment presented in Fig. 6(a). For optical multicast, it was 5% of the overall latency but 47% for the peer-to-peer multicast. We infer that peer-to-peer multicast has longer and varying connection overheads that increases mainly by the flows size and slightly with the multicast group size. However, the connection overhead for optical multicast is the fixed OSS reconfiguration time.

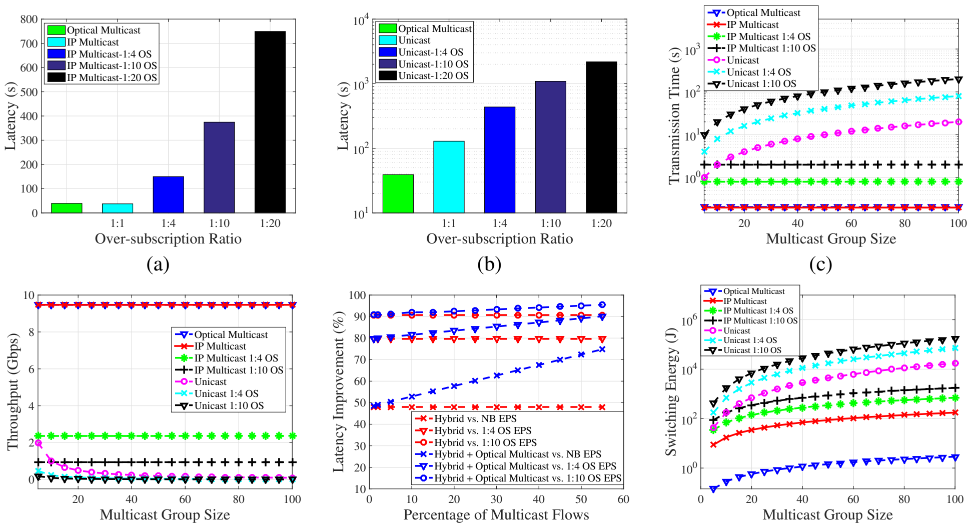
To summarize, optical multicast achieves similar performance as IP multicast regardless of the flow size. The performance of optical multicast is not degraded by increasing the multicast group size. Multicast transmission through sequence of unicast flows has link stress proportional to the number of receivers and results in higher latencies. Peer-to-peer multicast is not suitable for transmission of mice flows. However, it can be a reasonable solution for low-priority bulk multicasting. Furthermore, adding the optical multicast system to a hybrid network will significantly improve the performance of multicast flow transmission.

4.2. Numerical results

In order to evaluate the performance of the optical multicast system at scale, we developed a custom packet-based simulation environment using NS3 libraries [59]. For optical multicast, we used testbed measurements to adjust the channel end-to-end delay. We also added the OSS reconfiguration time to the connection overheads and used on-off communication patterns to generate data. To get statistically meaningful results, we repeated each experiment 10 times and averaged the results. For our evaluations, we used following network configurations:

- Physical layer optical multicast
- Transport layer IP multicast
  - Non-blocking EPS network
  - 1:4 Over-subscribed EPS network
  - 1:10 Over-subscribed EPS network
- Multicasting over sequence of unicasts

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015



(a) (b) (c)

(d) (e) (f)

Fig. 7. (a, b) Numerical results on the latency of delivering 320 multicast flows among 320 racks with ten 1:32 optical splitters, comparing optical with IP multicast and unicast on an EPS network in non-blocking and over-subscription configurations, (c, d) Effect of multicast group size on transmission time and throughput for a 250 MB flow, (e) Latency improvement of a hybrid and optical multicast-equipped data center network in delivering 20 TB of data compared to a sole EPS network vs. percentage of multicast flows, (f) Calculation of switching energy on delivering a 250 MB multicast flows to achieve similar latency vs. Multicast Group Size, (OS: Over-subscribed, NB: Non-blocking).

– Optical point-to-point OCS network
– Non-blocking EPS network
– 1:4 Over-subscribed EPS network
– 1:10 Over-subscribed EPS network

In the first evaluation, we consider a network of 320 racks and ten 1:32 optical splitters. A traffic matrix of 320 multicast flows with maximum multicast group size of 32 and flow size of 250 MB–2.5 GB (both uniformly distributed) is transmitted on the networks. We used the greedy resource allocation algorithm.

Figures 7(a) and 7(b) show the latencies for different network configurations. Optical multicast and non-blocking IP multicast achieve comparable latencies. For IP multicast, a 1:4 and 1:10 over-subscribed EPS network leads to 3x and 9x higher latencies, respectively. Multicast through sequence of unicast transmissions over a non-blocking EPS network results in 50% higher latency as compared to optical multicast. In this case, 1:4 and 1:10 over-subscription decrease the latencies by 5x and 13x, respectively. Confirming the experimental results, OSS reconfiguration time does not result in notable additional latency for bulk multicasting. Multicast through unicast transmissions is not efficient for transmitting multicast flows, especially in over-subscribed networks.

In the next set of numerical evaluations, we study the effect of multicast group size. Figure 7(c) shows the transmission time of a 250 MB flow to 5–100 multicast receivers. We observe that increasing the multicast group size does not affect the performance of optical and IP multicast. However, the performance of unicast transmission degrades as the multicast group size grows. Figure 7(d) shows the throughput with increasing multicast group size. Optical and IP multicast achieve close to line-rate (10 Gbps) throughput. Over-subscription by ratios of 4

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015

Table 3. Power consumption and cost of the EPS, OCS and the Optical Multicast System network components.

Component	**Per Port Energy (W)**	**Per Port Cost** ($)
10GBASE-SR SFP+	1	260
10GBASE-ZR SFP+	1.5	605
10G EPS Switching	8.75	575
Optical Switching	0.14	350
Optical Splitter	0	2

and 10, decreases the throughput of IP multicast to 2.36 and 0.94 Gbps, respectively. Finally, the throughput of unicast transmission is inversely proportional to the number of receivers.

In order to better understand the impact of the optical multicast system on a data center network, we computed the latency for delivering 20 TB of data based on the overall switching capacity (including switching delays) for the following network configurations: i) An EPS network in non-blocking, 1:4 and 1:10 over-subscription configurations, ii) A hybrid architecture consisting of an OCS network and an EPS network in non-blocking, 1:4 and 1:10 configurations, and iii) Optical multicast system with 320 splitter ports on a hybrid architecture with the EPS network in non-blocking, 1:4 and 1:10 over-subscription configurations. In the sole EPS configuration, the EPS layer serves both unicast and multicast (through sequence of unicast transmissions) flows. In the case of hybrid configurations, the optical layer first serves all multicast flows and then transmits unicast flows along with the EPS layer. We varied the volume of multicast flows 1–55% with average multicast group size of 16. As shown in Fig. 7(e), adding an extra OCS network results in 48% lower latency than a sole EPS network. With 15% of the total multicast flows, the optical multicast system reduces the latency by 55%. Adding the optical multicast system to an over-subscribed EPS network, the latency is reduced by 83% and 92% for 1:4 and 1:10 over-subscription ratios, respectively. The gains of adding the optical multicast system improve with increasing percentage of multicast flows and larger average multicast group sizes.

4.3. Cost and energy efficiency

Optical circuit switching consumes considerably less power than electronic packet switching. Table 3 shows the typical per port power values for commercially available EPS, OCS and the optical multicast system network components. In the optical multicast system prototype, the per port power consumption of optical switching is 60x lower than EPS. Furthermore, using OCS in data centers avoids unnecessary optical-electrical-optical conversions at the electronic packet switches. We computed the total switching energy to achieve similar latency values for different network configurations. It is calculated based on the per port energy consumption and the transmission time ($E_{(J)} = P_{(W)}\ \ t_{(s)}$), thus to achieve similar latency with electronic unicast as optical multicast, more EPS ports (more bandwidth) are required, i.e. more energy consumption. Figure 7(f) shows the switching energy as the multicast group size increases. To deliver a 250 MB multicast flow, optical multicast consumes an order of magnitude less switching energy than IP multicast. The difference grows to 2 orders of magnitude for IP multicast in a 1:4 over-subscribed network as well as the unicast method in a non-blocking configuration.

For a more comprehensive energy efficiency analysis, we computed the total energy consumption (sum of the transceiver and the switching energy consumptions) for delivering 20 TB of data in a 320 rack data center with an average multicast group size of 16. In Fig. 8(a), we compare a solely EPS network with a hybrid network equipped with optical multicast. With multicast flows constituting 15% of the total volume of the flows, a hybrid network equipped

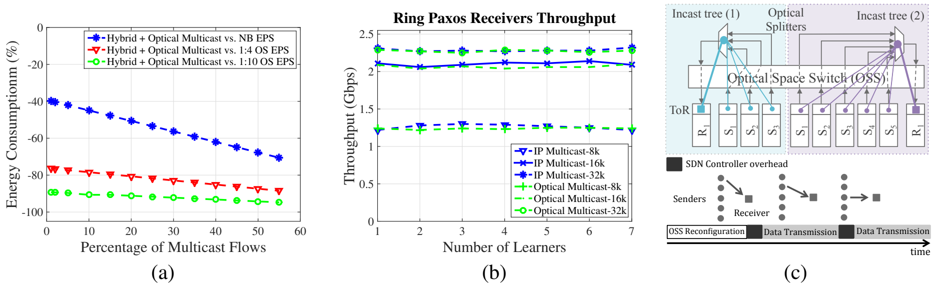


Fig. 8. (a) Improvement in energy consumption on delivering 20 TB of data with Hybrid + optical multicast network compared to a sole EPS network in non-blocking, 1:4 and 1:10 configurations (OS: Over-subscribed, NB: Non-blocking), (b) Ring Paxos run on IP multicast supported EPS network and optical multicast-enabled network (message size: 8, 16 and 32 kbytes), (c) Enabling optical incast using passive optical combiners and Time-Division Multiplexing of the senders by the SDN controller.

Table 4. Cost increase in adding an OCS network + Optical Multicast System to a 320 rack data center EPS network under different over-subscription conditions.

Network Configuration	Interconnection Network Cost Increase
Non-blocking EPS + OCS + Optical Multicast	156%
1:4 OS EPS + OCS + Optical Multicast	104%
1:10 OS EPS + OCS + Optical Multicast	87%

with optical multicast consumes up to 50, 80, and 91% less switching energy than a solely EPS network in non-blocking, 1:4, and 1:10 over-subscribed configurations, respectively.

Enabling optical multicast in a hybrid network requires optical transceivers with higher output power and extra optical switching ports to attach the splitters. The per port costs and energy values for a typical transceiver used in hybrid networks (SFP+ SR) and the ones required for the optical multicast architecture (SFP+ ZR) are presented in Table 3. We calculated the cost of building a hybrid network + optical multicast vs. a non-blocking EPS network. Adding an extra optical network increases the overall switching capacity of the data center. This allows for supporting more servers or increasing the over-subscription ratio of the EPS network. Table 4 shows the additional cost of building the optical multicast system, compared to a solely non-blocking EPS network in 3 configurations: i) non-blocking, ii) 1:4 over-subscribed, and iii) 1:10 over-subscribed. We ignored the links cost in our analysis as it is negligible. We assumed that the cost of an EPS network linearly reduces with increasing the over-subscription ratio. Building a hybrid + optical multicast network with a 1:4 over-subscribed EPS network will cost twice as much compared to a solely EPS non-blocking network. However, even with 10% of the total flows being multicast, it will provide approximately 80% lower latency (Fig. 7(e)) and energy consumption (Fig. 8(a)). Furthermore, considering that network is only 15% of the data center cost [60], the investment improves data center performance and reduces the operating costs. For large-scale data centers with more than 320 racks, the per port switch cost and switching energy consumption scales linearly.

5. Paxos with optical multicast

As discussed in Section 1, distributed files systems are widely used as a data storage solution in data center networks. Majority of these systems use Paxos algorithm or its variation to provide strong consistency guarantees. Paxos-type algorithms will significantly benefit from multicast-enabled networks. Ring Paxos [61] is a variation of Paxos that uses IP multicast to disseminate messages among the learners. We chose Ring Paxos since compared to other atomic broadcast protocols [62, 63], it achieves higher throughput, lower latency, and steady performance as the number of receivers increases. We implemented Ring Paxos on the servers of our prototype testbed and evaluated its performance over the optical multicast and an IP multicast-enabled EPS network. The configuration is 5 Acceptors, 1–7 Learners, and 8, 16 and 32 kbytes message sizes. Figure 8(b) shows the receiving throughput of the Learners that confirms successful end-to-end implementation of Paxos on the optical multicast system.

6. Optical incast

Optical multicast architecture is designed to enable direct integration of optical modules and subsystems such as Arrayed Waveguide Gratings (AWG) [64] and Wavelength Selective Switches (WSS) [65] to provide functionalities such as incast, all-to-all-cast, or aggregation/breakout of links with different data rates. These functionalities can also address inter data center network applications such as multicast and incast between data centers or providing rack-to-rack connectivity across data centers to improve scalability and reliability [66].

Optical splitters are bi-directional and work as combiners as well. This functionality allows enabling rack-to-rack optical incast. As demonstrated in Fig. 8(c), incast flows can be routed by i) building an incast tree between all the senders ($S_1,\ldots,S_n$) and the receiver (R_1) using the optical combiner and, ii) time-division multiplexing the senders using an SDN controller to utilize the full link capacity. Compared to optical multicast, optical incast does not require high power transmitters since the optical signal of the senders are added rather than divided. However, achieving efficient time-division multiplexing of senders requires a fast northbound API to minimize the controller overhead.

7. Conclusion and future work

In this paper, we presented a unique hardware-software system architecture to integrate circuit-based optical modules to data centers Ethernet network. We demonstrated an optical multicast system, to enable efficient physical layer multicast through passive optical splitters. It is built on a hybrid architecture that combines traditional electronic packet switching with optical circuit switching networks. The optical space switch is the switching fabric of the optical network and also the connectivity substrate of splitters. Network management and configurations are handled through a 3-layered SDN control plane. We built a hardware prototype and developed a simulation environment to evaluate the performance of the system.

Optical multicast delivers multicast flows to all receivers simultaneously irrespective of the multicast group size, similar to IP multicast. However, optical multicast performs a more efficient multicast in data centers considering that: i) It is built on an optical circuit switching substrate with lower energy consumption than electronic packet switching, ii) Does not require applying complex configurations on all switches and routers to enable IP multicast since multicast group management and tree formation is handled by the SDN controller. Compared to application layer multicast using peer-to-peer methods, optical multicast achieves considerably higher throughput for large range of flows sizes (up to 1.5 GB) with fixed, minimal connection overheads. Furthermore, optical multicast is future-proof since optical space switches and passive signal duplication by optical splitters are data rate transparent.

Acknowledgments

This research was supported in part by CIAN NSF ERC under grant EEC-0812072 and NSF CNS-1423105. We would also like to thank Juniper Networks and Calient Technologies for their generous donations to our testbed.

#241082 Received 25 May 2015; revised 31 Jul 2015; accepted 4 Aug 2015; published 14 Aug 2015