Toward Optical Switching in the Data Center

(Invited Paper)

William M. Mellette, Alex C. Snoeren, and George Porter
University of California, San Diego

***Abstract*—Optical switching may be instrumental in meeting the cost, power, and bandwidth requirements of future data center networks. However, optical switching faces many challenges to practical adoption. We discuss some of the physical- and control-layer challenges that have been uncovered in the research community, and potential ways to address them.**

I. Introduction

Optical switching has seen widespread adoption in wide-area telecommunications networks, where it is used to offload high-bandwidth traffic from electronic packet routers [1]. The bandwidth growth in data centers is outpacing that of telecom networks, begging the question of when optical switching will be deployed in data centers. The research community has been actively investigating this question for many years, and has identified a number of problems and solutions to deploying optical switching in these new environments. Here, we review a subset of our recent work in this space, highlighting a number of challenges and potential solutions.

The high distance-bandwidth product of links in today's data centers require optical fiber be used throughout most of the network. Switching, however, is still performed electronically. Optical switching is appealing in this context due to its ability to passively forward data traffic carried over fiber, eliminating the need for signals be converted between the optical and electronic domains. Since conversion and electronic switching are expensive at high data rates, optical switching has the potential to reduce overall network cost and power consumption if it can replace or augment a large fraction of the packet switches.

Unfortunately, optical switches are not drop-in replacements for electronic packet switches, and there are two general areas which require careful consideration to make optical switching practical in data centers: (1) the properties of physical layer hardware, and (2) the design of the control plane. At the physical layer, the characteristics of optical switching hardware determine the performance, scalability, and cost of the overall network. Optical switches also require a dramatically different approach to control than packet switches, and the control plane can ultimately be a limiting factor in the throughput, latency, and scalability of an optically-switched network. Here, we take a bottom-up approach to discussing the challenges and opportunities for improvement in these two general categories, beginning at the physical layer before moving on to the control plane.

II. Physical Layer Considerations

One of the primary differences between optical and electronic switching is the rate at which each technology can reconfigure its internal forwarding plane. While electronic switches reconfigure quickly enough to route traffic between switch ports at packet-level granularities, optical switches reconfigure much slower—limiting their ability to service latency-sensitive traffic. While faster optical switching technologies exist in the lab, they tend to have higher signal attenuation, requiring more expensive optical transceivers and ultimately impacting the cost and scalability of the network. In this section, we discuss the main physical-layer tradeoffs in more detail, ending with a proposed switch architecture that affords substantial physical layer benefits.

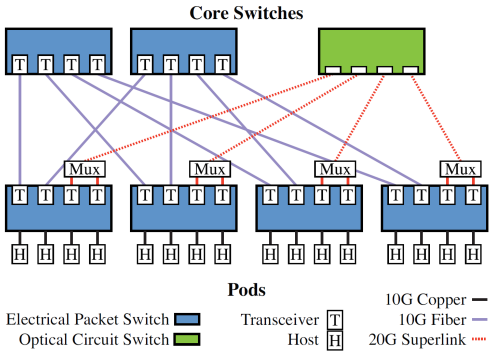

Fig. 1. In the Helios architecture [2], pods of servers are connected with both packet switches and optical circuit switches. Stable traffic is routed over the optical switches at a lower cost per switched bit.

A. Optical crossconnects are slow

One of the early motivations for optical switching in data centers was that traditional oversubscribed folded-Clos packet-switched networks could not support skewed traffic patterns efficiently. This is because their network bandwidth is statically distributed across the topology, leading to congested "hot spots" in the network. Rather than building an expensive, fully-provisioned network to handle skewed traffic, optical switches could be used to dynamically move network bandwidth to where it was needed, alleviating traffic hot-spots.

Figure 1 shows the architecture of Helios [2], one of a number of similar contemporary proposals for optically-switched data centers [3], [4]. The architecture was largely driven by the capabilities of commercially-available optical switching hardware: Helios (and contemporary proposals) employed optical "crossconnect" switches [5], [6], which were originally developed for telecom networks [7]. Crossconnects perform transparent optical switching using microscopic tiltable mirrors to reflect light between the desired input and

output ports. In addition to providing large per-port bandwidth, optical crossconnects scale to hundreds of ports and have low optical signal attenuation, allowing them to connect groups of hundreds of servers (called *pods*) and utilize low-cost optical transceivers (that cannot tolerate high optical signal attenuation).

However, the primary drawback of crossconnects is their relatively slow reconfiguration speed, typically ranging from 10 to 100 milliseconds. Further, because any traffic sent through the switch while it reconfigures is lost, a "dwell" period between reconfigurations is required to achieve high average throughput. For example, if the reconfiguration time is δ, a given circuit configuration must be maintained for 9δ in order to support 90% average throughput. This duty cycle penalty amplifies the effect of the reconfiguration time and limits how quickly new circuits can be established. Providing 90% average throughput with an optical crossconnect means, at best, a new circuit can be established only every 100 ms to 1 second. In Helios, for example, traffic needed to be stable for approximately four or more seconds in order to achieve high average throughput, given the switching and control overheads. Thus, crossconnects can only support highly-stable and/or delay-insensitive traffic.

However, some application traffic in data centers is latency-sensitive, and cannot wait multiple seconds for a new circuit configuration before it is sent. To support this traffic, early proposals used hybrid packet/circuit architectures, employing both optical circuit switching and electronic packet switching to interconnect server pods, as shown in Figure 1. Traffic could be classified at servers to either traverse the circuit network or the packet network, depending on its latency requirements. Assuming the majority of bytes sent over the network had no particular latency requirements, then most of the network core's bandwidth could be switched optically, lowering the total cost of the network.

B. Tradeoffs in speeding up crossconnects

If the optical switch could be made faster, additional cost and power savings might be possible. Faster optical switches could potentially support the burstier communications from top-of-rack (ToR) packet switches, rather than just the aggregated traffic from pods. If optical switches could support a larger fraction of the traffic and be used throughout more of the network, fewer resources would be needed in the packet-switched portion of the hybrid network, saving commensurate cost and power.

Unfortunately, increasing the switching speed of optical crossconnects incurs an overall performance tradeoff. The physical reconfiguration delay of an optical crossconnect is due to the mechanical motion of the tiltable micromirrors responsible for directing light through the switch. The micromirrors used in crossconnects are often referred to as three-dimensional microelectromechanical systems (3D MEMS) because they can tilt along two axes to steer beams of light through a 3D volume. In principle, the micromirror actuators can be redesigned to reconfigure more quickly, but due to physical-layer tradeoffs, doing so reduces the radix of the switch [8]. This, in turn, limits the scalability of the network—i.e. fewer pods can be connected when there are fewer switch ports. Large networks could theoretically be constructed using smaller optical switches by moving to transparent multistage topologies. However, in addition to the direct cost of additional switching hardware, each additional switching stage increases the optical signal attenuation, requiring more expensive transceivers which offsets (or potentially negates) the cost savings of moving to optical switching.

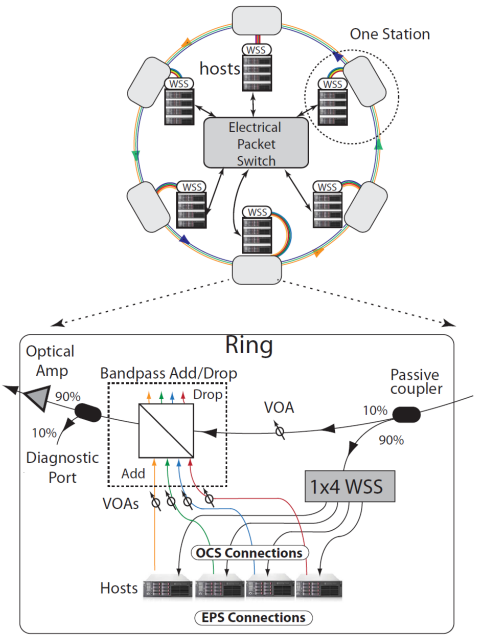

Fig. 2. The Mordia architecture [9] achieves fast switching, but requires a DWDM ring architecture composed of expensive telecom-grade components. Scalability is limited by the spectral bandwidth of the optical amplifiers.

C. Fast wavelength switches have ancillary costs

One option to achieve faster optical switching is to move to a different switch architecture. Wavelength selective switches (WSSes) are another commercially-available technology that, like crossconnects, were originally developed for telecom networks for use in reconfigurable optical add drop multiplexers (ROADMs). Unlike crossconnects, WSSes are often based on 2D MEMS micromirrors which only tilt along one axis, allowing them to switch more quickly than 3D MEMS [10], [11]. However, one axis of tilt restricts the maximum number of optical switch ports, and in general wavelength switching requires a completely different network topology than crossconnect switching.

Figure 2 shows Mordia [9], an optically-switched network based on WSSes capable of microsecond-scale reconfiguration. Similar to many metro-area telecom network designs, its topology is a wavelength division multiplexed (WDM) ring, but, rather than connecting central offices, its add/drop points are ToR switches. The WSS used to implement the prototype could switch 44 dense-WDM (DWDM) wavelength

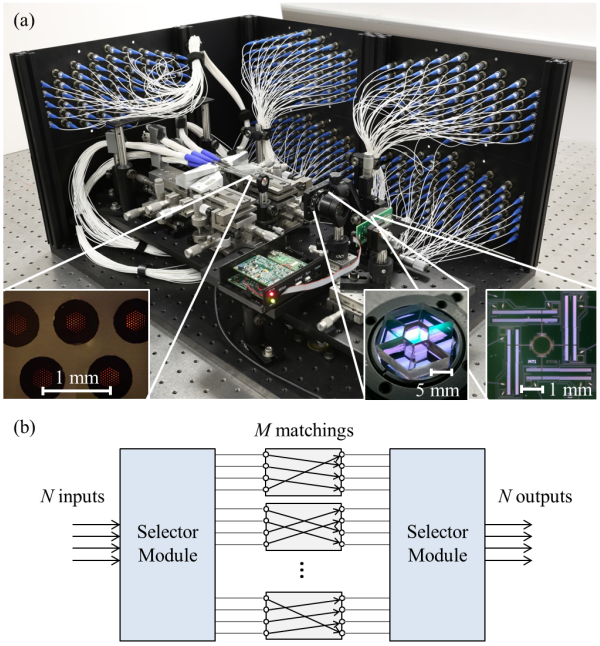

Fig. 3. (a) Prototype optical selector module capable of switching 61 single mode fiber signals to any of 4 61-fiber arrays in 150 μs [12]. (b) Schematic of an optical rotor switch containing an example set of pre-configured input-output port matchings.

channels between 4 output ports, supporting 4 endpoints per WSS station and up to 44 stations. The WSS enabled a fast system-level reconfiguration speed of 12 μs, several orders of magnitude faster than optical crossconnects.

However, from a practical perspective, Mordia's design sacrifices cost-efficiency by relying on telecom-grade equipment. The DWDM transceivers are orders of magnitude more expensive than the single-wavelength datacom transceivers typically used in data centers. Optical amplifiers are also a significant cost element not required in the point-to-point optical links found in traditional data centers. These factors offset the cost advantage of using optical switching. Further, Mordia relies on adding additional wavelengths to scale to larger network sizes, but the spectral bandwidth of the optical amplifiers limits the number of channels that can be added without moving to a blocking topology.

D. Advantages of partially-configurable optical switches

Both Helios and Mordia were based on commercially-available optical switches designed for telecom networks: crossconnects and wavelength switches, respectively. Because they were optimized for a different use-case, neither type of switch met the full set of physical layer performance metrics required to be practical in data center networks: (1) fast reconfiguration (needed to service traffic with stricter latency budgets), (2) large radix (necessary to connect many endpoints), and (3) low signal attenuation (required to use low-cost optical transceivers).

However, it is possible to achieve all three of the above physical layer metrics by trading off another aspect of the switch: arbitrary configurability [12]. The previous optical networks all enable any port to be connected to any other port at any time (subject to the reconfiguration delay) making them arbitrarily configurable, but at a performance or monetary cost. A *partially-configurable* switch, on the other hand, is only able to connect certain sets of ports at a given time, but, by virtue of its limited functionality, enables enhanced physical layer performance.

One promising approach to implementing a partially-configurable switch (illustrated in Figure 3(b)) involves physically separating the switching and routing functions. For switching, a *selector module* can be used to connect all N input ports to one of $M < N$ N-port arrays. Each of the M N-port arrays feeds into a different static, pre-configured *matching*, which performs the routing of input ports to output ports. Finally, a second selector module (operating in reverse), connects the output of the currently selected matching to the output ports of the overall switch. This implementation sidesteps the scaling limitations in Section II-B, which relied on the 3D-MEMS switching elements to implement both switching and routing concurrently, which ultimately limited their reconfiguration speed.

Figure 3(a) shows a prototype optical selector module, capable of coupling 61 single-mode fiber channels to any one of four matchings with a reconfiguration speed of 150 μs (100× faster than a crossconnect with the same number of ports). Optical modeling indicates that with custom components, partially-configurable optical switches can scale to thousands of ports with low signal attenuation and reconfiguration speeds on the order of 10 μs [12].

E. A partially-configurable network topology

While partially-configurable optical switches deliver many physical layer benefits, they cannot be used in conventional network topologies (e.g. folded Clos), which are based on nonblocking crossbar switches.

Instead, RotorNet [13], shown in Figure 4(b), is one approach to using partially configurable switches (called rotor switches in this context) to build an inter-rack data center network. RotorNet is based on the observation that only $N-1$ matchings are needed to provide full connectivity between a set of N endpoints (e.g. racks of servers in this case). Instead of connecting all endpoints with a single rotor switch containing all $N-1$ matchings (Figure 4(a)), the matchings can be partitioned across a set of parallel rotor switches so each contains $\ll N-1$ matchings (Figure 4(b)). This design leverages the features of partially-configurable switches—that they can only provide a limited set of matchings—while also ensuring fault tolerance and providing concurrent connectivity between racks.

F. Physical layer takeaways

Early investigations into using optical switching in data centers demonstrated the potential for cost and power savings, but relied heavily on the assumption that most flows in the network be stable for second-long timescales. This design

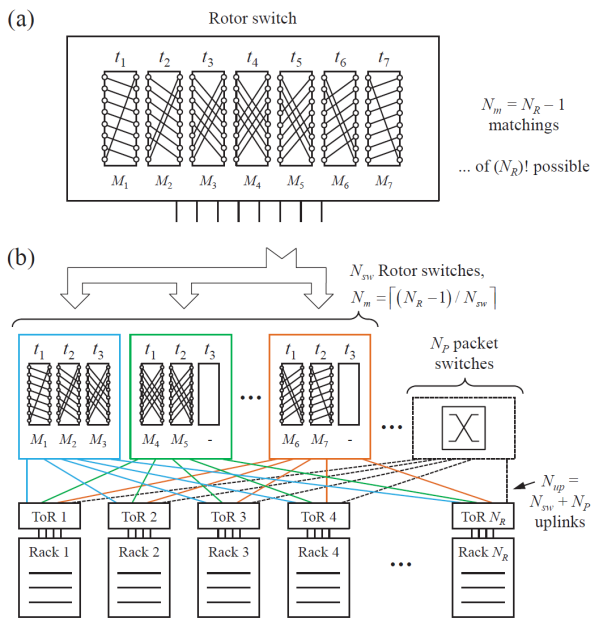

Fig. 4. (a) Connecting all endpoints with a single optical rotor switch is not practical, so (b) RotorNet [13] uses a set of parallel rotor switches to connect racks of servers. A hybrid architecture is used to support latency-sensitive traffic.

assumption limited the benefits conferred by optical switching in networks with less stable traffic characteristics. Faster switching is possible with optical wavelength switches, but wavelength routed networks require expensive telecom-grade components and have scaling limitations. Sacrificing arbitrary switch configurability enables fast, low-attenuation, and cost-effective switching between a large number of ports, but must be paired with careful topology design to ensure connectivity between network endpoints.

III. Control Plane Considerations

In addition to the physical-layer considerations discussed above, optical switching requires a different control paradigm than packet switching. The two primary reasons for this are that, unlike packet switches, optical switches cannot (1) buffer packets, or (2) inspect packets. The lack of buffering means that data transmissions must be synchronized with the (re)configuration of optical switches. The lack of packet inspection means that routing decisions must be made external to the optical switch by a separate control plane.

A. Transmission synchronization

Packet networks allow traffic to be sent at nearly any time because bursts can be absorbed by buffers in the network and packet switches can quickly multiplex packets destined for different endpoints. In optically-switched networks, traffic sent at the wrong time could be dropped or forwarded to the wrong endpoint. Microsecond-scale optical switching imposes tight bounds on the level of synchronization required to efficiently utilize the network. The problem is also complicated by the need to support a hybrid packet/circuit architecture, since traffic destined for the optical switch should be treated differently than that traversing the packet network.

REACToR [14] employed an approach which allowed pre-classified traffic from end hosts to be sent over either a packet-switched or optically-switched network. Unlike the hybrid design in systems like Helios, which could afford to use relatively slow mechanisms due to the long optical switching delay, REACToR was designed to operate with microsecond-scale optical switching. REACToR used a fast control protocol to synchronize end host transmissions with the optical switch configuration and set per-queue rate limits. Rate limiting was necessary to ensure that packet- and circuit-switched traffic efficiently shared links without overrunning link capacity. Ethernet pause frames were used to quickly pause and un-pause priority flow control (PFC) queues at the end hosts so packets were never emitted while the optical switch was reconfiguring. End-host queues containing traffic destined for the packet switch were never paused, providing on-demand low-latency connectivity through the packet network. For REACToR's 10-Gb/s links, the serialization latency of a 1500-byte packet added more packet emission variance than starting or pausing queues using PFC pause frames (for a total variance of 2.5 µs). Using a predetermined schedule, and by returning ACKs over the packet network, REACToR was able to send traffic over Mordia without negatively impacting stock TCP performance.

Still, today's implementations of PFC are limited to a handful of traffic classes, making it challenging to scale REACToR's implementation to a large number of network endpoints. Other approaches to rate limiting based on software may provide better scalability [15], but their performance in optically-switched networks remains undetermined.

B. Real-time scheduling

Synchronization is only part of the control problem in optically-switched networks. The other major challenge is demand collection and scheduling. Because optical switches cannot inspect packets in flight, they cannot make routing decisions. Instead some external entity must make those decisions and reconfigure the optical switch(es) appropriately at different points in time.

1) Batch scheduling: As was the case for synchronization, faster optical switching makes real-time scheduling more difficult. In systems like Helios, the scheduler was responsible for measuring the prevailing traffic demand, computing an optimal (or near-optimal) optical circuit assignment to serve that demand, and reconfiguring optical switches to implement those circuits. In the 8-port Helios prototype, these processes took about 300 ms; for large-scale networks, one iteration of the control loop could take multiple seconds. This matched well with the relatively slow reconfiguration speed of an optical crossconnect, but is much too slow to make effective use of microsecond-scale optical switches, and would limit their ability to support bursty, short-lived traffic.

One proposed approach to reduce the effective overhead of scheduling is to move away from "hot-spot scheduling" to a more generalized form of scheduling called traffic matrix

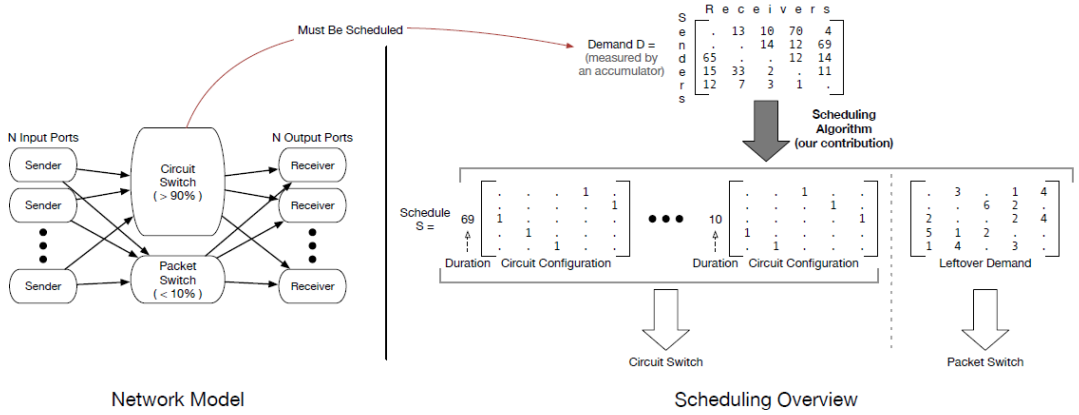

Fig. 5. Illustration showing a general hybrid network architecture and the Solstice scheduling algorithm [16].

scheduling (TMS) [17], [18]. Rather than computing circuit assignments one at a time, TMS is a batch process that computes a time sequence of circuit assignments along with their durations. This allows the control loop iterations to be longer than each circuit assignment at the cost of using demand information that is slightly stale. Still, the time to compute one iteration in TMS (consisting of a Sinkhorn matrix scaling operation and a Birkhoff-von Neumann decomposition, both $O(n^2)$) can still be a limiting factor for large networks. Also, TMS was not designed to support a hybrid architecture, meaning all traffic would transit the optical switch. Given TMS schedule lengths of hundreds of microseconds, traffic generated by latency-sensitive applications might still suffer undesirable delays.

2) Near-optimal hybrid batch scheduling: Improving on TMS scheduling and extending it to hybrid networks required a more detailed investigation. Solstice [16] formalized the hybrid scheduling problem, developed a purpose-built hybrid network scheduler, and investigated its practical utility.

Figure 5 shows an abstract model of a hybrid network and an overview of the Solstice algorithm. Hybrid networks impose additional constraints on scheduling than those found in a purely packet-switched context, including the non-trivial circuit reconfiguration time and the need to partition traffic between the packet and circuit networks. These factors made it difficult to extend existing scheduling algorithms to cover this new setting.

The goal of Solstice was to quickly find a schedule that minimized the time (including circuit reconfigurations) to serve a traffic demand, while efficiently using a 10:1 underprovisioned packet network to carry traffic that was not well-served by circuits. Finding the optimal solution is NP-hard ($O(n!)$ complexity), so Solstice used a number of heuristics to significantly reduce computational complexity (to $O(n^3 \log^2 n)$) while only trading off 14% in bandwidth utilization compared to an optimal schedule. Even with these optimizations, Solstice takes about 5 ms to compute the schedule for a 64-port network on a 2.8 GHz CPU. This is slightly longer than the duration of the actual schedule it produces for 64 ports, which consists of 24 circuit configurations and lasts 3 ms. Executing Solstice on a faster CPU or a custom ASIC could reduce runtime, but given that complexity scales as $O(n^3)$, it may not be feasible to compute schedules in real time for networks with more than about 100 ports. Eclipse [19] is another hybrid scheduler proposed shortly after Solstice with somewhat higher performance, but is also $O(n^3)$ in complexity, indicating its runtime would be comparable to Solstice.

Solstice improved network utilization by 2.9× compared to using standard scheduling algorithms not designed for hybrid networks, and delivered within 14% of the optimal throughput at a scale of 128 ports. Still, running Solstice in real time in a large network presents a number of scaling challenges. Further, Solstice did not provide a demand collection mechanism to feed the scheduler with up-to-date network-wide traffic demand. Demand collection presents another challenge to realizing a practical, scalable, and closed-loop hybrid network controller.

C. Using a static schedule

One promising approach to sidestep the scaling challenges of real-time scheduling and demand collection is to use a predetermined, static schedule.

RotorNet [13] uses such an approach, where, rather than configuring the optical switches in response to traffic demand, the optical switches "rotate" through their limited set of pre-defined matchings in a fixed, pre-determined sequence, spending an equal amount of time in each matching. This static schedule provisions uniform inter-rack bandwidth across time. Rather than requesting a circuit be set up in response to new traffic demand, servers buffer traffic and wait for the appropriate opportunities to send, either directly via one-hop paths or indirectly through an intermediate rack via two-hop paths. Critically, all control is peer-to-peer and no centralized demand collection or schedule computation is required. Perhaps surprisingly, RotorNet gives up at most a factor of two in bandwidth relative to an ideal network for relying on such simple control. For traffic patterns that are either heavily skewed or nearly uniform, RotorNet delivers close

to optimal throughput (scaled only by the duty cycle of the optical switches).

D. Control plane takeaways

Fast optical switching and large network sizes means synchronization, demand collection, and scheduling may be the ultimate system-level bottlenecks in optically-switched networks. Tight synchronous transmission of data can be achieved using priority flow control mechanisms, but scalability will likely favor a combination of software and hardware based rate limiters. Fast, accurate real-time scheduling in large hybrid networks remains a challenging problem, motivating the move to pre-determined, static schedules. Static scheduling also sidesteps the challenge of fast, centralized collection of network-wide demand, because the schedule is decoupled from the traffic demand.

In addition to scalable rate limiting, one remaining opportunity for improvement is that RotorNet still relies on a hybrid architecture to service traffic with sub-millisecond latency requirements. Supporting low-latency traffic without a hybrid network is a subject of ongoing investigation.

IV. Conclusion

Optical switching promises to support the growing bandwidth requirements of data center networks, but its widespread deployment has been gated by a number of technical challenges associated with its circuit-switched characteristics.

RotorNet serves as one example indicating that overcoming these challenges requires a joint design approach incorporating redesign at both the physical and control layers. Rotor switches are compatible with inexpensive datacom transceivers while still providing fast switching and large radix, addressing cost and scalability challenges. However, these partially-configurable switches would not be useful on their own in a standard network, necessitating topology and control plane changes to effectively use them. Ultimately, those changes resulted in a simpler control plane, eliminating one of the primary system-level bottlenecks facing fast optical switching.

Ongoing efforts to eliminate the need for a hybrid architecture and achieve scalable end host rate limiting may push optical switching into mainstream deployment in data centers.

Acknowledgements

We wish to thank those who made this line of research possible, including David Andersen, Hamid Bazzaz, Yeshaiahu Fainman, Nathan Farrington, Nicholar Feltman, Joseph Ford, Alex Forencich, Tara Javidi, Michael Kaminsky, Michael Kozuch, Conglong Li, He Liu, Feng Lu, Rob McGuinness, Matthew K. Mukerjee, T.s. Eugene Ng, George Papen, Sivasankar Radhakrishnan, Tajana Rosing, Arjun Roy, Stefan Savage, Glenn M. Schuster, Srinivasan Seshan, Richard Strong, Vikram Subramanya, Pang-Chen Sun, Malveeka Tewari, Amin Vahdat, Geoffrey M. Voelker, Chang-Heng Wang, Guohui Wang. This work was supported by the National Science Foundation (CNS-1564185, CNS-1553490, and CNS-1629973).

References

[1] B. G. Bathula, A. L. Chiu, R. K. Sinha, and S. L. Woodward, "Routing and regenerator planning in a carrier's core roadm network," in *Optical Fiber Communications Conference and Exhibition (OFC), 2017*. IEEE, 2017, pp. 1–3.

[2] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proceedings of the ACM SIGCOMM Conference*, New Delhi, India, Aug. 2010.

[3] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 327–338.

[4] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: a topology malleable data center network," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010.

[5] Calient, "Calient," http://www.calient.net/, 2018.

[6] Glimmerglass, "Glimmerglass," http://www.glimmerglass.com/, 2018.

[7] J. Kim, C. Nuzman, B. Kumar, D. Lieuwen, J. Kraus, A. Weiss, C. Lichtenwalner, A. Papazian, R. Frahm, N. Basavanhally *et al.*, "1100 x 1100 port mems-based optical crossconnect with 4-db maximum loss," *IEEE Photonics Technology Letters*, vol. 15, no. 11, 2003.

[8] W. M. Mellette and J. E. Ford, "Scaling limits of mems beam-steering switches for data center networks," *J. Lightwave Technol.*, vol. 33, no. 15, pp. 3308–3318, Aug 2015.

[9] N. Farrington, A. Forencich, G. Porter, P.-C. Sun, J. Ford, Y. Fainman, G. C. Papen, and A. Vahdat, "A multiport microsecond optical circuit switch for data center networking," *IEEE Photonics Technology Letters*, vol. 25, no. 16, pp. 1589–1592, 2013.

[10] J. E. Ford, V. A. Aksyuk, D. J. Bishop, and J. A. Walker, "Wavelength add-drop switching using tilting micromirrors," *Journal of lightwave technology*, vol. 17, no. 5, p. 904, 1999.

[11] Nistica, "Nistica," http://www.nistica.com/, 2018.

[12] W. M. Mellette, G. M. Schuster, G. Porter, G. Papen, and J. E. Ford, "A scalable, partially configurable optical switch for data center networks," *Journal of Lightwave Technology*, vol. 35, no. 2, pp. 136–144, 2017.

[13] W. M. Mellette, R. McGuinness, A. Roy, A. Forencich, G. Papen, A. C. Snoeren, and G. Porter, "RotorNet: a scalable, low-complexity, optical datacenter network," in *Proceedings of the ACM SIGCOMM Conference*, Los Angeles, California, Aug. 2017.

[14] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter, "Circuit Switching Under the Radar with REACToR," in *Proceedings of the 11th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Seattle, WA, Apr. 2014, pp. 1–15.

[15] A. Saeed, N. Dukkipati, V. Valancius, L. V. The, C. Contavalli, and A. Vahdat, "Carousel: Scalable traffic shaping at end hosts," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 404–417.

[16] H. Liu, M. K. Mukerjee, C. Li, N. Feltman, G. Papen, S. Savage, S. Seshan, G. M. Voelker, D. G. Andersen, M. Kaminsky, G. Porter, and A. C. Snoeren, "Scheduling techniques for hybrid circuit/packet networks," in *Proceedings of ACM CoNEXT*, Heidelberg, Germany, 2015.

[17] G. Porter, R. Strong, N. Farrington, A. Forencich, P.-C. Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," in *Proceedings of the ACM SIGCOMM Conference*, Hong Kong, China, Aug. 2013.

[18] N. Farrington, G. Porter, Y. Fainman, G. Papen, and A. Vahdat, "Hunting Mice with Microsecond Circuit Switches," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks (HotNets-XI)*, Redmond, WA, Oct. 2012.

[19] S. B. Venkatakrishnan, M. Alizadeh, and P. Viswanath, "Costly circuits, submodular schedules and approximate carathéodory theorems," *Queueing Systems*, vol. 88, no. 3-4, pp. 311–347, 2018.