JOURNAL OF
Optical Communications
and Networking

Optical networking within the Lightwave Energy-Efficient Datacenter project [Invited]

William M. Mellette,[1,2,3] Alex Forencich,[1] Jason Kelley,[1] Joseph Ford,[1]
George Porter,[2] Alex C. Snoeren,[2] and George Papen[1,*]

[1]Department of Electrical and Computer Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA
[2]Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA
[3]inFocus Networks, San Diego, California 92093, USA
*Corresponding author: gpapen@eng.ucsd.edu

Received 2 July 2020; revised 26 August 2020; accepted 13 September 2020; published 8 October 2020 (Doc. ID 401903)

The Lightwave Energy-Efficient Datacenter (LEED) project within the ARPA-e ENLITENED program is developing novel energy-efficient multichannel lightwave networks. These networks are enabled by a new optical "rotor" switch that can reconfigure the network topology in less than 20 μs and a field-programmable-gate-array-based network interface controller called Corundum that can provide precise network-wide synchronization of packets admitted into the lightwave network. Here we review the optical networking research within LEED and discuss future directions.

https://doi.org/10.1364/JOCN.401903

1. INTRODUCTION

The steady growth in computational bandwidth in datacenters and high-performance computers drives an ever-increasing need for networking bandwidth. Energy efficiency is increasingly difficult to maintain as these systems scale, a challenge that is exacerbated by the impending scaling limits of electronics. The Lightwave Energy-Efficient Datacenter (LEED) project within the ARPA-e ENLITENED program aims to address this issue by replacing today's electronically switched networks with a higher-bandwidth, energy-efficient, optically switched network. The goal of the optical networking research within the LEED project is to develop technologies that can lead to datacenter energy-efficiency improvements by tightly integrating: (a) an energy-efficient scalable optical circuit-switched architecture that has a deterministic, scalable distributed control plane and (b) a low-loss, scalable, rate-agnostic optical "rotor" switch. The combination of these technologies has the potential to provide substantial performance and energy-efficiency improvements compared to cost-comparable conventional networks.

2. BACKGROUND

There has been over a decade of extensive research on optical networking for datacenters. Numerous devices with impressive performance have been developed that vary in port count, switching speed, and insertion loss. (For a review, see Chapter 14 of [1].) Yet, despite this impressive research record, optical networking architectures that react on a per-flow or per-packet basis have not been deployed. In this background section, we provide a brief discussion of the necessary conditions for the practical implementation of an optical circuit switch (OCS) and how the work within the LEED project addresses these issues.

A. Electrical Packet Switching and Optical Circuit Switching

All networked switches, either electrical or optical, have a control plane that controls the state of the switch and a data plane over which the data is routed. Ideally, the speed of the control plane is matched to the rate of changing traffic demands in the data plane. For example, a large port-count, millisecond-speed OCS such as a 3D beam-steering micro-electromechanical systems (MEMS) switch is well matched to the speed of a software-defined network (SDN). For this kind of network, the network reconfiguration is derived from time-averaged or anticipated datacenter workloads with changes occurring on time scales ranging from minutes to months. This kind of dynamic network provisioning or topology management does not require "reactive" optical switching because the state of each switch does not react to individual packets or flows.

The control plane for a reactive OCS, which dynamically reconfigures based on per-flow or per-packet information, requires several network elements similar to those of an electrical packet switch (EPS). To understand these network elements, consider a basic EPS as shown in Fig. 1(a).

The incoming packets to be routed are converted from the optical domain into the electrical domain in fixed data-rate

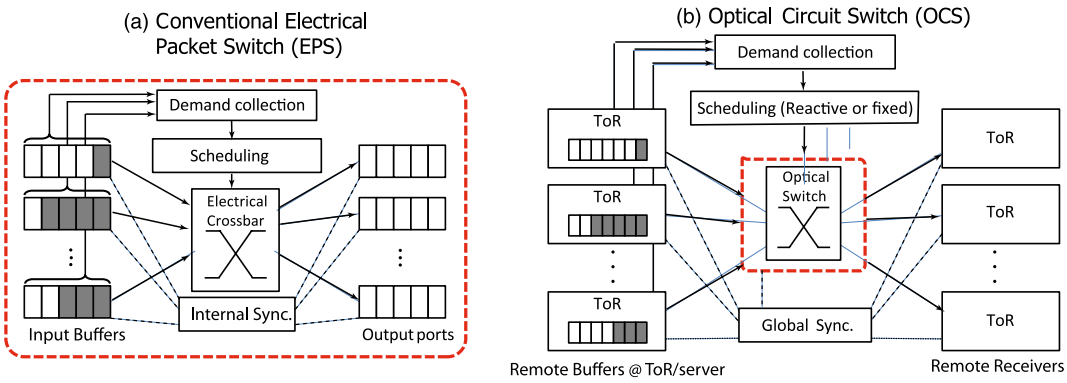

Fig. 1. (a) Electrical packet switch and (b) optical circuit switch. The elements within a switch are shown within the dashed red line (adapted from [2]).

transceivers and then queued in an electronic buffer at an input port to the switch. The header of each packet is read, and a local switch configuration is calculated that routes each input packet to an appropriate output port [3]. The buffering, crossbar switching, and synchronization are done locally within the switch enclosure, which is represented by the red dashed line in Fig. 1(a).

Now consider the optical switch shown within the red dashed line in Fig. 1(b). In this case, several other network elements are required for a reactive control plane that are not local to the optical switch. For example, the buffers that contain the data to be routed by the optical switch are not local because current practical optical switches cannot store packets in the optical domain. Further, practical optical switches cannot read packet headers and thus cannot make local routing decisions. In other words, optical switches simply create light paths or circuits between input ports and output ports.

Because it does not require the transceivers or electronics needed for an EPS, an OCS is rate agnostic in the sense that the data rate is limited only by the spectral bandwidth of the switch's constituent optical components. However, the lack of local buffers and packet inspection means that the state of every OCS in the network must be determined by a global control plane. Such a control plane could set the state of every switch in the network using information collected from each network end point or, alternatively, use a predetermined schedule of switch states. In either case, the data transmissions from all end hosts must be synchronized so that data follow the correct paths through the switches. At the physical layer, precise synchronization of the data plane to the control plane must ensure that the data are sent when the switches are in the correct state. Depending on the switching speed, they may also require burst-mode receivers [4,5].

The technologies used for optical switching can be compared using the metrics of port count, switching speed, crosstalk, and loss. Large port-count switches are desirable from an architectural perspective because they lead to "flattened" networks with fewer hops between end points. Low-loss, low-crosstalk switches can use commercial off-the-shelf (COTs) transceivers without the need for optical amplification. Faster switches can serve quickly changing network traffic conditions more efficiently.

Switches based on silicon photonics [6] have fast switching speeds but are currently limited in port count because of crosstalk and polarization-dependent coupling loss. MEMS-actuated-waveguide switches [7] balance switching speed and port count, scaling to hundreds of ports with low on-chip loss and excellent crosstalk. However, as planar devices, they still must address the non-negligible, and typically polarization-dependent, coupling loss.

Larger port-count switches with lower overall fiber-to-fiber loss can be built using non-planar free-space technologies based on piezo-electronic actuation [8] or 3D beam-steering MEMS [9,10]. These technologies have switching speeds measured in tens of milliseconds. These commercially available switches can be used in conjunction with software-defined networking to reconfigure the overall topology on long-duration time scales to match the time-averaged or anticipated datacenter workloads. However these switches cannot dynamically reconfigure the network on the time scale associated with an individual flow of packets or an isolated packet.

The optical networking research within LEED is based on a different kind of optical "pinwheel" rotor switch, which is described in Section 6. Similar to existing 3D MEMS and piezoelectric switches, the rotor switch can use COTS transceivers and has the potential to be scaled to thousands of ports [11]. This enables the development of large-scale flattened networks using standards-based optical interconnect technology. However, in contrast to commercial optical switches, the rotor switch is about three orders of magnitude faster. In comparison to polarization-sensitive planar-switching technologies such as MEMS-actuated-waveguide switches or silicon-photonics-based switches, the rotor switch is slower, but has lower overall loss and does not require polarization diversity.

B. Organization of the Paper

The unique combination of switch speed, switch loss, and switch port for a rotor switch has enabled the development of two novel architectures called RotorNet and Opera. Accordingly, the optical networking research within the LEED project presented in this paper is organized as follows. In Section 3, the RotorNet architecture [2] is presented. This is a parallel optical network that uses optical "rotor" switches

and a fixed time-division multiple access (TDMA) schedule to provide a known sequence of direct connections between all network endpoints over time.

Section 4 discusses the Opera architecture [12], which can be viewed as an extension of RotorNet. This architecture specifies that the time sequence of connectivity be a sequence of expander graph topologies. This ensures that every end point has an "open" (albeit indirect, or multi-hop) network connection to every other end point at every time instant. These "open" connections allow latency-sensitive packets to be forwarded immediately without waiting for direct connections to occur in the schedule.

Section 5 discusses the precise synchronization of the data plane to the control plane to ensure that data flows are sent when the optical switch is in the correct state. This synchronization is accomplished using a field-programmable gate array (FPGA)-based network interface controller (NIC) called Corundum [13].

The first five sections provide background for several new results presented in Section 6, which describes the development of a "rotor" switch and the use of Corundum to characterize its performance. These new results include a novel characterization tool for connection-level bit-error-rate (BER) measurements of an optical switch and running a standard Linux performance tool called *iperf* over RotorNet using commercially available transceivers. Finally, Section 7 discusses future work and provides a conclusion.

3. ROTORNET ARCHITECTURE

RotorNet is a parallel optical network designed to overcome the challenges of optical circuit switching discussed in Section 2.A. RotorNet avoids centralized scheduling because it does not attempt to reconfigure the optical switches to match network traffic conditions. Instead, each switch independently rotates through a predetermined, fixed set of network configurations in a round-robin fashion. Because the network configurations are predetermined, RotorNet completely eliminates the need for a centralized control plane and does not require demand estimation, schedule computation, or schedule distribution.

RotorNet has bounded delivery time and bounded host buffering requirements and is robust to link and switch failures. The network configurations are physically implemented via rotor switches. The design and characterization of a rotor switch are discussed in Section 6.

A. How RotorNet Works

In RotorNet, an individual network end point, which may be a top of rack switch (ToR) or a server, is connected to multiple rotor switches via separate uplinks. Each rotor switch cycles through a small set of network configurations—called "matchings"—in a round-robin manner, irrespective of instantaneous traffic demands, as shown in Fig. 2(a). Let N_R denote the total number of end points in the network, which for simplicity is taken to be a ToR. While there are $N_R!$ possible matchings between ToRs, only $N_R - 1$ matchings are required

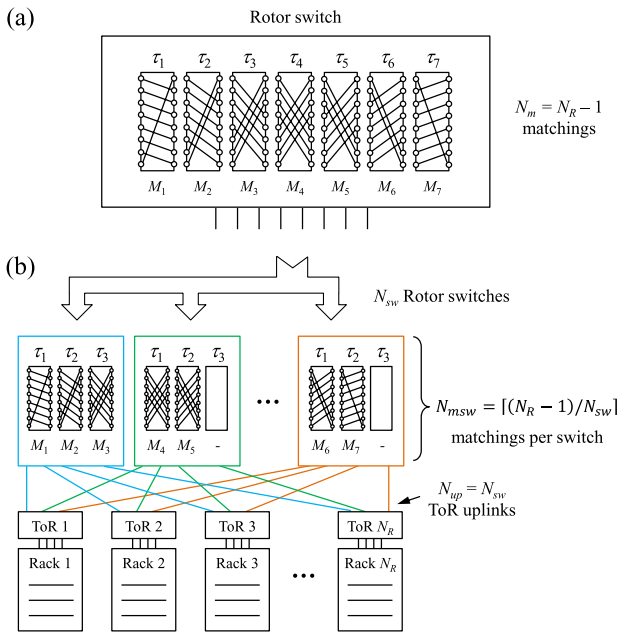

Fig. 2. (a) RotorNet cycles through N_m fixed network configurations or matchings in a round-robin fashion. (b) Total number of required matchings $N_R - 1$ to achieve full connectivity is distributed across N_{sw} rotor switches (from [2]).

to achieve full connectivity between ToRs. With $N_R - 1$ disjoint matchings, each ToR will have a direct connection to every other ToR within one full cycle of matchings.

RotorNet partitions these matchings across N_{sw} rotor switches, as shown in Fig. 2(b). The number of matchings N_{msw} per rotor switch is then given by $N_{msw} = \lceil (N_R - 1)/N_{sw} \rceil$. The use of parallel rotor switches each with a small set of matchings per switch significantly improves the overall time to cycle through all of the matchings.

RotorNet provides one direct connection between each pair of end points every full cycle of matchings. A basic communication protocol could simply buffer traffic at end hosts until a direct connection through an optical switch to the destination is available. Such a protocol would be ideal for uniform or nearly uniform traffic. However, traffic in datacenters is often heavily skewed, which would result in unused capacity elsewhere in the network. Instead, we developed a protocol called RotorLB, which detects skewed traffic conditions and uses two-hop routing (i.e., valiant load-balancing) to improve throughput to within 50%–100% of ideal throughput regardless of the degree of skew in the traffic pattern. One way to interpret this is that RotorNet pays at most 50% in terms of throughput to avoid the scheduling and control complexities and slower switch reconfiguration speeds associated with reactive optical circuit switching networks, which ultimately allows RotorNet to support a wider range of workloads with lower latency. Section 4 discusses alternate ways to send traffic in the Opera architecture.

B. Scalability

RotorNet's physical layer scalability stems from its relaxation of switch-hardware requirements. In particular, rotor switches

need only differentiate between a small number of matchings, rather than a large number of ports, as is the case for a standard switch. A standard N-port OCS implements a crossbar, meaning it can be configured to any of $N!$ matching patterns. This flexibility limits the switching speed and radix of beam-steering MEMS and piezoelectric OCSs because the physical requirements of each switching element are coupled to the switch radix [14]. As a result, commercially available OCSs have radices on the order of 300 ports and reconfiguration times of tens to hundreds of milliseconds. Detailed physical-optics design work has shown that rotor switches, on the other hand, can scale to thousands of ports with reconfiguration times of tens of microseconds, while maintaining an insertion loss comparable to a commercial OCS [11].

Connecting thousands of ToRs together with conventional OCSs requires those OCSs be cascaded in a multi-stage optical topology, which introduces significant signal attenuation. For example, a three-stage topology would be required to support more than about 300 racks, which would incur a loss of about 9 dB with commercial OCSs. This higher signal attenuation, in turn, requires higher sensitivity optical transceivers or optical amplification, which would almost certainly preclude using OCSs in datacenters. Similarly, MEMS-actuated-waveguide switches [7] that employ O(100)-port OCSs to connect pods (instead of racks) replace less of the electronic network and currently have high insertion loss, limiting their cost effectiveness.

In addition to multi-stage insertion loss concerns, commercially available OCSs reconfigure too slowly to support most datacenter traffic. Only large traffic flows with serialization delays of hundreds of milliseconds to seconds can use a conventional OCS efficiently. RotorNet supports flows with millisecond-scale serialization delays efficiently (a 100× improvement), and the Opera architecture extends this design with support for microsecond-scale flows over an all-optical switching core, eliminating the need for a less-cost-effective hybrid network.

While rotor switching can address loss, radix, and switching speed concerns, it does impose architectural constraints different from commercial OCSs. Specifically, larger networks require a larger number of rotor switches to keep the total time to cycle through the matchings low, in turn requiring more ports at ToR switches. Coincidentally, the industry is moving toward high-radix packet switching. For example, Facebook recently employed 128-port × 100 Gb/s switches rather than 32-port × 400 Gb/s switches in their new F16 fabric. This trend enables the larger number of rotor switches needed for large-scale RotorNet deployments.

4. OPERA ARCHITECTURE

Opera [12] is an extension of the RotorNet architecture that efficiently provisions network bandwidth for "bulk" traffic while ensuring low-latency delivery for the remaining (small fraction) of the traffic that cannot tolerate added delays. While the basic RotorNet architecture requires some amount of electronic switching to handle latency-sensitive traffic [as shown in Fig. 2(b)], Opera routes both bulk and latency-sensitive traffic using an entirely optically switched network core. This section summarizes our recent work on the implementation of the Opera architecture.

Like the RotorNet architecture discussed in Section 3, Opera employs a time sequence of predetermined direct optical connections between end points. However, Opera specifies the connectivity such that at all times the set of active connections forms an expander graph. Expander graphs are optimal in the sense that they have the lowest possible expected path length [15], meaning there are many potential short paths from a given source to a particular destination, which makes them desirable for latency-sensitive networks. They also have good fault-tolerance properties because if a switch or link fails, there are likely to be alternative paths through the network.

A. Enabling Low-Latency Connectivity

A key metric of network performance is flow completion time (FCT). Providing low FCTs for latency-sensitive traffic requires an "open connection" between every pair of end points at every instant in time. In a network with multiple circuit switches, this is not guaranteed when all switches reconfigure simultaneously. To avoid this scenario and allow for low-latency packet delivery, Opera leverages the parallelism of the basic RotorNet topology and offsets the reconfigurations of circuit switches, as shown in Fig. 3. (Larger networks with many rotor switches may have more than one switch reconfigure at the same time.)

Referring to Fig. 2, there are N_{sw} uplinks for each ToR. Each uplink is connected to a single rotor switch. The complete inter-ToR network topology is then the union of N_{sw} matchings. When the matchings are random and $N_{sw} > 3$, the complete inter-ToR topology is an expander graph with high probability [16]. Moreover, even when a switch is reconfiguring, there are still $N_{sw} - 1$ active matchings. This means that when $N_{sw} > 4$, the complete inter-ToR network topology will still be an expander with high probability, no matter which rotor switch is reconfiguring.

In Opera, the direct links in each expander graph provide bandwidth-efficient connectivity for bulk traffic in a manner similar to RotorNet. However, during the "dwell time" in each expander graph topology, Opera can use indirect routes through one or more intermediate ToR to send latency-sensitive traffic. This means that on a per-packet basis, Opera can either (1) immediately send a packet over the current

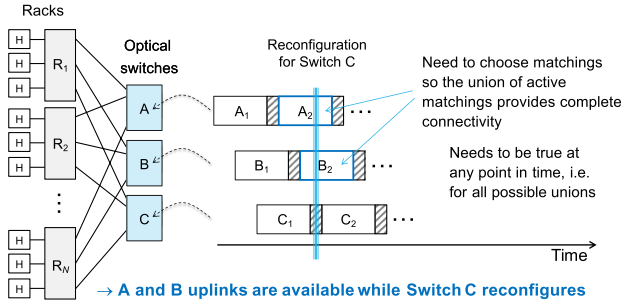

Fig. 3. Offsetting the reconfigurations of rotor switches permits continuous network connectivity for latency-sensitive traffic. When switch C is reconfiguring, connections from switch A and switch B still provide network connectivity.

network configuration, incurring a modest "bandwidth tax" on this small fraction of traffic, or (2) buffer the packet and wait until a direct link is established to the final destination, providing a bandwidth efficient connection or, equivalently, a low-bandwidth-tax connection. One way to determine the "cutoff" for which traffic should be sent using each approach is to consider the flow serialization delay. Flows that have a serialization delay longer than the time needed to cycle through all matchings are best classified as "bulk" and buffered to be sent using the RotorLB protocol, whereas flows with shorter serialization delay are best classified as "latency sensitive" and sent immediately over indirect paths. As discussed in more detail in [12], the time to cycle through all matchings is on the order of a few milliseconds in Opera networks. The net result is a *single* optically switched network fabric that can support both bulk and low-latency traffic, in contrast to the separate optical and electronic networks using the hybrid approach shown in Fig. 2(b).

B. Example Opera Network

Figure 4 shows a small-scale Opera network from [12]. Each of the eight ToRs has four uplinks to four different rotor (circuit) switches. By forwarding traffic through those ToRs, traffic can reach any ToR to which they, in turn, are connected. ToR forwarding is handled using time-indexed routing tables, and is described in more detail in [12]. Each rotor switch has two matchings, labeled A and B, and the complete set of eight matchings is disjoint. In this example topology, any ToR pair can communicate by utilizing any set of three matchings, meaning complete connectivity is maintained regardless of which matchings happen to be implemented by the switches at a given time.

Figure 4 shows two network-wide configurations. In Fig. 4(a) switches 2–4 are implementing matching A, and in Fig. 4(b), switches 2–4 implement matching B. In both cases, switch 1 is unavailable because it is reconfiguring. Referring to the figure, racks 1 and 8 are connected directly by the configuration shown in Fig. 4(b), and so the most bandwidth-efficient way to send bulk data from 1 to 8 would be to wait until matching B is instantiated in switch 2, and then to send the data through that circuit; such traffic would arrive at ToR 8 in a single hop. On the other hand, low-latency traffic from ToR 1 to ToR 8 can be sent immediately, e.g., during the configuration shown in Fig. 4(a), and simply take a longer path to get to ToR 8. The traffic would hop from ToR 1 to ToR 6 (via switch 4), then to ToR 8 (via switch 2), using two hops to complete the route. This two-hop path is less bandwidth efficient and therefore has a higher bandwidth tax. This is the fundamental trade-off in Opera.

C. Simulation of Opera

The performance of the Opera network was compared to two other cost-comparable networks using several network workloads with different combinations of bulk and latency-sensitive traffic. The first network was a 3:1 oversubscribed folded-Clos (FC) network. The second network was a static expander graph network with $u = 7$, meaning that there are seven uplinks from

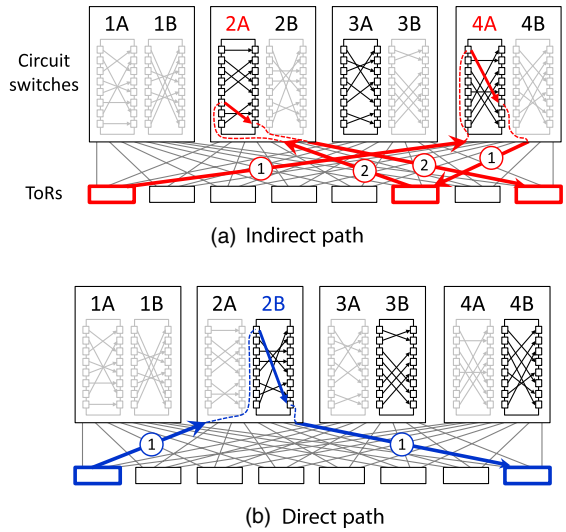

Fig. 4. Opera topology with eight ToR switches and four rotor circuit switches. Two different paths from rack 1 to rack 8 are highlighted: (a) two-hop path in red and (b) one-hop path in blue. Each direct inter-rack connection is implemented only once per configuration, while multi-hop paths are available between each rack pair at all times (from [12]).

every ToR into the network. In general, for these simulations, flows less than 15 MB were treated as low latency and routed over indirect paths, while flows greater than 15 MB were treated as bulk and routed over direct paths. Further details on the specifics of the simulation and the results are given in Section 5 of [12].

For the first workload, all flows were routed over direct paths. This workload establishes a baseline for the bandwidth efficiency of the Opera network compared to the other two networks when only bulk traffic is considered. The workload is an all-to-all shuffle operation using a flow size of 100 KB. This value was derived from the median inter-rack flow size reported in a Facebook Hadoop cluster [17] (c.f., Fig. 1). Figure 5 shows the network throughput over time for the three different networks.

The limited capacity of the 3:1 FC and the low bandwidth efficiency (high bandwidth tax) of the static expander network (exp) significantly extend the FCT of the shuffle operation, yielding 99th-percentile FCTs of 227 ms and 223 ms, respectively. Opera's direct bandwidth-efficient paths are bandwidth tax free, allowing higher throughput and reducing the 99th-percentile FCT to 60 ms.

The second workload has a mixture of bulk and low-latency traffic. For this simulation, we combine websearch traffic (low latency) and shuffle traffic (bulk) in varying proportions. Figure 6 shows the aggregate network throughput as a function of websearch (low-latency) traffic load. While Opera "gives up" a factor of two in low-latency capacity because of its relatively under provisioned ToRs, it gains a factor of two to four in bulk capacity from its bandwidth-efficient direct links. Further, Opera delivers comparable FCTs to the baseline networks across all flow sizes (see Section 5 of [12]).

In summary, Opera can accommodate a light overall low-latency load with little to no degradation in FCTs while

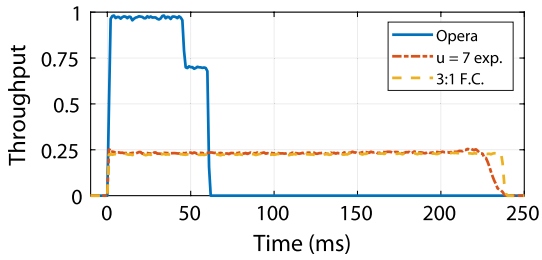

Fig. 5. Network throughput over time for a 100-KB all-to-all shuffle workload. Opera carries all traffic over direct paths, greatly increasing throughput (the small "step" down in Opera's throughput around 50 ms is due to some flows taking one additional cycle to finish) (from [12]).

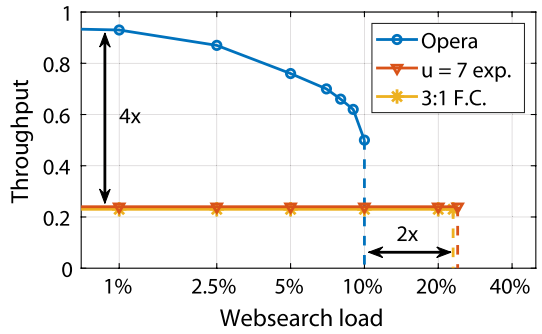

Fig. 6. Network throughput versus websearch traffic load for a combined websearch/shuffle workload (from [12]).

significantly improving throughput for bulk traffic, making it a good fit for many of today's datacenter workloads.

5. NETWORK SYNCHRONIZATION

Precise synchronous injection of packets at the edge of a circuit-switched network is essential to effectively utilize a high-speed optical network. Modern operating systems are inherently asynchronous, relying on interrupts, callbacks, and batched operations. As such, common operating systems cannot precisely synchronize an end host with a circuit switch at high data rates because they do not support real-time guarantees necessary to ensure packets are sent at the precise times required for a circuit-switched network. This means that high-precision injection of packets at realistic line rates requires some form of custom hardware.

The need for hardware-based synchronization was the motivation for the development of Corundum [13]. Corundum is an open-source FPGA-based NIC designed to provide a network interface similar in performance to a commercially available NIC, while enabling the implementation of additional hardware features needed for circuit switching. Corundum implements the standard IEEE 1588 precision time protocol (PTP). The PTP implementation in Corundum provides time synchronization to a precision of better than 100 ns, requiring around 10 sync packets per second. The net result is sub-microsecond precision time synchronization across multiple hosts with minimal overhead.

Combined with an efficient direct memory access (DMA) engine to transfer data from host memory, the unique open-source features of Corundum provide the capability to control packet transmissions with microsecond precision, enabling operation with microsecond-scale optical switches with no additional software overhead. The development of such a network interface enables the construction of practical sub-microsecond circuit-switched networks at scale.

Corundum also provides direct access to physical-layer components, enabling exhaustive *in situ* physical-layer BER link characterization. This unique measurement capability will be discussed further in Section 6.C.

A. Overview of Corundum

A high-level block diagram of the Corundum architecture is shown in Fig. 7. Corundum consists of 3 main nested modules. The top-level module primarily contains support and interfacing components. These components include a DMA interface, the IEEE 1588 PTP hardware clock that synchronizes network components, and Ethernet interface components including medium access controllers (MACs), physical layers (PHYs), and associated serializers. The top-level module also includes one or more interface module instances. Each interface module corresponds to an operating-system-level network interface (e.g., eth0). Each interface module contains the queue management logic, which maintains the queue state for all of the NIC queues. Each port module contains a transmit scheduler and transmit and receive engines.

At the host, there are multiple packet queues set up depending on the number of connected end hosts. Based on the rotor switch matching schedule, Corundum "pulls" data from the queues in main memory across the peripheral component interconnect express (PCIe) bus using DMA. RotorNet's deterministic performance and behavior greatly simplify this task, since complex non-deterministic polynomial-time hardness (NP-hard) scheduling problems are not required. Corundum then organizes these packets into a separate schedule for each uplink of the parallel network shown in Fig. 2(b).

Transfers into and out of host memory must ultimately be controlled by the operating system and device driver. To that end, hardware queues are used to enable communication between the device driver and the NIC. The driver is responsible for initializing the hardware, allocating DMA accessible memory for all of the queues, and passing packets between the networking stack and the NIC.

The queue management logic is specifically designed to support a large number of transmit queues so that traffic can

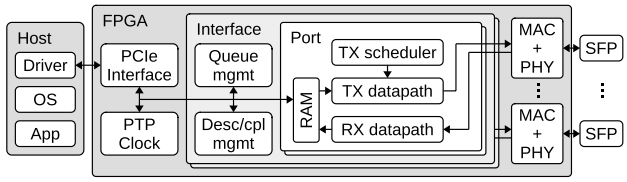

Fig. 7. High-level block diagram of a Corundum FPGA-based NIC (from [13]).

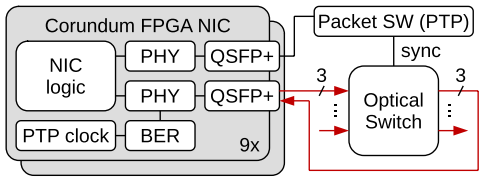

Fig. 8. Interface between the Corundum FPGA-based NIC and an optical switch. The packet switch instantiates PTP. This protocol is then used to synchronize each server containing a Corundum FPGA-based NIC with the rotor switch.

be controlled by the NIC on a per-destination basis. The current design of Corundum can support over 1000 queues and multiple ports per network interface.

B. Corundum in RotorNet

A simplified block diagram of how Corundum is interfaced with Rotornet is shown in Fig. 8. To support RotorNet, we implemented a reference design for TDMA with a fixed schedule on Corundum. Operating in a circuit-switched environment requires precise control of packet transmit timing, synchronized between the optical rotor switch and the end hosts. IEEE 1588 PTP uses hardware timestamping to enable time synchronization over the network with sub-microsecond precision. A PTP clock and PTP transmit and receive timestamping support have been implemented on Corundum, along with driver support. PTP is also used to control the speed of the rotor switch, which is discussed in Section 6.C.

TDMA is implemented by enabling and disabling queues in the transmit scheduler according to PTP time, under the control of the TDMA scheduler control module. Timing signals for the TDMA schedule are generated from PTP time. Using a maximum transmissoin unit (MTU) of 9 kB and eight instances of the network performance measurement tool *iperf3*, Corundum achieved a data rate of 94.0 Gb/s and could control the data leaving the NIC with a precision of two packet lengths or 1.4 μs. This precision is 7% of the nominal 20 μs switching speed of the rotor switch.

The TDMA scheduler was then configured to run a schedule with period 200 μs containing two timeslots of 100 μs, enabling all transmit queues in the first timeslot and disabling them in the second timeslot for an overall duty cycle of 50%. In this configuration, the throughput dropped to 48.5 Gb/s, which is nearly the expected value of $94.0/2 = 47.0$ Gb/s. Further details are provided in [13].

6. OPTICAL ROTOR SWITCH

RotorNet and Opera are based on rotor switches, which are optical switches that sequentially and periodically route a group of fiber inputs through a limited set of predetermined network connection patterns. The network connection patterns themselves are "hard-wired" into the switch using passive optics (e.g., fiber patch panels in our prototype). Sequential reconfiguration between these connection patterns is performed by a rotating disk that has been patterned with a set of diffraction gratings on its surface (resembling a pinwheel). A

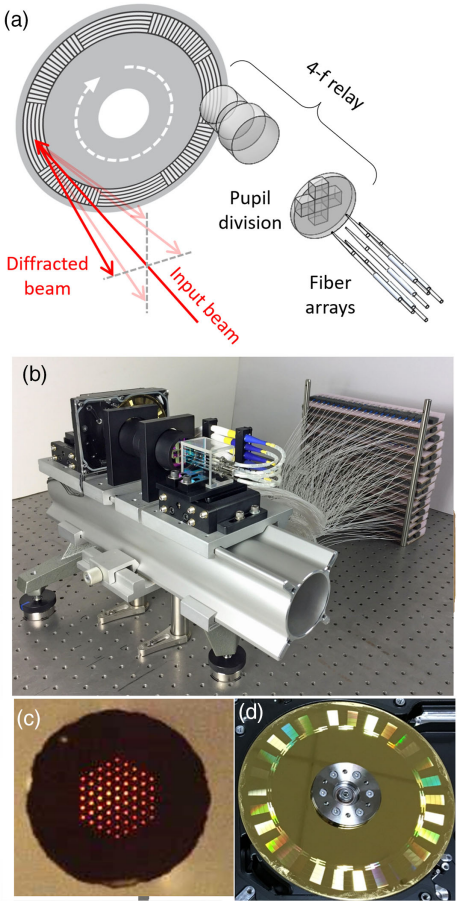

Fig. 9. (a) Pinwheel switch layout showing four beam deflection states and the optical relay between the fiber arrays and the disk. Photographs of (b) the benchtop prototype switch, (c) the fiber array, and (d) the grating pinwheel.

schematic of this pinwheel switch is shown in Fig. 9(a). The switch can make use of commercial magnetic hard-drive platters and spindles, which have both been engineered for high reliability. At a rotation rate of 7200 rpm (standard for hard-drive spindles), the switch can achieve a 15 μs reconfiguration time for 61 single-mode fiber signals, which is three orders of magnitude faster than a conventional MEMS beam-steering optical switch. This is also 10 times faster than our previous "selector switch" prototype [11], which provided random access (rather than sequential access) to the same number of connection patterns using a MEMS tilt mirror. The pinwheel achieves this speedup by decoupling the beam-steering angle, optical aperture, and actuator mechanics, which are inherently coupled in MEMS tilt mirror devices.

This switching platform is compatible with both single-mode and multimode signaling, and optimized designs provide a unique combination of high port count (>1000), low insertion loss (≈ 3 dB), and a fast switch reconfiguration time (<20 μs) [11]. Because of their simplified internal design, rotor switches can scale to thousands of ports and tens of connection patterns while remaining practical to build and deploy.

A. Operation of a Rotor Switch

Figure 9(a) shows the basic layout and operating principle of the pinwheel rotor switch. A 4-f relay is used to image a two-dimensional array of 61 single-mode input signals onto a disk patterned with blazed diffraction gratings near its perimeter. As the disk spins, each grating sector diffracts the beam into one of four directions. Were the grating grooves simply linear, the beam angle would wander as the disk rotated through each sector. To avoid this, we conformally map each grating region to an annular sector, enabling quasi-static beam steering with no beam wander within each sector as the disk rotates. The pupil of the 4-f relay optic is divided using an array of lenses so that after diffraction by the grating, the input signal array is imaged onto one of four output fiber arrays. For this prototype, a fiber patch panel is used to implement the connection pattern for each of the four fiber arrays. These four connection patterns define the four network topologies that the rotor switch cycles through.

B. Switch Assembly and Characterization

Figure 9(b) shows the assembled benchtop prototype rotor switch with the optical track mounted to a robust rail system. Figure 9(c) shows the end-face of a Chiral Photonics optical fiber array used in the switch [18]. Five fiber arrays are packaged into a rigid glass mounting fixture, which obviates the need for individual fiber positioning stages and eliminates positional drift of the arrays over time. Figure 9(d) shows the pinwheel switching element, which was fabricated by direct-write grayscale laser lithography. The pinwheel disk was mounted onto a commodity hard-drive spindle. We repeated the four unique grating sectors 14 times each (for a total of 56 sectors) so the reconfiguration time for the entire fiber array would be 10% of the dwell time in each switch state (for a 90% duty cycle).

At 5000 rpm, the measured reconfiguration time for an individual fiber channel was 500 ns, and the measured reconfiguration time for the entire 61-channel array was 25 μs [Fig. 10(a)]. Reconfiguration time is proportional to disk speed, enabling sub-10-μs reconfiguration with commodity 15,000 rpm spindles.

Figure 10(b) shows spectral transmission measurements for a representative switch port. To test the custom-fabricated 4-f relay optics, we first positioned the pinwheel so an unpatterned region (acting as a flat mirror) reflected the signal to couple back into the input fiber, double-passing the 4-f relay, which corresponds to a single pass through the switch. Using a fiber-optic circulator, we measured a maximum loss of 0.5 dB over a 120 nm spectrum. Next we aligned the pinwheel so the signal diffracted from the grating and coupled into one of the output fiber arrays. The loss was between 3.5 dB and 4.5 dB over the same 120 nm spectrum (including approximately 0.5 dB of polarization-dependent loss).

The primary source of loss in the current prototype switch comes from low diffraction efficiency in the prototype pinwheel. We are currently refabricating the pinwheel to improve its diffraction efficiency. Initial measurements indicate the potential to reduce loss by nearly 2 dB per pass using a refabricated pinwheel, yielding a projected total double-pass switch

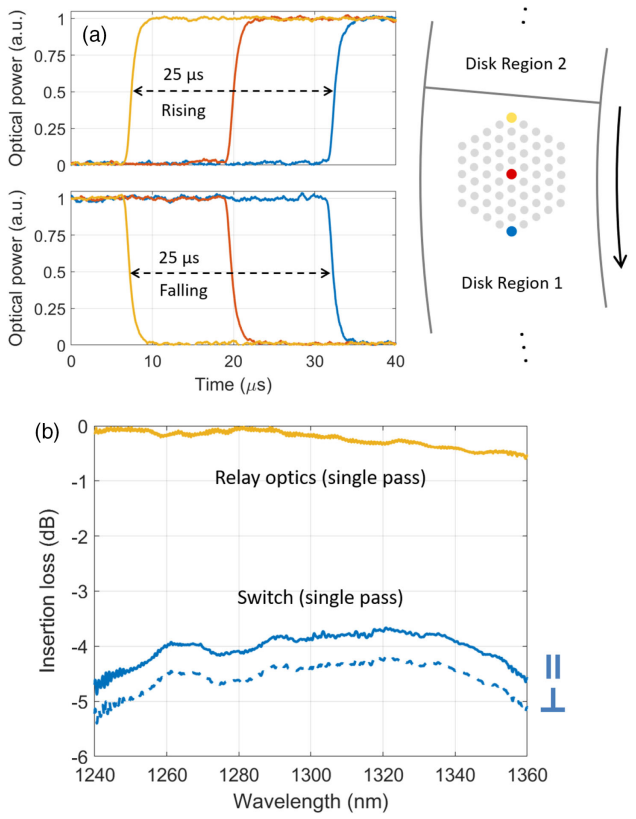

Fig. 10. (a) Full-array switch reconfiguration at 5000 rpm (2 μs detector response). Colors indicate the fiber connection measured within the fiber array. (b) Single-pass spectral transmission of the relay optics and switch.

loss of about 4 dB. While this is about 1 dB greater loss than that of commercial switches [8,10], it still permits the use of commercial transceiver technology; as shown in the following subsections, commodity PSM4 transceivers have a sufficient link margin to close a link through the current prototype rotor switch.

C. Data Transmission Through the Rotor Switch

Synchronized data transmission experiments were conducted in a network testbed. To enable these measurements, the breadboard version of the switch was "ruggedized" in a standard 6U enclosure for rack mounting. This ruggedized prototype is shown in the top of Fig. 11. The racked switch was then connected to nine servers in the testbed shown in the bottom of Fig. 11.

The Corundum FPGA-based NIC discussed in Section 5 controlled the synchronization between the end hosts in the testbed and the rotor switch using PTP, as shown in Fig. 12. Referring to this figure, a 40G Arista packet switch was used as a PTP boundary clock. To control the rotor switch, an additional end-host with a different FPGA-based NIC (ExaNIC) was used to generate an analog square-wave locked to the PTP signal from the Arista switch. To control the spindle motor on the pinwheel switch, we modified the firmware of an open-source motor driver (ODrive) to accept an external pulse-per-revolution analog signal from the ExaNIC card

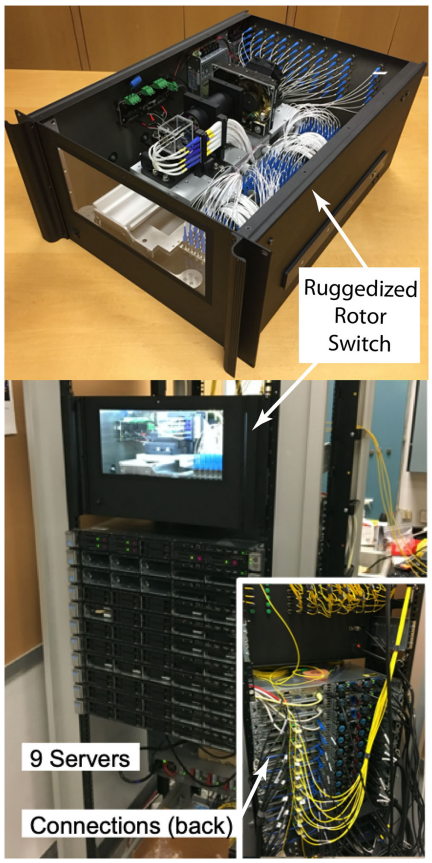

Fig. 11. Top: assembled rotor switch in a "ruggedized" rack mountable enclosure. Bottom: racked rotor switch in a small-scale system testbed.

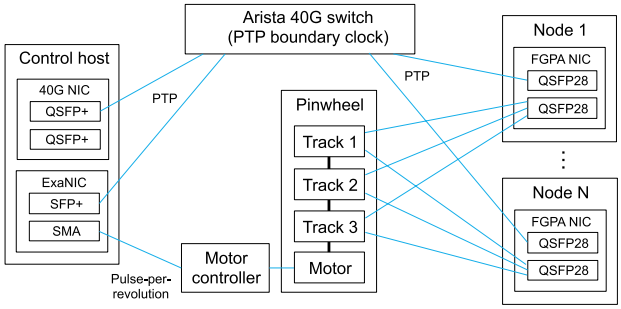

Fig. 12. Block diagram of the synchronization of the network.

and phase-locked the motor to that signal in an open-loop configuration. At a speed of 5000 rpm, the motor controller maintained the pinwheel phase to within 15 μs over time intervals exceeding 2 h.

D. BER Measurements

BER measurements were performed for a complete double-pass through the rotor switch using a gated BER detector that was implemented on Corundum. Figure 13 illustrates the experimental setup. The FPGA design contained multiple instances of a gated error detector, all connected to the same

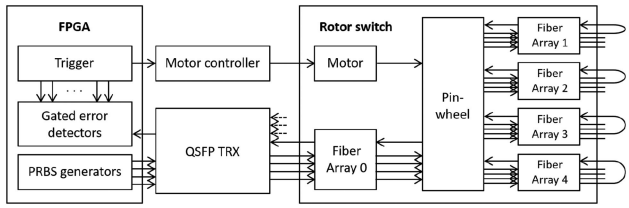

Fig. 13. Gated BER measurement used to test synchronous data transmission through the rotor switch.

receiver. A trigger generator was designed to gate each error detector instance as well as generate the synchronization signal for the motor controller. When the disc is phase-locked to the trigger signal, the gating signals can be adjusted to measure the error rate of each grating sector on the pinwheel in parallel.

For this set of experiments, a COTS 40G PSM4 QSFP+ transceiver was used. This transceiver had a measured error rate of 10^{-9} when 15.7 dB of link attenuation was added. A pseudo-random binary sequence (PRBS) was generated in the Corundum NIC. The number of instances of the gated BER measurement implemented on each NIC depends on the network configuration and number of uplinks per node. The experiment used a nine-server testbed with three 10 Gb/s uplinks per server connected to separate tracks of a single rotor switch. This switch was configured to support 54 periodic network configurations (three configurations repeated 18 times) for each uplink.

The gated BER measurement module on the Corundum NIC can concurrently acquire 32 time bins for each of the 54 configurations. The switch reconfigures every 222 μs, so capturing 128 bins per configuration in four offset measurements results in a resolution of 1.7 μs. Each NIC collected $3 \times 54 \times 32 = 5184$ concurrent measurements. Across the nine hosts, a total of $5184 \times 9 = 46{,}656$ concurrent measurements can be collected. This kind of automated diagnostic analysis is essential for identifying and correcting link-level problems in large-scale optically switched networks.

The BER measurements for one of three Rx channels on two hosts are shown in Fig. 14 in the form of a "heatmap" with the y axis denoting the network configuration. The x axis is the time offset for a transmission window of 222 μs. Each 222 μs slot was divided into 128 1.7 μs slots for a measurement. The heatmap shown in Fig. 14 shows color-coded BER versus time for one input connection through 54 sectors of one full disk rotation. The system-level switching time of approximately 40 μs is the width of the yellow band of high errors shown in Fig. 14. This system-level switching time includes the physical switch reconfiguration time (≈22 μs), the automatic gain control (AGC), and clock-data recovery lock time (≈10 μs) for a switched connection with a small power offset during the switch reconfiguration, and the disk synchronization (≈10 μs) associated with the motor controller. It does not include the Ethernet 64b/66b frame sync or the NIC transmit timing accuracy.

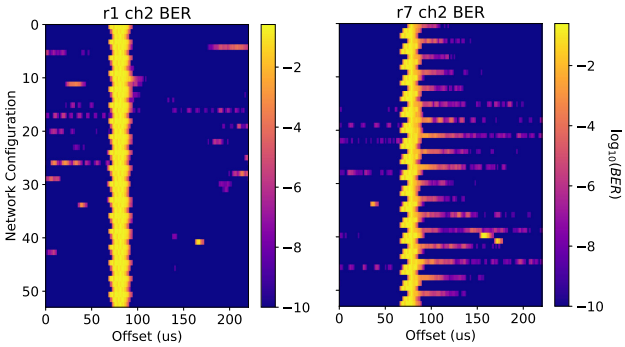

Fig. 14. Selection of time-resolved BER heatmaps from the network BER measurements, representing 2 out of 27 receivers.

E. Power Offsets Across a Switch Reconfiguration

As can be seen in the two heatmaps shown in Fig. 14, there is a large variation in the time-resolved BER for different network connections. This variation occurs because the lock time of the receiver is a function of the optical power offset across a switch reconfiguration. This power offset is a combination of differences in transmit power between transceivers and path-dependent loss through the optical network. For specific connections, such as the connections shown in the heatmap on the left of Fig. 14, the power offset is small for most connections. This leads to similar lock-time characteristics after a switch reconfiguration.

For specific connections, we used the synchronization capabilities of the Corundum NIC to run an unmodified server application called *iperf*, which is a standard networking characterization tool. Our ability to run *iperf* over the RotorNet testbed using COTS transceivers and a standard network stack that includes Transmission Control Protocol (TCP) demonstrates the viability of Opera/RotorNet in a datacenter environment when there is little or no power offset during a switch reconfiguration.

For other connections, such as some of those shown in the heatmap on the right side of Fig. 14, variable power offsets across a switch reconfiguration led to variable lock times [19]. In turn, the variable lock times led to extended time intervals over which there was a high BER, as indicated by the red-hued lines in the heatmap on the right of Fig. 14.

Figure 15 quantifies the effect of a power offset on the lock time across a switch reconfiguration between two channels denoted A and B. The switch used for this experiment had a 76 ns reconfiguration time, 0.4 dB insertion loss, and −28 dB crosstalk. The fast reconfiguration time was used to isolate the locking dynamics of the transceiver from the dynamics of the optical switch.

The top part of the figure is a BER heatmap with the y axis being the power offset ratio $10\log_{10}(P_A/P_B)$ between the two channels and the x axis being time. The mean channel attenuation for both channels was set to 10 dB to emulate a realistic optical network. The times to achieve BER $= 10^{-8}$ as a function of the power offset for both switch reconfigurations are shown below the heatmap.

The figure shows a smooth increase in the lock time as a function of the power offset. This behavior is consistent with

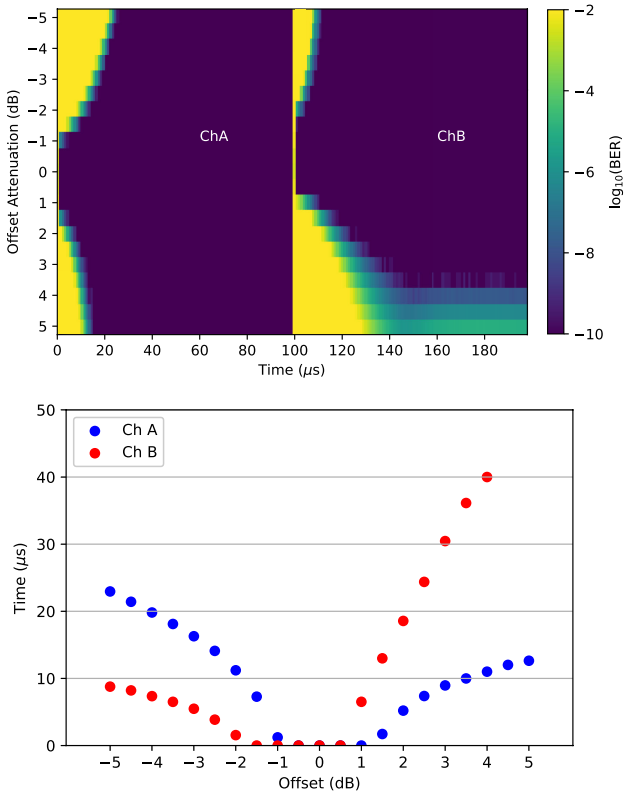

Fig. 15. Top: BER heatmap of a COTS 10 Gb/s transceiver when switched between two transmit channels A and B with a mean attenuation of 10 dB that are offset in power from the mean. Bottom: extracted lock times to achieve a BER of 10^{-8} for both switch reconfigurations.

an optical receiver circuit that uses AC coupling between gain stages [20] so that the threshold detection circuits must wait until the capacitor recharges before it can lock onto the new data stream. Other transceivers did not exhibit this behavior and cannot be explained by a simple time constant.

The longer reconfiguration time of a rotor switch can significantly change the lock-time characteristics compared to the fast switch results shown in Fig. 15 [19]. For example, for a 20 μs switch reconfiguration time, which is representative of a rotor switch, one COTS transceiver that was tested had a nearly constant lock time with a weak dependence on the power offset. For the same reconfiguration time, other transceivers showed a much stronger dependence of the lock time on the power offset.

The differences in the lock times are likely due to the part of the receiver that sets the threshold. Under the premise that the transceiver that had nearly a constant lock time uses AC coupling, the resistor–capacitor (RC) time constant is much shorter than the 20 μs reconfiguration time so that the capacitor completely discharges. In this case, the receiver "forgets" where the threshold was set and needs to re-lock after every switch reconfiguration. Other transceivers show a greater dependence on the power offset. This behavior is again not surprising because COTS transceivers are not designed for burst-mode optical switching.

These results also demonstrate the importance of matching the locking characteristics of the transceivers to the reconfiguration time of the optical switch. Without a burst-mode standard for datacom transceivers or detailed knowledge of the COTS transceivers, this matching must be done empirically on a case-by-case basis.

7. FUTURE WORK

Our future work focuses on three key areas. The first is identifying additional workloads for which the Opera/RotorNet architecture is a viable alternative to existing datacenter networks. Our initial results in [12], summarized in Section 4.C, indicate that Opera can provide an improvement in network throughput for map-reduced style workloads with large correlated shuffle operations (e.g., sorting), as well as "heavy-tailed" workloads in which most of the communicated bytes are contained in large "bulk" transfers (e.g., Microsoft's published datamining workload [21]). Opera also shows promise for large-block I/O for parallel filesystems. As part of our ongoing work, we are developing a framework to identify additional workloads that can benefit from our optical networking architectures, and are considering both hyperscale datacenter workloads and high-performance computing (HPC) workloads.

The spectrum of workloads can be loosely classified along two dimensions—the first is the distribution of communication payload sizes. Flow and message payloads can span anything from a few bytes to very large, bulk transfers that are measured in mega- or even gigabytes of data per transfer. The second dimension is the spatial and temporal structure of the communication pattern. At one end of this spectrum is random communication that is impossible to predict and is therefore exceptionally challenging to optimize hardware for, and at the other end are highly structured communication routines that can be easily and concisely described. Our goal is to determine regions in this application space where optical networking architectures can provide performance improvements over existing architectures.

Our second focus area is completing the implementation of the Opera architecture in an end-to-end system-level testbed. This implementation will leverage the unique functionality of the Corundum FPGA-based NIC, which will handle the low-level processing required to implement Opera in real time. The complete testbed will allow us to measure and validate Opera's networking performance and compare it with the simulations shown in Section 4.C. The calibrated simulations can then be used to determine a validated relationship between the optical switch reconfiguration time and the fraction of traffic that Opera can support over direct connections. The justification for the use of Opera becomes stronger as the switch reconfiguration time decreases and more traffic can be carried over direct connections. Determining a validated model of this relationship is a key goal of future work within LEED.

Finally, we are working with industrial partners to determine conditions under which COTS transceivers can be used with our optical networking architectures without the system impairments in our current testbed. Our preliminary results [19] indicate that commercial transceivers may be viable for prototyping, but are probably not viable for large-scale commercial deployments without new optical interconnect standards. A large effort within LEED, which was not discussed in this paper, is the development of cost-effective burst-mode interconnect technology that can be used with a rotor switch without the need for optical amplification.

In conclusion, the optical networking research conducted within LEED has shown both the promise of optical networking for computing applications as well as the additional work required for optical networking to become a practical alternative to conventional networking. Nevertheless, as the optical switching hardware and control plane improve and the appropriate network interface capabilities are developed, it is anticipated that optical switching will begin to replace standard packet switching for application-specific workloads.

Funding. National Science Foundation (CNS-1314921, CNS-1553490, CNS-1564185, CNS-1911104, CSR-1629973, SBIR-1842768); Advanced Research Projects Agency - Energy (DE-AR000084).

Acknowledgment. We thank Facebook for supporting this work through a gift. The work at inFocus Networks is supported by an NSF Phase I SBIR under award number 1842768.

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. A. Willner, *Optical Fiber Telecommunications VII* (Elsevier, 2019).
2. 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 *Conference of the ACM Special Interest Group on Data Communication* (ACM, New York, New York, USA, 2017), pp. 267–280.
3. P. Gupta and N. McKeown, "Designing and implementing a fast crossbar scheduler," IEEE Micro **19**, 20–28 (1999).
4. A. Rylyakov, J. E. Proesel, S. Rylov, B. G. Lee, J. F. Bulzacchelli, A. Ardey, B. Parker, M. Beakes, C. W. Baks, C. L. Schow, and M. Meghelli, "A 25 gb/s burst-mode receiver for low latency photonic switch networks," IEEE J. Solid-State Circuits **50**, 3120–3132 (2015).
5. A. Cevrero, I. Ozkaya, P. A. Francese, C. Menolfi, M. Braendli, T. Morf, D. Kuchta, M. Kossel, L. Kull, D. Luu, J. Proesel, Y. Leblebici, and T. Toifl, "A 60 Gb/s 1.9 pJ/bit NRZ optical-receiver with low latency digital CDR in 14 nm CMOS FinFET," in *Symposium on VLSI Circuits* (2017), pp. C320–C321.
6. K. Suzuki, S. Namiki, H. Kawashima, K. Ikeda, R. Konoike, N. Yokoyama, M. Seki, M. Ohtsuka, S. Saitoh, S. Suda, H. Matsuura, and K. Yamada, "Nonduplicate polarization-diversity 32×32 silicon photonics switch based on a SiN/Si double-layer platform," J. Lightwave Technol. **38**, 226–232 (2020).
7. S. Han, T. J. Seok, N. Quack, B.-W. Yoo, and M. C. Wu, "Large-scale silicon photonic switches with movable directional couplers," Optica **2**, 370–375 (2015).
8. Series 7000n datasheet, https://www.polatis.com/series-7000-384x384-port-software-controlled-optical-circuit-switch-sdn-enabled.asp.
9. J. Kim, C. J. Nuzman, B. Kumar, D. F. Lieuwen, J. S. Kraus, A. Weiss, C. P. Lichtenwalner, A. R. Papazian, R. E. Frahm, N. R. Basavanhally, D. A. Ramsey, V. A. Aksyuk, F. Pardo, M. E. Simon, V. Lifton, H. B. Chan, M. Haueis, A. Gasparyan, H. R. Shea, S. Arney, C. A. Bolle, P. R. Kolodner, R. Ryf, D. T. Neilson, and J. V. Gates,

"1100 × 1100 port MEMS-based optical crossconnect with 4-dB maximum loss," IEEE Photon. Technol. Lett. **15**, 1537–1539 (2003).
10. S series optical circuit switch, https://www.calient.net/products/s-series-photonic-switch.
11. W. M. Mellette, G. M. Schuster, G. Porter, G. Papen, and J. E. Ford, "A scalable, partially configurable optical switch for data center networks," J. Lightwave Technol. **35**, 136–144 (2017).
12. W. M. Mellette, R. Das, Y. Guo, R. McGuinness, A. C. Snoeren, and G. Porter, "Expanding across time to deliver bandwidth efficiency and low latency," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (USENIX Association, 2020), pp. 1–18.
13. A. Forencich, A. C. Snoeren, G. Porter, and G. Papen, "Corundum: an open-source 100-Gbps NIC," in *IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)* (2020), pp. 38–46.
14. W. M. Mellette and J. E. Ford, "Scaling limits of MEMS beam-steering switches for data center networks," J. Lightwave Technol. **33**, 3308–3318 (2015).
15. S. A. Jyothi, A. Singla, P. B. Godfrey, and A. Kolla, "Measuring and understanding throughput of network topologies," in *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2016), pp. 761–772.
16. N. Alon, "Eigenvalues and expanders," Combinatorica **6**, 83–96 (1986).
17. A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *ACM Conference on Special Interest Group on Data Communication* (ACM, New York, New York, USA, 2015), pp. 123–137.
18. Chiral Photonics, https://www.chiralphotonics.com/products/two-dimensional.
19. J. Kelley, A. Forencich, and G. Papen, "Burst-mode characteristics of datacom transceivers," in IEEE Optical Interconnects Conference 2020 (accepted for presentation).
20. X. Z. Qiu, X. Yin, J. Verbrugghe, B. Moeneclaey, A. Vyncke, C. V. Praet, G. Torfs, J. Bauwelinck, and J. Vandewege, "Fast synchronization 3R burst-mode receivers for passive optical networks," J. Lightwave Technol. **32**, 644–659 (2014).
21. 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," in *ACM SIGCOMM 2009 Conference on Data Communication* (ACM, New York, New York, USA, 2009), pp. 51–62.

William M. Mellette received his Ph.D. in photonics from the University of California, San Diego, in 2016, where he conducted research on fiber optic switches for data center networks. From 2016 to 2018, he was a postdoctoral researcher in UC San Diego's Computer Science Department, where he researched optically switched network architectures. Since 2019, he has been CEO of inFocus Networks, a startup company commercializing optical networking technologies.

Alex Forencich received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of California, San Diego, in 2012, 2015, and 2020, respectively. He is currently a postdoctoral researcher at the University of California, San Diego. During his Ph.D. studies he interned at the IBM T. J. Watson Research Center in Yorktown Heights, New York. His research interests include datacenter networking, optical switching, and reconfigurable hardware.

Jason Kelley attended the University of California, San Diego, and graduated with an M.S. in electrical engineering in June 2020. His focus is on RF circuits & systems, with additional background in antenna design and optics. His M.S. thesis research on using COTS optical transceivers in burst-mode applications resulted in first authorship of a paper submitted to Optical Interconnects 2020. He currently works full time at GA-EMS Orbital Technologies as an RF & EMI/EMC System Engineer.

Joseph Ford is a Professor of ECE at the University of California, San Diego, where he leads the Photonics Systems Integration Lab, which does design and integration of free-space optical components including compact imagers and fiber optic switches. Dr. Ford was a member of Bell Labs' Advanced Photonics Research Department from 1994 to 2000, where he demonstrated the first MEMS spectral equalizer and wavelength add/drop switches. Dr. Ford is an OSA and IEEE Fellow, author of over 200 journal articles and conference proceedings, and inventor on more than 50 United States patents on optical communications and imaging.

George Porter is an Associate Professor in the Systems and Networking Group in the Department of Computer Science and Engineering at the University of California, San Diego. His research interests span the fields of computer networks, data-intensive computing, and computer systems. He is the Co-Director of the Center for Networked Systems and a Co-Founder of inFocus Networks. He has received a Google Focused Research Award, a NetApp Faculty Fellowship, and the NSF CAREER award.

Alex C. Snoeren is a Professor in the Computer Science and Engineering Department at the University of California, San Diego, where he is a member of the Systems and Networking and Security research groups. His research interests include operating systems, distributed computing, mobile and wide-area networking, and many aspects of Internet security and privacy. He is a Fellow of the ACM (2018) and IEEE (2020) and recipient of an Alfred P. Sloan Fellowship (2009), a National Science Foundation CAREER Award (2004), and best-paper awards at the ACM SIGCOMM (2001, 2007, 2018) and USENIX OSDI (2008) conferences.

George C. Papen is a Professor of ECE at the University of California, San Diego, where he leads the Lightwave Energy Efficient Datacenter (LEED) project within the ARPA-e ENLITENED program. His research area is in optical systems for communications and sensing. Recent achievements include the publication of the book *Lightwave Communications* in 2019 (Cambridge) and the 2020 IEEE Photonics Society Engineering Achievement award.