Performance evaluation of hybrid optical switch architecture for data center networks

Muhammad Imran, Martin Collier, Pascal Landais, Kostas Katrinis

In response to the need for high bandwidth and power efficient data center interconnection networks, different interconnects have been proposed based on the optical technology used: micro-electromechanical system (MEMS), optical cross connects (OXCs), arrayed waveguide grating routers (AWGRs) and semiconductor optical amplifier (SOAs). MEMS switches are based on mature technology, have low insertion loss and cross-talk, and are data rate inde-pendent. They are also the most scalable and the cheapest class of optical switches. However, the reconfiguration time of these switches is of the order of tens of milliseconds while fast optical switches have switching time in the range of a few nanoseconds. Fast optical switches can be based on AWGRs in conjunction with tunable wavelength converters or tunable lasers or they are based on SOAs in broadcast-and-select architecture. In this paper, we propose an optical interconnect architecture for the large scale data centers. The proposed interconnect: Hybrid Optical Switch Architecture (HOSA) is a hybrid design that features slow and fast optical switches. The hybrid design leverages strengths of both types of optical switches. To reduce complexity, we employ a single stage core topology that can be easily scaled up (in capacity) and scaled out (in the number of racks) without requiring major re-cabling and network reconfiguration. We investigate the scalability of the HOSA and show that by using a single stage core topology, it can be scaled to a hundreds of thousands of servers. We also investigate a trade-off between cost and power consumption of our design by comparing it with other well-known interconnects by using analytical modelling. We demonstrate power efficiency as compared to other conventional interconnects on account of upfront CAPEX but the additional CAPEX incurred in deploying our solution instead of traditional architecture is mitigated to some extent by reduced OPEX, due to its greater energy efficiency. We evaluate the performance of the system using network-level simulation by considering diverse workload communication patterns and system design parameters. Our results show low latency and high throughput with different workload communication patterns.

1. Introduction

Internet traffic has shown an exponential increase in recent years due to the advent of cloud computing based applications. Cloud computing infrastructure is deployed in data centers (DCs). The traditional architecture of the data center network (DCN) is based on a hierarchical design as shown in Fig. 1. It features several layers of electrical switches. At the front end, the content and load balance switches are connected to the Internet through the gateway routers, while at the back end, they are linked to the core switches. The core switches are linked to the aggregate switches and the aggregate switches are connected to the Top of the Rack (ToR) switches. Each ToR switch is connected to the servers in the rack. All the switches feature an electronic switch fabric and the links between them can be either copper cables or optical fibers. In the case of optical fiber links, optical–electrical–optical (O–E–O) conversion is required at every port of the switch. When a request comes from the external network, it first comes to the load balance and content switches which route the request to the appropriate servers. To fulfil the request, the servers can coordinate with other servers within the same or different racks. For example, the application servers can coordinate with the database servers to process the request. After completing the request, the response is sent to the external network through the gateway routers.

1.1. Limitations of traditional DCNs

There are significant challenges to meeting growing performance requirements with current data center architectures. These are described below.

1.1.1. Power

The electrical switches at different layers of a DCN and the transceivers required for O–E–O conversion are significant sources of power consumption in traditional DCN designs. The power consumption of the current inter-connection network incurs 23% of the total IT power consumption in a DCN while it is predicted that the interconnection network will incur a much higher percentage of overall IT power consumption in future DCNs [2]. It is shown in Table 1 that the peak performance required of data centers will continue to rise tremendously but the affordable budget for the total permissible power dissipation by the data centers is increasing at a much slower rate i.e. it doubles every 4 years due to various thermal dissipation factors.

1.1.2. Scalability

Large cloud computing data centers owned by Amazon, Microsoft and Google have tens of thousands of servers. With the expected growth in data center traffic, the number of servers in data centers is destined to increase which poses a significant challenge to the data center interconnection network.

1.1.3. Traffic locality

The projection of traffic growth in data centers according to the Cisco cloud index [3] is shown in Fig. 2. Observe that during the period from 2013 to 2018, the majority of data center traffic will remain within the data center while only a small portion of the traffic will go to the external network. Some of the traffic will also be exchanged between data centers for distributed and replicated services between databases in different data centers. Due to this high traffic locality, high bandwidth and low latency interconnections are required.

1.1.4. Higher bit rates

The performance of communication systems at high data rates using electrical transmission lines is degraded by dielectric losses and losses incurred due to skin effect. Power dissipation increases as data rates increase in electrical transmission lines. For example, for 10 Gig E, power restrictions limit cable length to about 10 m. Longer cables are possible, but power can exceed 6 W/port which is not feasible in large-scale data centers [4]. On the other hand, power consumption in optical fiber is independent of the data rate.

1.1.5. Latency

Latency is introduced by queuing in buffers and by propagation delays incurred by packets during transmission from one node to another. Packets have to be buffered by switches during packet processing and this delay can be long when there is congestion in the network. Although the switching speed of electronic switches is of the order of micro or nano-seconds but overall end-to-end packet delay is significant and will need to be reduced in future data centers.

1.1.6. Performance

The performance of data centers has been increasing on the order of 10 times every 4 years with bandwidth increasing on the order of 20 times in the same interval as shown in Table 1. Power consumption can only be allowed to increase perhaps twofold and cost by a factor of 1.5. There is an upper limit of power consumption of 20 MW and cost of $500M for exascale computing to take place [6].

Optical interconnects address these challenges because they are power efficient and can provide huge bandwidths. The performance of the optical interconnects is directly related to the type of optical switches used. Traditional optical switches are based on micro-electro-mechanical system (MEMS) technology. The attractions of MEMS switches include (a) excellent power efficiency due to the use of passive switching, (b) high port density, (c) low insertion loss and crosstalk, (d) an absence of transceivers due to using all-optical switching, (e) lower cost, (f) support of bidirectional communication, and (g) data rate independence. They are also highly scalable and are commercially available e.g. 3D-MEMS [7]. However, they have high switching times, of the order of tens of milliseconds. Fast optical switches using technologies such as arrayed waveguide grating routers (AWGRs) and semiconductor optical amplifiers (SOAs) are now available. An AWGR is a passive device and works in combination with tunable lasers (TLs) or tunable wavelength converters (TWCs). The switching time of these switches is determined by the tuning speed of TLs or TWCs which is in the order of a few nanoseconds [8]. An SOA works as an ON/OFF switch that allows light to pass through it or not and also compensates for losses that occur during transmission of optical signals. SOA switches also have a switching time in the range of a few nanoseconds [9,10]. Although both types of switches are fast, they are expensive in comparison to MEMS switches of the same capacity.

In this paper, we extend our recent work [11]. We propose a novel optical interconnection scheme based on fast and slow optical switches. The proposed technique leverages strengths of both types of optical switches. The strengths of one type of optical switch compensate for the weaknesses of the other type. The main idea is to utilize resources so as to ensure minimum latency. Instead of using optical circuit switching (OCS) or optical packet switching (OPS), we use optical burst switching (OBS) [12]. The OCS paradigm has been used in the backbone optical core network for many years. The OBS was also proposed for the backbone optical core network but it has not replaced OCS due to its limitation of high burst loss in this application. We implement OBS with a two-way reservation protocol that ensures zero burst loss. The two-way reservation is not suitable for longhaul backbone optical networks due to the high round trip time (RTT) of the control packet but for our optical interconnect for the DCN, this RTT is not high for several reasons: (1) the propagation delay is negligible, (2) faster optical switches are used at the core, (3) a fast optical control plane is used, processing of the control packet is rapid and (5) a single hop topology is used. We have shown the feasibility of OBS using fast optical switches in the context of the DCN in our recent work [13].

We design a resource allocation algorithm for efficient utilization of the resources that results in high throughput and low latency. We evaluate the performance of the proposed system using network-level simulation by consider-ing different capacities of slow and fast optical switches and also investigate a trade-off between cost and power consumption of our design by comparing it with well-known interconnects. We also evaluate the scalability of our new architecture by considering different capacities of servers in a rack and different ratios of fast and slow optical switches.

The rest of the paper is organized as follows. In Section 2, an overview of existing optical interconnects is presented. In Section 3, we describe our proposed architecture and we evaluate its performance in Section 4. Section 5 contains our conclusions.

Related work

We categorize existing optical interconnects into three categories: (1) hybrid optical/electrical architectures; (2) interconnects based on fast optical switches; and (3) interconnects based on both fast and slow optical switches.

Hybrid optical/electrical architectures are based on optical MEMS switches and electrical switches [14–17]. Helios [14] and c-Through [15] use MEMS switches for optical circuit switching and electrical switches for traditional electrical packet switching. Long-lived traffic flows are routed through MEMS switches while bursty and short-lived traffic is routed through electrical packet switches. Energy efficiency is still the major concern of these solutions because they use power hungry transceivers and electrical switches in the core of the network. The OSA [16] and Hydra [17] designs employ a multi-hopping technique. ToR switches generating high traffic volumes are connected to each other by a single hop while short-lived traffic flows are routed via multi-hop paths. The multi-hopping technique increases the latency and energy consumption of ToR switches for traffic using multi-hops.

Optical interconnects based on fast optical switches have been presented [18,10,8,19–24]. The LIONS [8] exploits a switching fabric based on AWGR in combination with TLs or TWCs at the input ports and multiple receivers per output port. Cost and scalability are their major limitations. The scalability of the LIONS has been addressed in the H-LIONS but the cost of the interconnect is still a major concern due to expensive TWCs/TLs. An orthogonal frequency-division multiplexing (OFDM) based architecture [21] is also based on AWGR switches but it uses OFDM-based transmitters instead of TWCs/TLs and a single receiver using parallel signal detection (PSD) technology to detect multiple OFDM signals. Apart from scalability, this design is also less power efficient due to the use of power hungry OFDM-based transmitters. SOA-based Interconnects [10,18] use a broadcast-and-select configuration and utilize space, time and wavelength domains. The broadcast-and-select architecture is an expensive design. The data vortex is also based on SOA switches but it uses a ring architecture [23,24]. The Mordia architecture [22] is based on wavelength selective switches (WSSs). WSSs have switching time in the order of a few microseconds. The Mordia uses time division multiplexing to share the optical bandwidth among the hosts. The scalability is the major limitation in this design due to the support of a limited number of wavelengths in WSSs.

A hybrid optical switching scheme [25] based on fast and slow optical switches uses a three stage Clos architecture at the core node. The edge node classifies incoming traffic into four types. Circuit and long-burst traffic is routed through slow MEMS switches while short bursts and packets are routed through SOA-based fast optical switches. Packet and burst losses are the major limitations in this architecture which not only increase end-to-end latency due to retransmission but also decrease overall throughput. The LIGHTNESS project is based on fast optical and slow MEMS switches [26,27]. It uses OCS with MEMS switches and OPS with fast optical switches. Its principle of operation is to route high demanding traffic through MEMS switches using OCS and short-lived flows through optical fast switches using

OPS. LIGHTNESS employs a software defined control plane to configure the topology and optical switches.

3. Hybrid optical switch architecture – HOSA

We employ OBS in the proposed data center network architecture. We aggregate packet traffic to create burst of short duration. A control packet is created to request the allocation of resources needed to transmit the burst from the controller by using a two-way reservation process similar to that proposed for optical burst switching networks [12]. Although such two-way reservation is not feasible in a longhaul backbone network, in data centers it is suitable for the reasons presented earlier. The controller assigns resources and sends the control packet back to the originating node as an acknowledgement. The burst is then transmitted on the pre-established path configured by the controller.

The proposed hybrid optical switch architecture for DCNs named HOSA is shown in Fig. 3. We use a two layer topology comprising electrical ToR switches at the edge and an array of optical switches at the core. The optical switches include both slow and fast optical switches. Servers in a rack are connected to ToR switches using bidirectional fiber links. Each ToR switch has X optical transceivers, in which K transceivers are linked to the slow optical switches and X-K transceivers are interfaced to the fast optical switches, where 1<K <X. If we consider N the total number of ToR switches in the network, then ðN NÞ is the minimum configuration for both fast and slow optical switches so that at least one port from all ToR switches connects to every (NxN) optical switch.

HOSA features separate data and control planes. The control plane is realized by using a centralized controller. Routing, scheduling, switch configuration and traffic matrix calculation are the main tasks of the controller. It handles connection requests from all ToR switches, finds routes to the destination ToR switch through optical switches, assigns timeslots to the connection requests by selecting a suitable channel to the destination ToR switch, and configures optical switches with respect to the timeslots allocated. In order to realize these functions, the controller maintains a record of the global connectivity state of the optical switches. It also collects traffic statistics to perform traffic matrix calculation. Traffic matrix calculation is used to configure each slow optical switch and it ensures that elephant flows are routed through slow optical switches. The data plane is realized by using optical switches, performing data forwarding on pre-established lightpaths configured by the controller. Each ToR switch has a dedicated optical transceiver which is connected to the controller through a management network.

3.1. ToR switch design

The ToR switch design is shown in Fig. 4. The ToR switch has an electronic switch fabric which is connected to the servers in the rack to perform intra-rack (within rack) switching in the electrical domain. To perform inter-rack (between racks) switching, we employ ðTRK 1Þ virtual out-put queues (VOQs) where TRK is the number of ToR switches in the network. State of the art ToR switches support hundreds of VOQs. For example, the Cisco Nexus 5500 supports up to 384 VOQs, the Cisco 5548P supports up to 18,432 VOQs and the Cisco 5596 supports up to 37,728 VOQs [28,29]. There is a VOQ for each destination ToR switch in the DCN. Packets destined to the same ToR are aggregated into the same VOQ. The VOQ not only aggregates traffic to the same destination ToR switch but it also avoids head of line blocking (HOL). Each VOQ is configured for a destination network address. Each ToR switch maintains a VOQ table where entries comprise the destination rack network address and the VOQ number. The dispatcher module matches the destination network address of the packet with the entry in this table and forwards the packet on the required VOQ.

3.1.1. Dynamic allocation of VOQs

For a very large scale DCN, the number of VOQs provided by the ToR switch can be less than the total number of racks in the DCN e.g. in the case of the Cisco 5500 switch, it supports only 384 VOQs. In this case we can use a subset of the total racks with which a given ToR communicates over a specified period of time. For example, in a thousand rack network, each rack may communicate with only a few other racks over a given period of time. So each rack would not require 999 VOQs. In this case we can dynamically allocate VOQs for the destination rack which the ToR is sending traffic.

For dynamic allocation, we need another field, i.e. time-stamp, in the VOQ table. In the first stage, the VOQ table contains only a list of VOQ entries in it without any corresponding destination network addresses. When a packet arrives at the dispatcher module, it looks up the destination network address in the list of VOQs but it does not find any match. Then it takes the first empty entry from the list of VOQs and assigns the destination network address and updates this field with the current timestamp and forwards the packet to this VOQ. When another packet arrives requesting the same destination rack, the dispatcher finds a match for this network address in the table, it updates its entry with the new timestamp and forwards the packet to the same VOQ. There is also a daemon process in the ToR switch that checks the VOQ list after a particular time interval. If the VOQ entry in the list has not been updated for that particular time, the destination network address entry from the list is deleted on the assumption that there is no more traffic for the destination rack. In this way, this VOQ can be assigned to another destination rack, after a timeout.

3.2. Control packet format

The format of the control packet is shown in Fig. 5. The control packet is 440 bits long and contains two main fields, routing and reservation. The routing field contains source and destination IP addresses and IDs of the ToR switches. These are the IP addresses of network interface cards (NICs) reserved for the control plane in the ToR switches. We consider 128 bits for IPv6 addresses, however this length could be reduced to 32 bits using IPv4 addresses making overall control packet length to 31 bytes.

The reservation field is 96 bits long, and is divided into 3 sub-fields: (1) burst length, (2) start time and (3) port number. The burst length field is filled by the ToR switch to request a timeslot from the controller. The controller fills rest of the two fields after processing the control packet. All of these three fields are 4 bytes long. The burst length field contains burst length expressed in bytes while start time contains time when the burst will be sent and the port number is the port of the ToR switch in which the burst is to be sent. The CRC field is reserved for cyclic redundancy check and a couple of optional fields are reserved for flags.

3.2.1. Burst assembly/disassembly

Burst assembly can be timer based, length based or a combination of both [12]. We consider the mixed approach in which either a timer expires or the burst length exceeds a threshold. The timer starts when a packet arrives at the empty VOQ. If the VOQ is not empty when the packet arrives, it joins other packets in the VOQ. The control packet is generated after the timer expires or the burst length exceeds the threshold and is sent to the controller using transceiver dedicated for the control plane. The control packet at this stage contains information of the burst length, IP addresses of source and destination ToR switches and IDs of source and destination ToR switches. Each ToR switch is assigned a unique ID. The range of IDs of ToR switches is from 0 to N 1 in N rack network. These IDs are used by the controller to perform routing and scheduling algorithm. The controller processes the control packet, assigns start time and port number of the ToR switch on which a burst is to be transmitted and sends it back to the source ToR switch. The control packet processing mechanism is described in Section 3.3. When the control packet arrives at the ToR switch, the scheduler module of the ToR switch generates a burst according to the timeslot assigned by the controller. The timeslot refers to the duration of time assigned for a burst in an optical switch path. The generated burst is then sent to the queue of the allocated port. The scheduler module also initiates a new timer if the VOQ is not empty after the burst generation because new packets might have been arrived during the RTT of the control packet. In order to realize bidirectional communication, the controller also generates a new control packet and sends it to the destination ToR switch. The destination TOR switch also generates a burst according to the timeslot allocated and sends it to the queue of the allocated transmitter.

The ToR switch also has burst disassembler and packet extractor module to disassemble the bursts received through the receivers. The receivers perform O–E conversion and send bursts to the disassembler module where packets are extracted from them and are sent to the electronic switch fabric and finally to the destination servers using electronic switching.

3.3. Control plane processing

The controller performs routing, scheduling and switch configuration functions. It also performs traffic matrix calculation to configure slow optical switch so that elephant flows are routed through the slow path.

Routing and scheduling operation is performed when a control packet arrives at the controller for a new timeslot and is described in Algorithm 1. There are different steps in the algorithm. First, the controller gets the source and the destination IDs of the ToR switches from the control packet. The next step is to check whether the same ToR pair has been assigned a timeslot recently or not to avoid duplicate timeslot allocation. For example, ToR 1 and 2 send control packets to the controller at the same time or with very little time difference. The controller receives the first control packet and schedules it and sends it to both ToR 1 and 2. Meanwhile it also receives the second control packet. To avoid duplicate time slot allocation, it deletes the control packet if ToR pair has been assigned timeslot recently. For this purpose, we define three parameters Tpre,Tcur and Tdup. Tpre is the previous reservation time for the ToR pair, Tcur is the current time and Tdup is the time to check for duplicate allocation. The control packet is deleted if condition in line 5 of Algorithm 1 is satisfied.

Next step is to find the latest horizon for both fast and slow paths. The term horizon refers to the latest available time when the channel will be free. The controller maintains a routing table which contains pre-defined routes of all source and destination ToR pairs and selects best route according to the latest horizon as described in lines 8–11 in Algorithm 1. Tfast and Tslow represent horizons of optimal fast and slow paths respectively. Tsch represents the horizon of an already established slow path between source and destination ToR pair and TRL is the length of timeslot to be allocated on the basis of burst length ðBLÞ in the control packet.

In order to configure the slow switch, the controller maintains a matrix table as shown in Table 2. It consists of three fields i.e. traffic (T), connections exist (CE) and connections allowed (CA) for every source destination pair. The controller updates traffic entry in the table for both source and destination ToR switches as described in lines and 24 of Algorithm 1. We define Interval time Tinterval after which the controller updates entry of the CA in the matrix table. The CA is maximum number of connections for slow path that a given source destination ToR pair can have in Tinterval and is calculated by the following formula:

$$CA = \lceil \frac{T \times 1024 \times 8}{datarate \times T_{interval}} \rceil \qquad (1)$$

Tidle represents the idle time for which the channel has not been used. The algorithm sets up a new slow path if Tcur-Tslow >=Tidle and CE <CA as shown in lines 25–29 in Algorithm 1. We assume that the channel in the slow switch can be assigned to the new request if it is idle since Tidle and traffic matrix also predicts this. All new paths of slow switch are assigned on the basis of this principle. Although the

slow path is established in this way, the current control packet request is assigned to a fast path if there is no slow path already established or (Tfast+TRL)<Tsch). Otherwise, already established slow path is assigned to the current request by updating its horizon in the controller. After assigning time-slots, start time and port number fields in the control packet are filled by the controller (lines 38 and 41). The controller also generates a new control packet by duplicating existing control packet and updates port number, source and destination addresses and their relevant IDs in the control packet because source and destination ToR will have different port numbers (lines 39–44 and 53–57). In the end, the original control packet is sent back to the source ToR switch and the newly generated control packet is sent to the destination ToR switch for bidirectional communication (lines 46, 47, 60, and 61).

The ports of the slow switches are not necessarily be connected. Fortunately, connectivity is easy to achieve via the port exchange operation as described in [16]. First,we find unconnected ports and we select two unconnected ports a-b and two connected ports c-d and connect them via replacing links a->b and c->d with a->c and b->d. There is a daemon process in the controller which runs after periodic intervals to check and connect the unconnected ports in the slow switches.

Switch configuration is the final operation of the controller. After processing the control packet, a configuration message is generated and is sent to the switch controller to configure the optical switch. The configuration message contains the source and destination port numbers and the connection start-time. It does not contain the connection end-time because connections are established with unlim-ited duration. The connection end-time information is only maintained by the controller. The switch controller config-ures the optical switch according to the instructions in the configuration message.

The biggest advantage of establishing a connection with an unlimited duration is that when a new connection request arrives at the controller for an already established connection in the slow path, the controller only updates its horizon with a new time and nothing is done in the switch controller. But in order to ensure fairness, this principle does not apply on the fast switch path. In the fast switch path, all the traffic has an equal probability of getting a timeslot while in slow switch paths, the probability of being assigned a timeslot on an already established connection is higher than of a timeslot on an as yet unestablished connection. This is to avoid frequent reconfiguration of MEMS switches so that persistent traffic flows are routed through the slow switch paths.

4. Performance analysis

In this section, we evaluate the HOSA design by ana-lyzing its scalability, cost and power consumption. We also investigate its latency and throughput performance by using network-level simulation.

4.1. Scalability

Slow optical MEMS switches with 320 bidirectional ports are commercially available while fast optical switches can be built using technologies as described in our recent work [30]. Due to the constraint of maximum port size of fast or slow optical switches, the scalability of the architecture with only one fast and one slow switch is limited to a few thousand servers as shown in the first two rows of Table 3. This is suitable for departmental and medium-scale enterprise data centers, but we target large-scale high performance computing data centers in the order of O(10 K) of servers. In order to achieve this, we employ multiple optical fast and slow switches in the core arranged in a single stage topology as shown in Fig. 3.

Table 3 describes different configurations of slow and fast switches (SS and FS respectively), ratio of their capacities, number of slow and fast switches required (NSS and NFS respectively), servers/rack (SRK), and various core/edge oversubscription ratios (CO). It can be seen that maximum system size with only one fast and one slow switch, each having ½320 320 configurations and oversubscription ratios of 4 and 2 is limited to 2560 and 1280 servers respectively. The maximum size of the system reaches to 12,800 servers having 40 servers/rack and ½320 320 switches configuration with different capacities of slow and fast switches. It can be observed that the number of slow and fast optical switches required are varied with the capacity of slow:fast switches. Slow MEMS switches with 1024 ports are feasible [31,7] and fast optical switch using SOAs with 1024 ports has also been proposed [10]. The system size of 40,960 servers with 40 servers/rack and 81,920 servers with 80 servers/rack can be achieved with the proposed single stage topology without converting to multi-stage core topologies. Eighty servers/rack can be integrated by using two 64 port ToR switches per rack. Similarly, if we consider a pod switch instead of the ToR switch that has the capacity to integrate several ToR switches into a single unit and can aggregate a few hundreds to thousand servers [14], leads to the scalability up to 245,760 servers by considering 240 servers per pod which is ideal for future large scale data centers.

It can be observed that the size of the optical switch (port density) controls the maximum number of racks while the number of optical switches controls the core oversubscription ratio. Single-stage core interconnect topology with multiple optical switches allows our design to both incrementally scaled up (in capacity) and scaled out (in the number of racks) without requiring major re-cabling and network re-configuration similar to the topology used in reconfigurable architecture [32].

We avoid multi-stage core topology due to the complexity of the control plane and optical signal degradation at every intermediate optical switch (providing all optical switching) due to insertion losses and crosstalk. Optical amplifiers may be required in multi-stage core topologies that will not only increase overall cost of the interconnect but also the power consumption. Although the multi-stage designs can be scaled to a very large topology but scaling is expensive and is not incremental.

4.2. Cost and power consumption

In our analysis of cost and power consumption, we consider only cost and power consumption of the network elements that are used in the interconnection network. Table 4 highlights the cost and power consumption of different net-work elements that we use in our analysis. We consider four interconnection networks to perform a comparative analysis. These networks are Fat tree, BCube, Traditional-electrical (TE) and Optical–electrical (OE). We assume different capacities of fast and slow switches in the HOSA to perform a fair comparison with these four networks. Our design is scalable to 40,960 and 81,920 servers using ToR switches at the edge as described in Section 4.1, so we consider these two values for servers to compare our design with other networks.

4.2.1. Fat tree network

Fat tree (FT) is the most common tree topology that is used in DCNs [33]. The FT with n-port switches can connect $n^3/4$ hosts with a total number of $5n^3/4$ switch ports. The power consumption of the FT network PFT is calculated by

$$P_{FT} = \frac{5n^3}{4}(P_{CMOS}+P_{TR})+\frac{n^3}{4}P_{TR} \qquad (2)$$

where PCMOS is the power consumption of an electrical switch port and PTR is the power consumption of a transceiver. For example, in order to calculate power consumption of 40,960 servers, we consider 54 port switches, 54^3/4=39,366 servers and total number of 5x54^3/4 switch ports. So the power consumption of the FT network with 39,366 servers in this case is given by:

$$P_{FT} = \frac{5 \times 54^3}{4}(12.5+1) + \frac{54^3}{4} \times 1 = 2.696571 \text{ MW} \qquad (3)$$

The FT network cannot have exactly 40,960 servers, so we normalize this power consumption value to 40,960 servers that results in 2.80576 MW as shown in Fig. 6(b). Similar phenomenon is used for the power consumption of 81,920 servers. The CAPEX cost of the FT network CCAPEXFT is calculated by using the following formula:

$$C_{FT}^{CAPEX} = \frac{5n^3}{4}(C_{CMOS} + C_{TR}) + \frac{n^3}{4}C_{TR} \qquad (4)$$

where CCMOS is the cost of the electrical switch port and CTR is the cost of the transceiver. Similarly, the CAPEX cost of the FT network having 39,366 servers is given by:

$$\begin{aligned} C_{FT}^{CAPEX} &= \frac{5 \times 54^3}{4}(500+400) + \frac{54^3}{4} \\ &\times 400 = 192.8934 \text{ M US\$} \qquad (5) \end{aligned}$$

For 40,960 servers, the normalized cost results in 200.704 Million US $ as shown in Fig. 6(a). The OPEX cost is related with the power consumption. In order to cal-ulate OPEX cost, we consider 0.1 cent per unit cost of electricity that is used in the United States. The OPEX cost of the FT network COPEXFT is calculated by using the following formula:

$$C_{FT}^{OPEX} = P_{FT} \times 1000 \times 24 \times 365 \times 0.1 \times Years \qquad (6)$$

$$C_{FT}^{OPEX} = P_{FT} \times 1000 \times 24 \times 365 \times 0.1 \times Years \tag{6}$$

4.2.2. BCube network

The BCube network is a non-tree based architecture which is proposed for modular data centers (MDCs) [34]. The BCube network takes a server-centric approach, rather than a switch-oriented approach. Servers have multiple ports that are connected with the different levels of electrical switches. Servers not only send their traffic to other servers but they also work as switches to forward traffic on behalf of other servers. The $BCube_0$ has n servers which are connected to an n-port switch. The $BCube_k$ ($k \geq 1$) is constructed from n $BCube_{k-1}s$ and n^k n-port switches. There are $k+1$ level of switches and each level has n^k n-port switches. There are a total of $N = n^{k+1}$ servers and each server has $k+1$ ports. The power consumption of the $BCube_k$ network P_{BCube} is calculated by using the following formula:

$$P_{BCube} = n^k(k+1).n(P_{CMOS}+P_{TR})+n^{k+1}(k+1)(P_{TR}) \tag{7}$$

We consider a $BCube_3$ network with $k=3, n=14$ having $N = 14^{3+1} = 38,416$ servers and where each server has $k+1=4$ ports. So the power consumption of this network is given by:

$$P_{BCube_3} = 14^3(3+1) \times 14(12.5+1) + 14^{3+1}(3+1)(1) = 2.228128 \text{ MW} \tag{8}$$

As with the Fat tree network, the BCube network cannot have exactly 40,960 servers, so we normalize this power consumption value to 40960 servers that results in 2.37568 MW as shown in Fig. 6(b). The CAPEX cost of the $BCube_k$ network C_{BCube}^{CAPEX} is given by:

$$C_{BCube}^{CAPEX} = n^k(k+1).n(C_{CMOS}+C_{TR})+n^{k+1}(k+1)(C_{TR}) \tag{9}$$

Similarly, the CAPEX cost of the BCube network with 38,416 servers is given by:

$$C_{BCube}^{CAPEX} = 14^3(3+1) \times 14(500+400)+14^{3+1}(3+1) \times 400 = 199.7632\text{M US \$} \tag{10}$$

For 40,960 servers, the normalized cost results in 212.992 Million US $ as shown in Fig. 6(a). The OPEX cost of the BCube network C_{BCube}^{OPEX} is calculated by using the following formula:

$$C_{BCube}^{OPEX} = P_{BCube} \times 1000 \times 24 \times 365 \times 0.1 \times Years \tag{11}$$

4.2.3. Traditional electrical network

For the traditional electrical (TE) network, we consider a 2:1 oversubscribed network having edge/pod and core switches. We consider the edge/pod switch to be a cluster of the ToR switches making the aggregation layer [14]. The power consumption of the TE network P_{TE} is calculated by using the following formula:

$$P_{TE} = P_{EDGE}^{TE} + P_{CORE}^{TE} + T_{SR} \cdot P_{TR} \tag{12}$$

where P_{EDGE}^{TE} and P_{CORE}^{TE} represent total power consumption at the edge and the core switches respectively. T_{SR} is the total number of servers in the network. P_{EDGE}^{TE} is calculated using the following formula:

$$P_{EDGE}^{TE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_A\right)(P_{CMOS}+P_{TR}) \tag{13}$$

where T_{RK} represents total racks in the network and S_{RK} denotes the number of servers in the rack. N_A is the number of ports per ToR switch connecting other ToR switches to make the edge/pod switch and $\frac{S_{RK}}{2}$ represents the number of ports of ToR switches which are linked to the core switches. The P_{CORE}^{TE} is calculated using the following formula:

$$P_{CORE}^{TE} = \left(\frac{T_{RK}}{2} \cdot S_{RK}\right)(P_{CMOS}+P_{TR}) \tag{14}$$

For example, we get the power consumption P_{TE} for 40,960 servers by substituting different values as shown below:

$$\begin{aligned} P^{TE} &= 1024\left(40+\frac{40}{2}+16\right)(12.5+1) \\ &\quad + \left(\frac{1024}{2} \times 40\right)(12.5+1)+40,960 \times 1 \\ &= 1.368064 \text{ MW} \end{aligned} \tag{15}$$

Similarly, we calculate the CAPEX cost of the TE network C_{TE}^{CAPEX} by using formula:

$$C_{TE}^{CAPEX} = C_{EDGE}^{TE} + C_{CORE}^{TE} + T_{SR}.C_{TR} \tag{16}$$

where C_{EDGE}^{TE} and C_{CORE}^{TE} represent the total cost of the edge and the core switches respectively. C_{EDGE}^{TE} and C_{CORE}^{TE} are calculated by using the following formulae:

$$C_{EDGE}^{TE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_A\right)(C_{CMOS}+C_{TR}) \tag{17}$$

$$C_{CORE}^{TE} = \left(\frac{T_{RK}}{2} \cdot S_{RK}\right)(C_{CMOS}+C_{TR}) \tag{18}$$

For example, we get the CAPEX cost C_{TE}^{CAPEX} for 40,960 servers by substituting different values as shown below:

$$\begin{aligned} C_{TE}^{CAPEX} &= 1024\left(40+\frac{40}{2}+16\right)(500+400) \\ &\quad + \left(\frac{1024}{2} \times 40\right)(500+400)+40,960 \times 400 \\ &= 104.8576\text{M US\$} \end{aligned} \tag{19}$$

The OPEX cost of the TE network C_{TE}^{OPEX} is calculated by using the following formula:

$$C_{TE}^{OPEX} = P_{TE} \times 1000 \times 24 \times 365 \times 0.1 \times Years \tag{20}$$

4.2.4. Optical/electrical network

We consider an abstract model for the optical/electrical network similar to Helios [14]. It consists of two layer of switches: edge/pod and core. The edge switches are cluster of ToR switches while the core switches are combination of electrical and MEMS switches. The core layer is 2:1 oversubscribed in which half of the links are connected to the electrical switches and other half to the MEMS switch. The power consumption of the OE network P_{OE} is calculated by using the following formula:

$$P_{OE} = P_{EDGE}^{OE} + P_{CORE}^{OE} + T_{SR}.P_{TR} + P_{CP}^{OE} \tag{21}$$

where P_{EDGE}^{OE} and P_{CORE}^{OE} represent the total power consumption at the edge and the core switches respectively while P^{OE}_{CP} is the total power consumption of the control plane. P^{OE}_{EDGE} and P^{OE}_{CORE} are calculated by using the following formulae:

$$P^{OE}_{EDGE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_A+N_{CP}\right)(P_{CMOS}+P_{TR}) \tag{22}$$

$$P^{OE}_{CORE} = \left(\frac{T_{RK}}{4}.S_{RK}\right)(P_{CMOS}+P_{TR})+\left(\frac{T_{RK}}{4}\cdot S_{RK}\right)(P_M) \tag{23}$$

where N_{CP} is the number of transceiver in each ToR switch dedicated for the control plane and $\frac{T_{RK}}{4}$ are the number of links each in the electrical and the MEMS switches which are connected to the edge switches. The P^{OE}_{CP} is calculated using the following formula:

$$P^{OE}_{CP} = T_{RK}(P_{CMOS}+P_{TR})+N_{TR_{Cont}}\cdot P_{TR}+N_{SW}\cdot P_{SWC} \tag{24}$$

where $N_{TR_{Cont}}$ is the number of transceivers in the controller server and N_{SW} represents the number of the switch controller for the MEMS switch and P_{SWC} is the power consumption of the switch controller. For example, we get the power consumption P_{OE} for 40,960 servers by substituting different values from Table 4 as shown below:

$$\begin{aligned} P_{OE} = {} & 1024\left(40+\frac{40}{2}+16+1\right)(12.5+1) \\ & +\left(\frac{1024}{4}\times 40\right)(12.5+1)+\left(\frac{1024}{4}\times 40\right)(0.24) \\ & +40,960\times 1+1024(12.5+1)+1\times 1+10\times 300 \\ = {} & 1.2491066\ \text{MW} \end{aligned} \tag{25}$$

Similar phenomenon is used to calculate CAPEX cost of the OE network. The CAPEX cost of the OE network C^{CAPEX}_{OE} is given by:

$$C^{CAPEX}_{OE} = C^{OE}_{EDGE}+C^{OE}_{CORE}+T_{SR}\cdot C_{TR}+C^{OE}_{CP} \tag{26}$$

where C^{OE}_{EDGE} and C^{OE}_{CORE} represent the total cost of the edge and the core switches, respectively and C^{OE}_{CP} is the total cost of the control plane. The C^{OE}_{EDGE}, C^{OE}_{CORE} and C^{OE}_{CP} are calculated by using the following formulae:

$$C^{OE}_{EDGE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_A+N_{CP}\right)(C_{CMOS}+C_{TR}) \tag{27}$$

$$C^{OE}_{CORE} = \left(\frac{T_{RK}}{4}\cdot S_{RK}\right)(C_{CMOS}+C_{TR})+\left(\frac{T_{RK}}{4}\cdot S_{RK}\right)(C_M) \tag{28}$$

$$C^{OE}_{CP} = T_{RK}(C_{CMOS}+C_{TR})+N_{TR_{Cont}}.C_{TR}+N_{SW}\cdot C_{SWC} \tag{29}$$

where C_M is the cost of the MEMS switch port and C_{SWC} represents the cost of the switch controller. For example, we get the CAPEX cost C_{OE} for 40,960 servers by substituting different values from Table 4 as shown below:

$$\begin{aligned} C^{CAPEX}_{OE} = {} & 1024\left(40+\frac{40}{2}+16+1\right)(500+400) \\ & +\left(\frac{1024}{4}\times 40\right)(500+400)+\left(\frac{1024}{4}\times 40\right)(500) \\ & +40,960(400)+1024(500+400)+1\times 400 \\ & +10\times 1000 = 107.5016\ \text{M US\$} \end{aligned} \tag{30}$$

The OPEX cost of the OE network C^{OPEX}_{OE} is calculated by using the following formula:

$$C^{OPEX}_{OE} = P_{OE}\times 1000\times 24\times 365\times 0.1\times \mathit{Years} \tag{31}$$

4.2.5. HOSA

Our proposed design HOSA also consists of two layers of switches: ToR switches at the edge and an array of optical switches at the core. Similar to the TE and OE networks, the core layer is also 2:1 oversubscribed from the ToR switches. Studies have shown that the Fat tree network is under-utilized i.e. 40% of the fat tree resources are utilized [35] and fully subscribed network is inefficient. So in order to efficiently utilize resources, network oversubscription is used.

The power consumption of the HOSA P_{HOSA} is calculated by using the following formula:

$$P_{HOSA} = P^{HOSA}_{EDGE}+P^{HOSA}_{CORE}+T_{SR}\cdot P_{TR}+P^{HOSA}_{CP} \tag{32}$$

where P^{HOSA}_{EDGE} and P^{HOSA}_{CORE} represent the total power consumption at the edge and the core switches while P^{HOSA}_{CP} is the total power consumption of the control plane. The P^{HOSA}_{EDGE} and P^{HOSA}_{CORE} are calculated by using the following formulae:

$$P^{HOSA}_{EDGE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_{CP}\right)(P_{CMOS}+P_{TR}) \tag{33}$$

$$P^{HOSA}_{CORE} = T_{RK}\cdot NS_{RK}\cdot P_M+T_{RK}\cdot NF_{RK}\cdot P_F \tag{34}$$

where NS_{RK} and NF_{RK} represent the number of transceivers per ToR switch connected to the slow and fast switches respectively while P_M and P_F are the power consumption per port of the MEMS and the fast optical switches respectively. The P^{HOSA}_{CP} is calculated using the following formula:

$$\begin{aligned} P^{HOSA}_{CP} = {} & T_{RK}(P_{CMOS}+P_{TR})+N_{TR_{Cont}}\cdot P_{TR}+NS_{SW_{Cont}}\cdot P_{SWC} \\ & +NF_{SW_{Cont}}\cdot P_{SWC} \end{aligned} \tag{35}$$

where $NS_{SW_{Cont}}$ and $NF_{SW_{Cont}}$ are the number of the switch controllers for the MEMS and the fast optical switches respectively while P_{SWC} is the power consumption of the switch controller. For example, we get the power consumption P_{HOSA} for 40,960 servers with FS=0.5 and SS=0.5 by substituting different values from Table 4 as shown below:

$$\begin{aligned} P_{HOSA} = {} & 1024\left(40+\frac{40}{2}+1\right)(12.5+1)+1024\times 10 \\ & \times 0.24+1024\times 10\times 3+40,960 \\ & \times 11,024(12.5+1)+1\times 40+10 \\ & \times 300+10\times 300 = 0.9065456\ \text{MW} \end{aligned} \tag{36}$$

We calculate the CAPEX cost of the HOSA C^{CAPEX}_{HOSA} using the same method as we use for the calculation of power consumption. The C^{CAPEX}_{HOSA} is calculated by using the following formula:

$$C^{CAPEX}_{HOSA} = C^{HOSA}_{EDGE}+C^{HOSA}_{CORE}+T_{SR}\cdot C_{TR}+C^{HOSA}_{CP} \tag{37}$$

where C^{HOSA}_{EDGE} and C^{HOSA}_{CORE} are the total cost of the edge and the core switches respectively while C^{HOSA}_{CP} represents the cost of the control plane. The C^{HOSA}_{EDGE} and C^{HOSA}_{CORE} are calculated by using the following formulae:

$$C^{HOSA}_{EDGE} = T_{RK}\left(S_{RK}+\frac{S_{RK}}{2}+N_{CP}\right)(C_{CMOS}+C_{TR}) \tag{38}$$

$$C^{HOSA}_{CORE} = T_{RK}\cdot NS_{RK}\cdot C_M+T_{RK}\cdot NF_{RK}\cdot C_F \tag{39}$$

where C_M and C_F represent the cost per port of the MEMS and the fast optical switch respectively. The C_{CP}^{HOSA} is calculated by using the following formula:

$$C_{CP}^{HOSA} = T_{RK}(C_{CMOS} + C_{TR}) + N_{TR_{Cont}} \cdot C_{TR} + NS_{SW_{Cont}} \cdot CS_{SW_{Cont}} + NF_{SW_{Cont}} \cdot CF_{SW_{Cont}} \tag{40}$$

where $CS_{SW_{Cont}}$ and $CF_{SW_{Cont}}$ are the cost of the switch controller for slow and fast switches respectively. For example, we get the CAPEX cost C_{HOSA} for 40960 servers with FS=0.5 and SS=0.5 configuration as shown below:

$$\begin{aligned} C_{HOSA}^{CAPEX} = {} & 1024\left(40 + \frac{40}{2} + 1\right)(500 + 400) + 1024 \times 10 \\ & \times 500 + 1024 \times 10 \times 15,000 + 40,960 \\ & \times 11,024(500 + 400) + 1 \times 400 + 10 \times 1000 \\ & + 10 \times 1000 = 232.2636 \text{ M US\$} \end{aligned} \tag{41}$$

The OPEX cost of the HOSA C_{HOSA}^{OPEX} is calculated by using the following formula:

$$C_{HOSA}^{OPEX} = P_{HOSA} \times 1000 \times 24 \times 365 \times 0.1 \times Years \tag{42}$$

4.2.6. Results

We calculate CAPEX and OPEX costs, and power consumption of the FT, BCube, TE, OE and HOSA networks and results are shown in Figs. 6 and 7.

We calculate cost and power consumption of the HOSA by using different capacities of the fast and slow optical switches. In Figs. 6(a) and (b) and 7, FS represents the capacity of the fast optical switches and SS represents the capacity of the slow optical switches. FS,SS= 0,1 means that there is no fast optical switch and all of the switching capacity is provided by the slow optical switches only while FS,SS= 1,0 reveals that there is no slow optical switch and all of the switching capacity is provided by the fast optical switches only. These are the two extreme cases, which we consider as the worst and the best case respectively. Similarly, FS,SS=0.2,0.8 means that 20% of the switching capacity is provided by the fast optical switches and the remaining 80% capacity is provided by the slow optical switches. We compare different combi-nation of the switching capacities in HOSA with two extreme cases as well as with other networks.

Fig. 6 (a) shows that the CAPEX cost of the interconnection network using only fast optical switches is double as that of the FT and BCube network while it is quadruple as that of TE or OE networks, but this cost is reduced to almost half by considering HOSA with only 40% of the switches being fast. The cost of HOSA with this combination is almost the same as that of the FT and BCube network but it is still double as that of TE/OE networks. This extra upfront cost is mitigated to some extent by its reduced OPEX cost as shown in Fig. 7. The reduced OPEX cost is observed due to the improvement in power consumption as shown in Fig. 6(b). It can be inferred from Fig. 6(b) that with different capacities of the HOSA, a 70–65% improvement in power consumption is achieved over the FT and BCube networks while a 27–33% improvement in power consumption is achieved over the OE and TE networks respectively.

4.3. Modelling approach

To assess the latency and throughput performance of the HOSA, we developed simulations models using OMNeT++simulation framework [36]. Our simulation models consist of models for ToR switches, fast/slow optical switches and controller. We use OMNeTþ þ inet models for servers and electrical switches. The simulated topology consists of TR=40 total racks. Each rack has SRK = 40 servers and 1 ToR switch. Servers are connected to the ToR switch using bidirectional fiber links. Each ToR switch is also linked with the electrical switch using bidirectional fiber link via a transceiver reserved for use by control plane. The electrical switch in the control plane is connected to all the ToR switches, the controller and all optical switches. We con-sider two optical switches, one for slow path and other for fast path. We consider 2:1 core over-subscription by using X¼ 20 optical transceivers per ToR switch connected to the optical switches. To evaluate performance at different switching capacities for slow and fast optical switches, we use K ={0; 10; 12; 14; 16} links for the slow optical switch and X -K ={20; 10; 8; 6; 4g} links for the fast optical switch.

4.3.1. Traffic generation

To the best of our knowledge, there is no theoretical model or benchmark of the data center traffic has been established yet but there are few studies [37,38,31] that have investigated the nature of the data center traffic. The traffic characteristics of data centers is bursty in nature and shows evidence of ON–OFF behavior [38]. We use a Markov Chain Process model for bursty traffic with an ON period of 800 μs and an OFF period of 200 μs which are exponentially distributed. We consider various exponential inter-arrival rates of packets during the ON period to investigate traffic at different loads. We define two para-meters to control traffic generation. These are:

Stability: It is the lifetime (in milliseconds) of a traffic flow between two ToR switches.

Topological degree of communication (TDC): It is the number of simultaneous destinations ToR switches that a given source ToR switch sends traffic to.

The TDC parameter represents diversity of traffic work-loads. We select different values of TDC parameters to evaluate performance at low, medium and high traffic diversity. The stability is used to evaluate the ability of the core interconnect to adapt to constantly changing communication patterns.

4.3.2. Simulation parameters

The key simulation parameters are shown in Table 5. We choose a value of 1 μs for the processing time of the control packet by the controller. We use a value of 10 ms for the switching time of the slow MEMS switch [16]. We select a value of 1 μs for the switching time of the fast optical switches because this is a conservative choice, although in some types of fast optical switch this value can be as low as few nanoseconds [8]. The RTT of the control packet includes its processing time at the controller (Tproc) and the overhead time (Toverhead). The overhead time comprises propagation delay (5 ns for 1 m optical fiber), the processing delay of the control packet at the electrical switch, and the optical–electrical–optical (O–E–O) conversion delay. The aggregate value of Toverhead is conservatively set to 1 μs although all these delays are negligible (at most a few nanoseconds [16]).

We consider a value of 2 ms for interval time (Tint) for matrix calculation. For burst generation, we choose a combination of 100 μs for aggregation time (Ta) and 500 KB for burst length (BL). We choose three cases for TDC by using values drawn from the set f1; 10; 20g and a value of 500 ms for stability in order to evaluate their impact on performance of the system. Simulation time was set to 2 s.

4.3.3. Baseline electrical network

We benchmark the performance against an ideal traditional electrical (TE) packet switching network that features a two layer leaf-spine topology [1] as shown in Fig. 8. Its latency and throughput performance provide a baseline against which to compare the performance of the new networks. The TE network acts as an ideal electrical packet switching network that has low latency and high throughput as compared to the FT, BCube and OE networks due to higher number of hops in these networks.

4.3.4. End-to-end delay

We define end-to-end delay as the time between a packet is generated by the source server and the time in which packet is received by the destination server. The traffic within the same rack has negligible end-to-end delays because the ToR switches have the capacity to switch packets within nanoseconds range. We only investigate inter-rack traffic so that the performance of optical interconnect could be evaluated. The end-to-end delay is the sum of packet delay at the ToR switch and the propagation delay from the source to the destination servers. The packet delay at the ToR switch is the sum of packet queuing delay at NIC, packet processing delay, packet delay for burst assembly, packet delay till burst departure and delay due to O–E–O conversion. There is no queuing or processing delay at the optical switch due to all optical switching.

The simulation results obtained for latency are shown in Fig. 9. Fig. 9 shows the delay performance at different values of offered load by considering three values for TDC. Five of the curves at each plot in Fig. 9 represent end-to-end delay versus offered load using different capacities of fast and slow optical switches, while the sixth curve shows the corresponding performance of the baseline electrical network. The electrical network acts as the performance benchmark while the curve with FS,SS=1,0 is the best case in which all of the switching capacity is provided by the fast switches only. The other four curves represent hybrid switching capacities for mixed fast and slow switches. It can be seen in Fig. 9(a) that end-to-end delay increases by increasing traffic load for different hybrid capacities of fast and slow switches. This is due to the high switching time of slow optical switches but is still below 1 ms for various switching capacities as compared to the 10 ms switching time of the slow optical switches. A similar trend is also observed with high diversity traffic as shown in Fig. 9 (b)and (c). It can be noticed that the hybrid system where only 40% of the switching capacity is provided by the fast optical switches shows a performance comparable to the system only using fast optical switches until the load reaches 40%. The improvement of our design in scalability, cost and power consumption comes at the cost of latency. This is due to the traffic aggregation delay that is the inherent limitation of optical burst switching and also due to the higher switching time of MEMS switches but it is still comparable to the baseline electrical network.

4.3.5. Throughput

Fig. 10 shows the throughput performance observed at 80% offered load at the core by considering three values for TDC. There are four sets of four bars at each plot which represent average bandwidth achieved at fast switches, slow switches, average bandwidth of the interconnect using both fast and slow switches and average bandwidth in baseline electrical network. Each set of bars represent combination of different switching capacities as shown in the x-axis of all three plots.

It can be seen in Fig. 10(a) that the average bandwidth of slow optical switches is higher than the average band-width of fast optical switches and a similar trend is observed with high diversity traffic as shown in Fig. 10 (b) and (c). This is because the majority of the traffic is routed through slow optical switches and it also results in decreasing overall power consumption of the inter-connection network because slow MEMS switches are more power efficient than fast optical switches due to the use of passive switching.

It can also be observed that the overall interconnect bandwidth using both type of switches remains close to 8 Gbps at 80% load with TDC=1 and TDC=10 as shown in Fig. 10(a) and (b). It decreases slightly with high diversity traffic with TDC= 20 but is still comparable to the baseline electrical network as shown in Fig. 10(c). This is because with high diversity traffic, there are a plenty of requests for new connections and each request is delayed by the RTT of the control packet. The bandwidth of the interconnection is wasted during this RTT which results in decreased overall network through-put in the presence of high diversity traffic. A similar trend of performance of decreasing bandwidth with increasing traffic diversity is also observed in other optical interconnects [16,32].

4.3.6. Performance of the control plane

In order to assess the performance of the routing and scheduling algorithm of the control plane, we ran our algorithm on an Intel host with a Core i7, 2.17 GHz processor and 16 GB RAM. The results were obtained for several combinations of parameters. For statistical significance, we averaged the results of 1000 runs and the results are shown in Table 6. Table 6 shows the execution time of the algorithms for different network sizes N in terms of racks, different values of topological degree of communication (TDC), and different values of degree of ToR switches.

When a control packet arrives at the controller, the controller performs routing and scheduling operations as described in Algorithm 1. The complexity of the routing and scheduling algorithm is $O(2(2K + L) + \mu)$, where K is the number of ports of the ToR switch dedicated for the slow switch paths, L is the number of ports of the ToR switch assigned for the fast switch paths and μ represents the aggregate processing time of all other instructions. This is assumed to be a constant of negligibly low value. We measure the algorithm execution time in a 4:1 oversubscribed network when (K; L) = 5, in a 2:1 oversubscribed network when (K; L) = 10 and in a fully subscribed network when (K; L) = 20 using 40 servers per rack as shown in first three rows of Table 6. Fourth row of Table 6 shows its execution in a fully subscribed networking using 80 servers per rack. It can be inferred that the processing time of the control packet is independent of the network size and the TDC values. The execution time of the routing and scheduling algorithm is very low in 4:1 and 2:1 oversubscribed networks while it increases slightly because of the increased number of ports of ToR switches in a fully subscribed network.

The traffic matrix scheduling is used to measure the traffic statistics that are ultimately used by the routing and scheduling algorithm to configure each slow optical switch. The complexity of this algorithm is O(Nx(N-1) + μ). The performance of this algorithm depends upon the network size and the TDC parameter. It is independent of the network over-subscription as shown in Table 6. This algorithm runs periodically to predict the new traffic matrix. It can be seen that the execution time is proportional to network size and the TDC. In a very large network with worst case scenario, e.g. with N = 1024 and TDC =1023, we get an execution time around 30 ms that is infeasibly high. In a real network scenario, the TDC would not be too high because different studies on data center traffic [37,38,31] have shown that traffic within data centers is bounded in degrees and racks communicate with only a few other racks over a given period of time. A hardware implementation of the algorithms would reduce this time. Implementing our algorithm in hardware such as in an FPGA would reduce this time to a few microseconds even in an extreme worst case. We will explore the viability of a hardware implementation in future work.

5. Conclusion

We propose a novel optical interconnect based on a combination of slow and fast optical switches in a single stage core topology. The hybrid design exploits the strengths of fast and slow optical switches. We use OBS with two-way reservation to get zero burst loss. The two-way reservation is not appropriate for traditional backbone optical networks due to the high RTT of the control packet but in a DCN, this RTT is not high. We design a resource allocation algorithm in the controller that ensures minimum latency by allocating resources efficiently. We use network-level simulation by considering different combinations of slow and fast optical switches to validate our design.

We perform a scalability analysis of the proposed interconnect by investigating various ratios of slow and fast optical switches. The single stage core topology can be easily scaled up (in capacity) and scaled out (in the num-ber of racks) without requiring major re-cabling and net-work reconfiguration. We also investigate a trade-off between cost and power consumption of our design by comparing it with conventional interconnects by using analytical modelling. The additional upfront cost incurred in deploying our solution instead of conventional archi-tecture is mitigated to some extent by its reduced opera-tional cost, due to its greater energy efficiency.

Acknowledgment

The work was supported by the Irish Research Council (IRC) through the Enterprise Partnership Scheme and IBM Ireland. We are also grateful to the National Centre for Physics, Pakistan, for providing access to the high speed servers for running simulations.

References

[1] C. Kachris, I. Tomkos, A survey on optical interconnects for data centers, IEEE Commun. Surv. Tutor. 14 (4) (2012) 1021–1036.

[2] A. Benner, Optical interconnect opportunities in supercomputers and high end computing, in: Optical Fiber Communication Con-ference, Optical Society of America, Los Angeles, 2012, pp. OTu2B–4.

[3] Cisco Global Cloud Index: Forecast and Methodology, 2012–2017, URL ⟨http://www.cisco.com/c/en/us/solutions/collateral/service-pro vider/global-cloud-index-gci/Cloud_Index_White_Paper.html⟩.

[4] A.V. Krishnamoorthy, The intimate integration of photonics and electronics, in: Advances in Information Optics and Photonics, vol. 1, 2008, p. 581.

[5] M. Taubenblatt, J. Kash, Y. Taira, Optical interconnects for high per-formance computing, in: Communications and Photonics Con-ference and Exhibition (ACP), 2009, pp. 1–2.

[6] J. Shalf, S. Dosanjh, J. Morrison, Exascale computing technology challenges, in: High Performance Computing for Computational Science–VECPAR 2010, Springer, Berlin Heidelberg, 2011, pp. 1–25.

[7] Photonic Optical Circuit Switching j CALIENT Technologies, URL ⟨http://www.calient.net/⟩.

[8] Y. Yin, R. Proietti, X. Ye, C.J. Nitta, V. Akella, S. Yoo, LIONS: an awgr-based low-latency optical switch for high-performance computing and data centers, IEEE J. Select. Top. Quant. Electron. 19 (2) (2013) 3600409.

[9] S. Aleksic, Analysis of power consumption in future high-capacity network nodes, IEEE/OSA J. Opt. Commun. Netw. 1 (3) (2009) 245–258.

[10] O. Liboiron-Ladouceur, I. Cerutti, P.G. Raponi, N. Andriolli, P. Castoldi, Energy-efficient design of a scalable optical multiplane inter-connection architecture, IEEE J. Select. Top. Quant. Electron. 17 (2) (2011) 377–383.

[11] M. Imran, M. Collier, P. Landais, K. Katrinis, HOSA: hybrid optical switch architecture for data center networks, in: Proceedings of the 12th ACM International Conference on Computing Frontiers, CF '15, ACM, New York, NY, USA, 2015, pp. 27:1–27:8. http://dx.doi.org/10. 1145/2742854.2742877.

[12] Y. Chen, C. Qiao, X. Yu, Optical burst switching: a new area in optical networking research, IEEE Netw. 18 (3) (2004) 16–23.

[13] M. Imran, P. Landais, M. Collier, K. Katrinis, Performance analysis of optical burst switching with fast optical switches for data center networks, in: 2015 17th International Conference on Transparent Optical Networks (ICTON), 2015, pp. 1–4, http://dx.doi.org/10.1109/ ICTON.2015.7193596.

[14] N. Farrington, G. Porter, S. Radhakrishnan, H.H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, A. Vahdat, Helios: a hybrid electrical/optical switch architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 41 (4) (2011) 339–350.

[15] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, T.S. Ng, M. Kozuch, M. Ryan, c-Through: part-time optics in data centers, in: ACM SIGCOMM Computer Communication Review, vol. 40, 2010, pp. 327–338.

[16] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, Y.Chen, Osa: an optical switching architecture for data center networks with unprecedented flexibility, IEEE/ACM Trans. Netw. 22 (2) (2014) 498–511, http://dx.doi.org/10.1109/TNET.2013.2253120.

[17] K. Christodoulopoulos, D. Lugones, K. Katrinis, M. Ruffini, D. O'Mahony, Performance evaluation of a hybrid optical/electrical interconnect, IEEE/OSA J. Opt. Commun. Netw. 7 (3) (2015) 193–204, http://dx.doi.org/10.1364/JOCN.7.000193.

[18] O. Liboiron-Ladouceur, P.G. Raponi, N. Andriolli, I. Cerutti, M.S. Hai, P. Castoldi, A scalable space time multi-plane optical interconnection network using energy-efficient enabling technologies [invited], IEEE/OSA J. Opt. Commun. Netw. 3 (8) (2011) A1–A11.

[19] R. Proietti, Z. Cao, C. Nitta, Y. Li, S. Yoo, A scalable, low-latency, high-throughput, optical interconnect architecture based on arrayed waveguide grating routers, J. Lightw. Technol. 33 (4) (2015) 911–920, http://dx.doi.org/10.1109/JLT.2015.2395352.

[20] G. Wu, H. Gu, K. Wang, X. Yu, Y. Guo, A scalable awg-based data center network for cloud computing, Opt. Switch. Netw. 16 (2015) 46–51.

[21] P.N. Ji, D. Qian, K. Kanonakis, C. Kachris, I. Tomkos, Design and evaluation of a flexible-bandwidth OFDM-based intra-data center interconnect, IEEE J. Select. Top. Quant. Electron. 19 (2) (2013) 3700310.

[22] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun,
T. Rosing, Y. Fainman, G. Papen, A. Vahdat, Integrating Microsecond Circuit Switching into the Data Center, vol. 43, ACM, New York, 2013.

[23] O. Liboiron-Ladouceur, A. Shacham, B.A. Small, B.G. Lee, H. Wang, C.P. Lai,
A. Biberman, K. Bergman, The data vortex optical packet switched interconnection network, J. Lightw. Technol. 26 (13) (2008) 1777–1789.

[24] Q. Yang, Latency-optimized high performance data vortex optical switching network, Opt. Switch. Netw. 18 (2015) 1–10.

[25] M. Fiorani, S. Aleksic, M. Casoni, Hybrid optical switching for data center networks, J. Electr. Comput. Eng. (2014).

[26] J. Perelló, S. Spadaro, S. Ricciardi, D. Careglio, S. Peng, R. Nejabati,
G. Zervas, D. Simeonidou, A. Predieri, M. Biancani, et al., All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput, IEEE Netw. 27 (6) (2013) 14–22.

[27] S. Peng, D. Simeonidou, G. Zervas, R. Nejabati, Y. Yan, Y. Shu, S. Spadaro, J. Perello, F. Agraz, D. Careglio, et al., A novel sdn enabled hybrid optical packet/circuit switched data centre network: the lightness approach, in: 2014 European Conference on Networks and Communications (EuCNC), IEEE, Bologna, 2014, pp. 1–5.

[28] Cisco Nexus 5596, URL ⟨http://www.cisco.com/c/en/us/products/col lateral/switches/nexus-5548p-switch/white_paper_c11-622479.html⟩.

[29] Cisco Nexus 5548p, 5548up, 5596up, and 5596t Switches Data Sheet, URL ⟨http://www.cisco.com/c/en/us/products/collateral/switches/nexus-5000-series-switches/data_sheet_c78-618603.html⟩.

[30] M. Imran, M. Collier, P. Landais, K. Katrinis, Software-controlled next generation optical circuit switching for HPC and cloud computing datacenters, Electronics 4 (4) (2015) 909, http://dx.doi.org/10.3390/ electronics4040909. http://www.mdpi.com/2079-9292/4/4/909.

[31] K.J. Barker, A. Benner, R. Hoare, A. Hoisie, A.K. Jones, D.K. Kerbyson,
D. Li, R. Melhem, R. Rajamony, E. Schenfeld, et al., On the feasibility of optical circuit switching for high performance computing sys-tems, in: Proceedings of the 2005 ACM/IEEE Conference on Super-computing, IEEE Computer Society, Washington, 2005, p. 16.

[32] D. Lugones, K. Katrinis, G. Theodoropoulos, M. Collier, A reconfigurable, regular-topology cluster/datacenter network using commodity optical switches, Future Gener. Comput. Syst. 30 (2014) 78–89.

[33] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 63–74.

[34] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, S. Lu, Bcube: a high performance, server-centric network architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 63–74.

[35] S. Kamil, L. Oliker, A. Pinar, J. Shalf, Communication requirements and interconnect optimization for high-end scientific applications, IEEE Trans. Parallel Distrib. Syst. 21 (2) (2010) 188–202.

[36] OMNeTþ þ Simulation Framework, URL ⟨http://omnetpp.org/⟩.

[37] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, R. Chaiken, The nat-ure of data center traffic: measurements& analysis, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, 2009, pp. 202–208.

[38] T. Benson, A. Akella, D.A. Maltz, Network traffic characteristics of data centers in the wild, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, 2010, pp. 267–280.

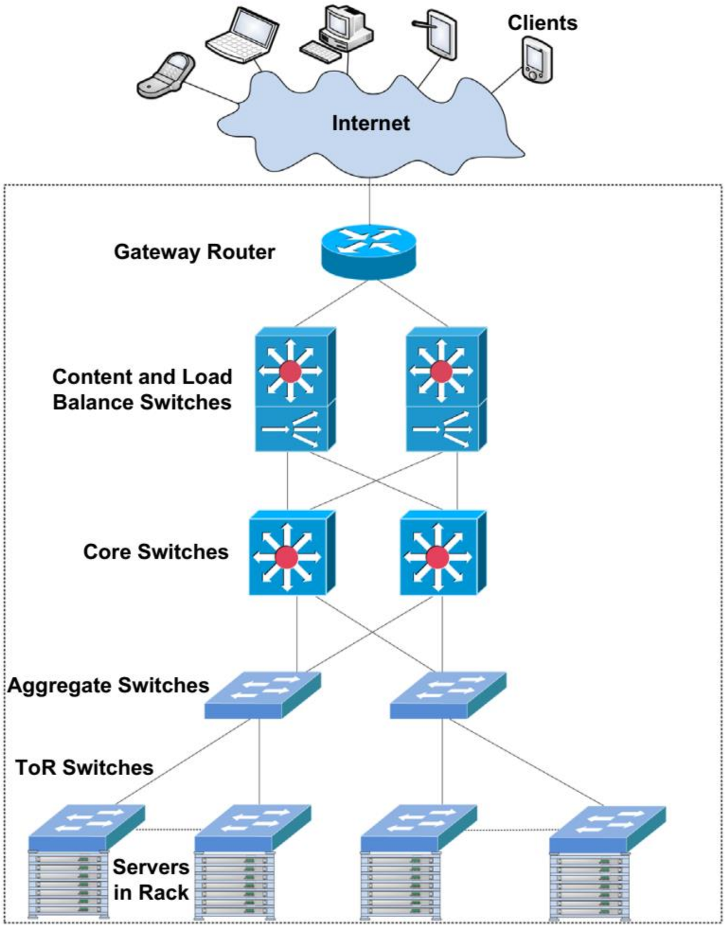


Fig. 1. Traditional architecture of data center networks [1].

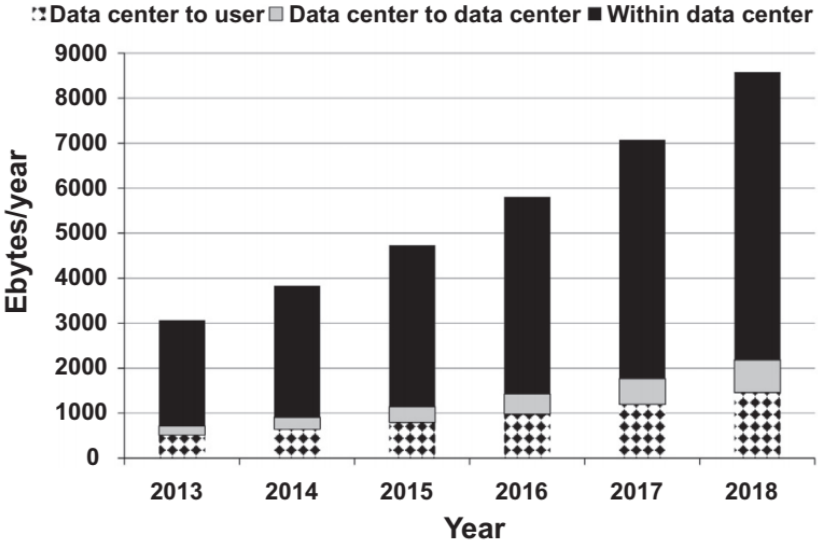


Fig. 2. Traffic movement projection in DCNs from 2012 to 2017 [3].

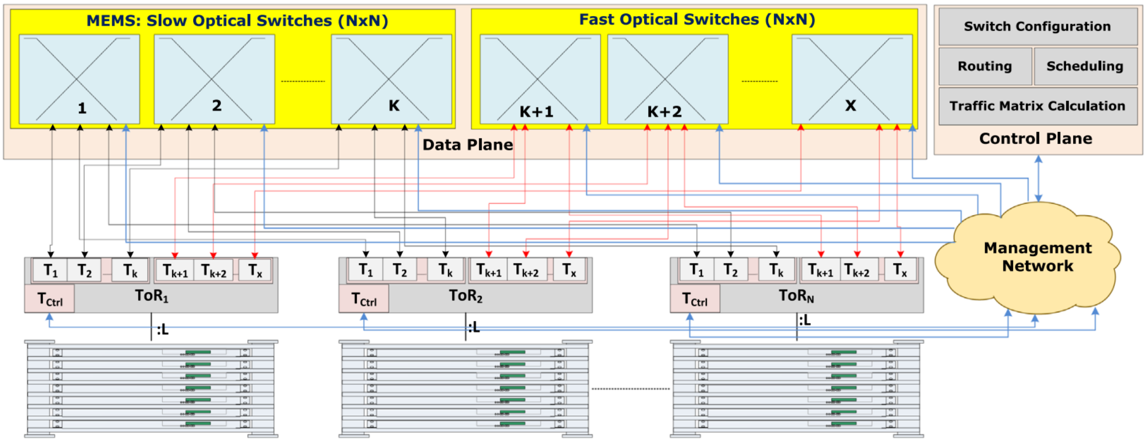

MEMS: Slow Optical Switches (NxN)
1
2
K
Fast Optical Switches (NxN)
K+1
K+2
X
Data Plane
Switch Configuration
Routing
Scheduling
Traffic Matrix Calculation
Control Plane
Management Network
T_1 T_2 T_k T_{k+1} T_{k+2} T_x
T_{Ctrl}
ToR_1
ToR_2
ToR_N
:L

Fig. 3. Proposed architecture: HOSA

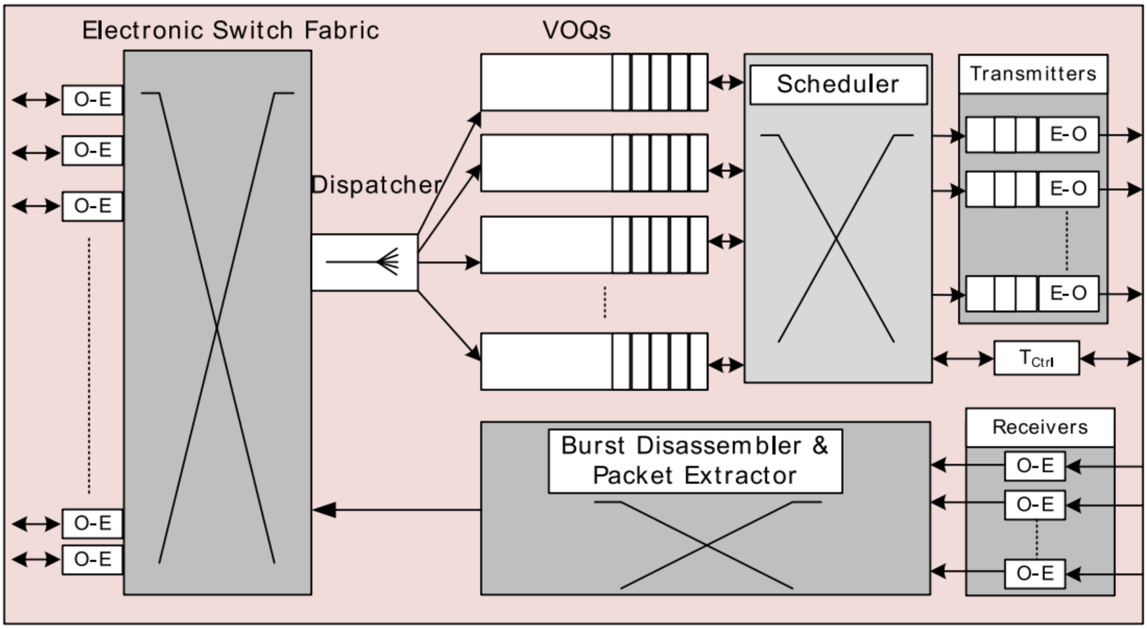


Fig. 4. ToR switch design

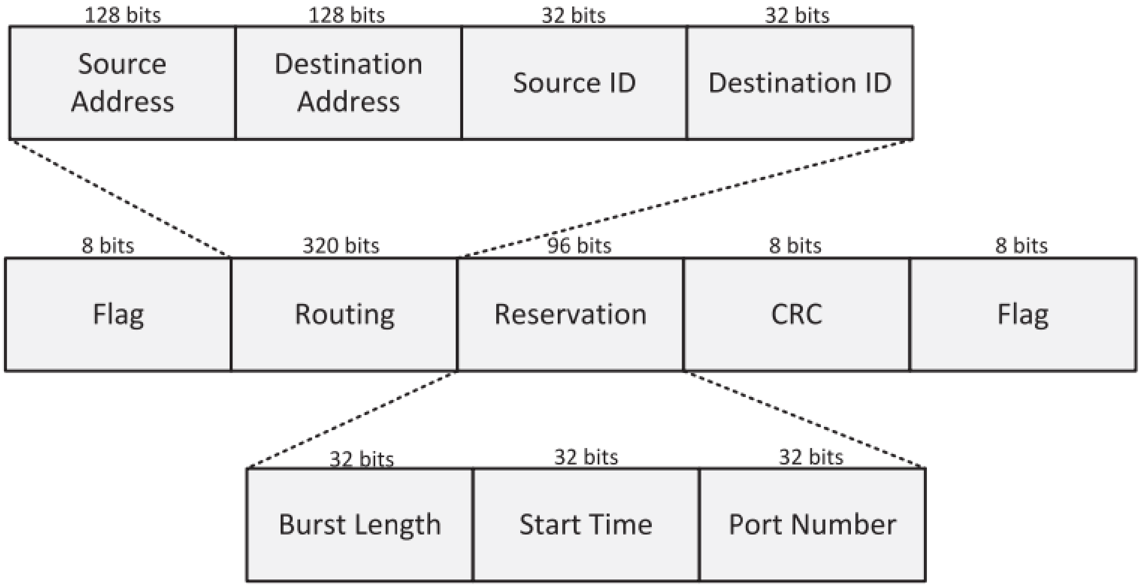
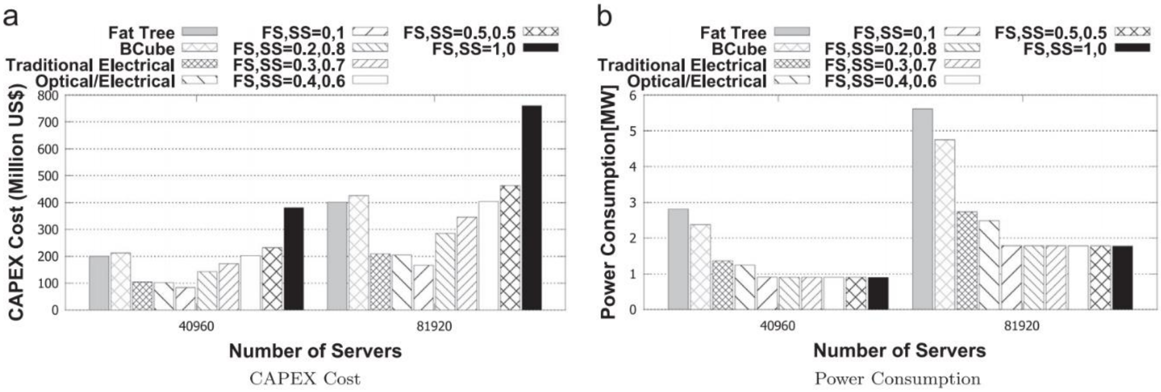


Figure 5: Control packet format



CAPEX Cost

Power Consumption

Figure 6: Total CAPEX cost and power consumption of different interconnection networks with respect to different values for the number of servers: (a) CAPEX cost. (b) Power consumption.

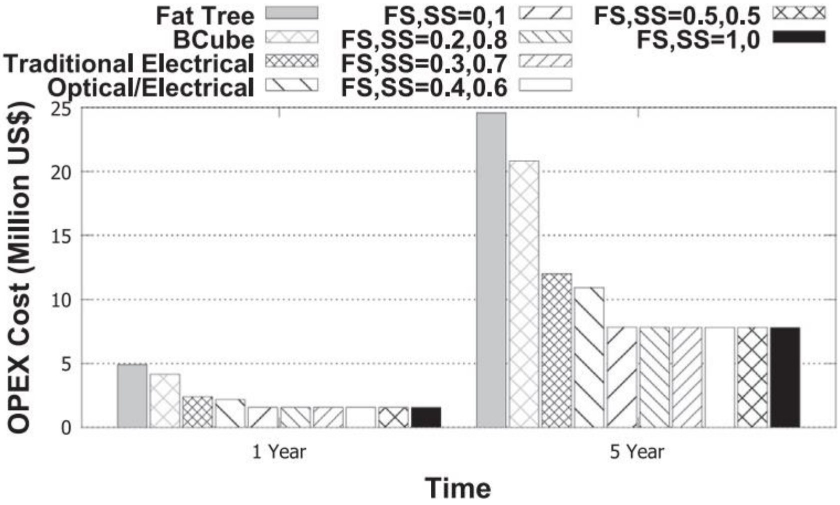


Figure 7: Total OPEX cost of different interconnection networks with respect to years using 40,960 servers.

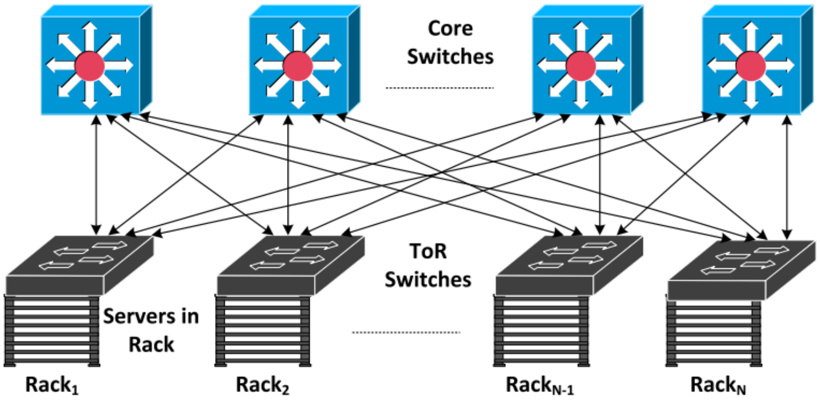
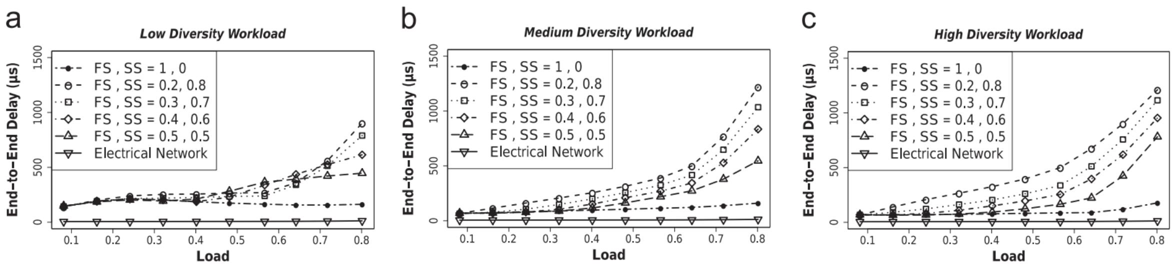


Figure 8: Topology diagram for the baseline traditional electrical network (leaf-spine topology).

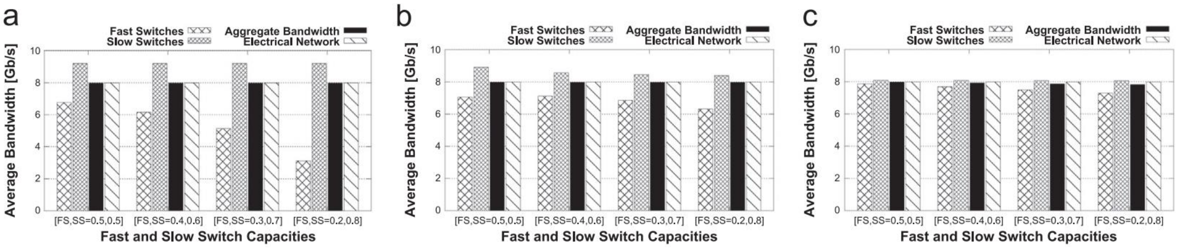


(a) TDC=1

(b) TDC=10

(c) TDC=20

Figure 9: Load vs. end-to-end delay with different capacities of fast and slow switches and with respect to different TDC values: (a) TDC=1, (b) TDC=10, and (c) TDC=20.



TDC=1

TDC=10

TDC=20

Figure 10: Average bandwidth (Gb/s) with different capacities of fast and slow switches and with respect to different TDC values: (a) TDC=1, (b) TDC=10, and (c) TDC=20.

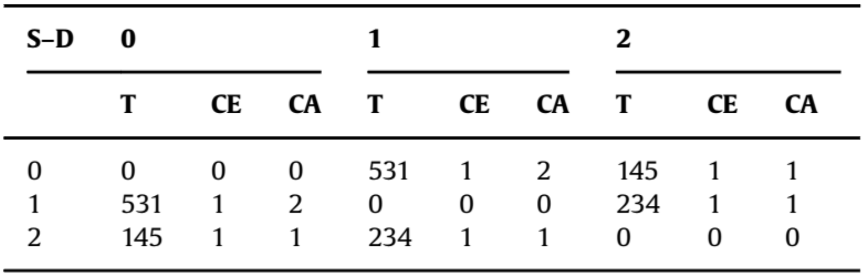
Tables

Table 1

Requirements for data centers [5,3,1].

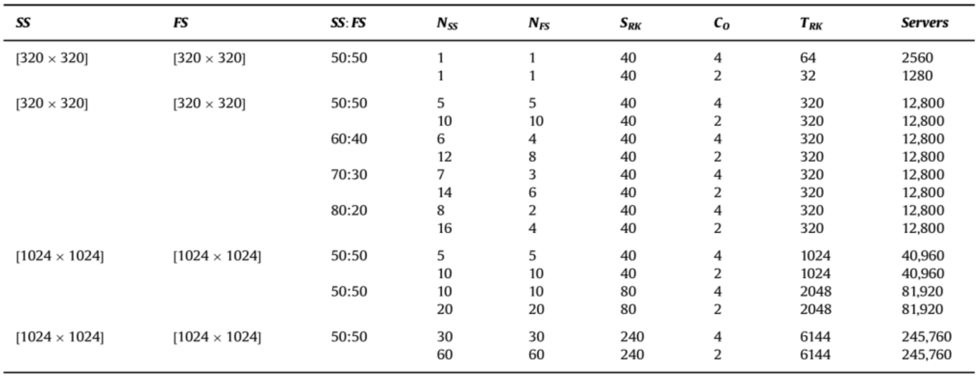
Year	Peak performance (PF)	Power consumption (MW)	Equipment cost	Bandwidth (Pbytes/s)
2012	10	5	$225M	1
2016	100	10	$350M	20
2020	1000	20	$500M	400

Table 2
Matrix table



S–D	0			1			2		
	T	CE	CA	T	CE	CA	T	CE	CA
0	0	0	0	531	1	2	145	1	1
1	531	1	2	0	0	0	234	1	1
2	145	1	1	234	1	1	0	0	0

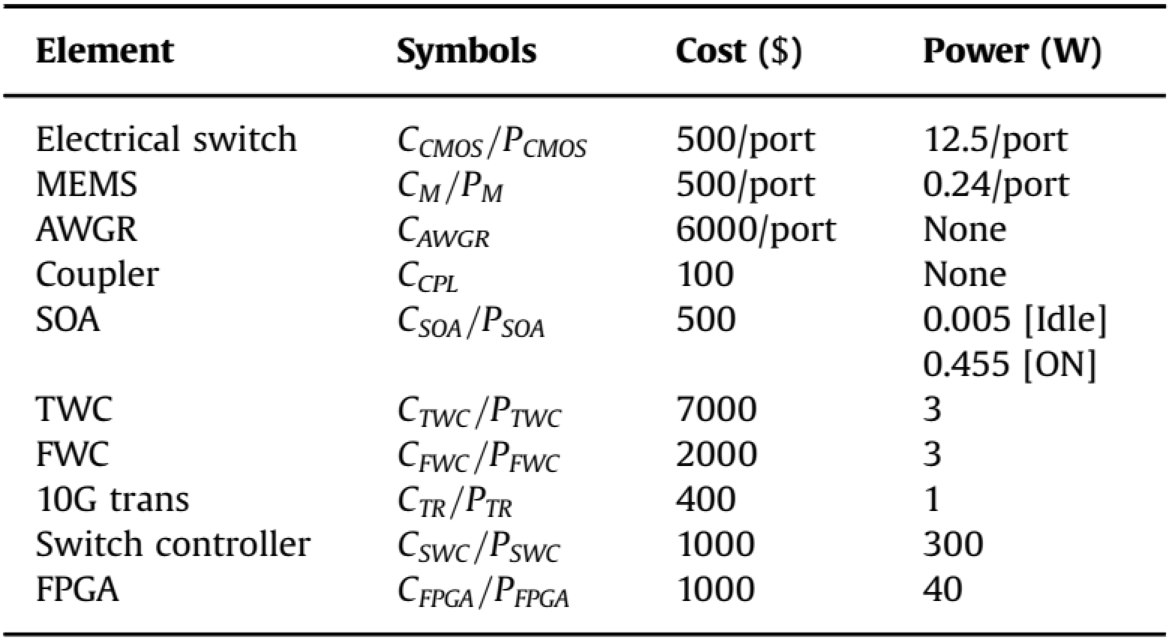
Table 3
Scalability table



SS	FS	SS: FS	N_{SS}	N_{FS}	S_{RK}	C_O	T_{RK}	Servers
[320 × 320]	[320 × 320]	50:50	1	1	40	4	64	2560
			1	1	40	2	32	1280
[320 × 320]	[320 × 320]	50:50	5	5	40	4	320	12,800
			10	10	40	2	320	12,800
		60:40	6	4	40	4	320	12,800
			12	8	40	2	320	12,800
		70:30	7	3	40	4	320	12,800
			14	6	40	2	320	12,800
		80:20	8	2	40	4	320	12,800
			16	4	40	2	320	12,800
[1024 × 1024]	[1024 × 1024]	50:50	5	5	40	4	1024	40,960
			10	10	40	2	1024	40,960
		50:50	10	10	80	4	2048	81,920
			20	20	80	2	2048	81,920
[1024 × 1024]	[1024 × 1024]	50:50	30	30	240	4	6144	245,760
			60	60	240	2	6144	245,760

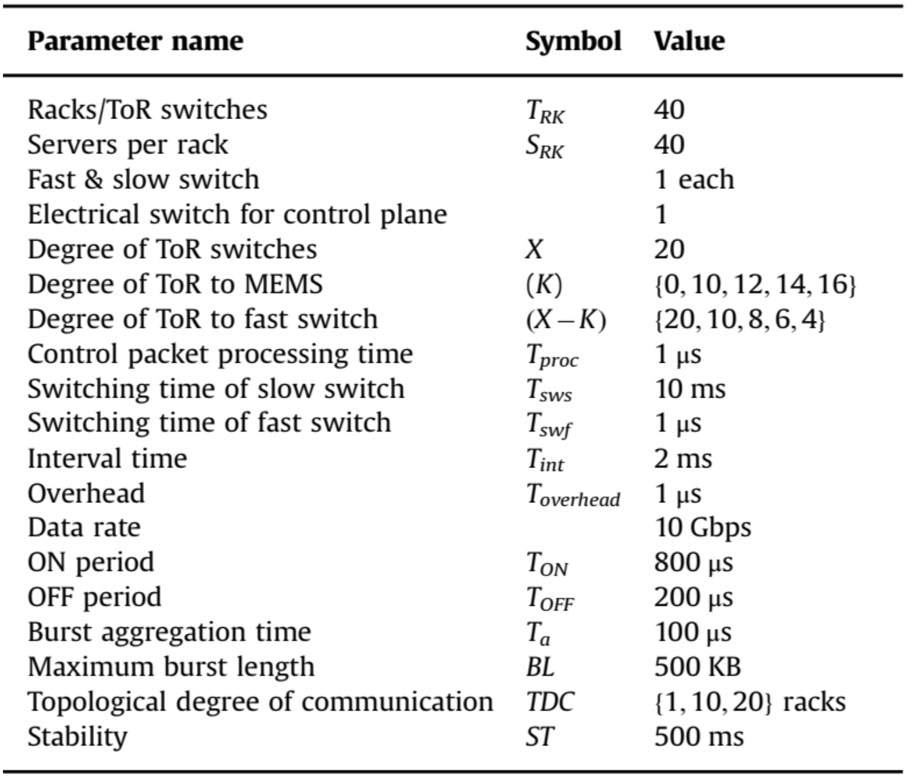
Table 4

Cost and power consumption of network elements.



Element	Symbols	Cost ($)	Power (W)
Electrical switch	C_{CMOS}/P_{CMOS}	500/port	12.5/port
MEMS	C_M/P_M	500/port	0.24/port
AWGR	C_{AWGR}	6000/port	None
Coupler	C_{CPL}	100	None
SOA	C_{SOA}/P_{SOA}	500	0.005 [Idle] 0.455 [ON]
TWC	C_{TWC}/P_{TWC}	7000	3
FWC	C_{FWC}/P_{FWC}	2000	3
10G trans	C_{TR}/P_{TR}	400	1
Switch controller	C_{SWC}/P_{SWC}	1000	300
FPGA	C_{FPGA}/P_{FPGA}	1000	40

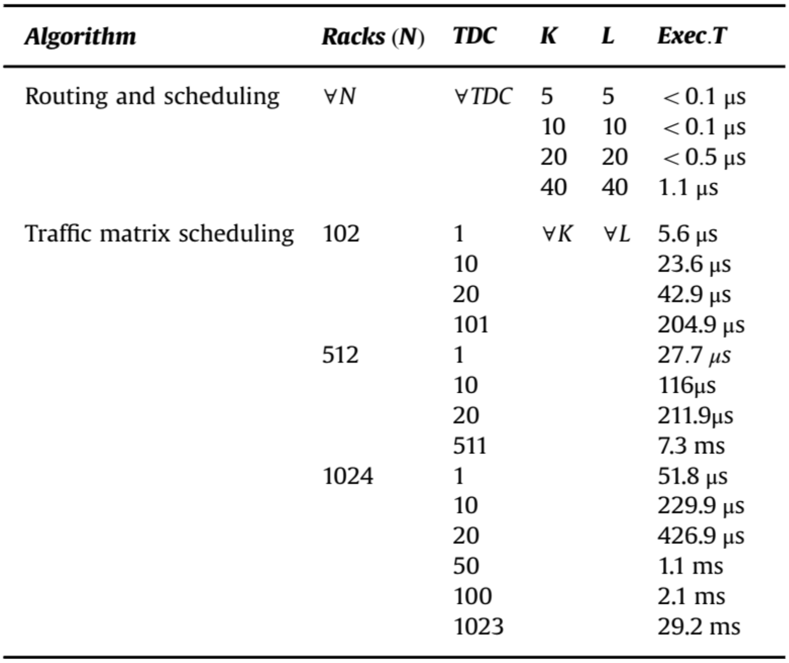
Table 5
Simulation parameters



Parameter name	Symbol	Value
Racks/ToR switches	T_{RK}	40
Servers per rack	S_{RK}	40
Fast & slow switch		1 each
Electrical switch for control plane		1
Degree of ToR switches	X	20
Degree of ToR to MEMS	(K)	$\{0, 10, 12, 14, 16\}$
Degree of ToR to fast switch	$(X-K)$	$\{20, 10, 8, 6, 4\}$
Control packet processing time	T_{proc}	1 µs
Switching time of slow switch	T_{sws}	10 ms
Switching time of fast switch	T_{swf}	1 µs
Interval time	T_{int}	2 ms
Overhead	$T_{overhead}$	1 µs
Data rate		10 Gbps
ON period	T_{ON}	800 µs
OFF period	T_{OFF}	200 µs
Burst aggregation time	T_a	100 µs
Maximum burst length	BL	500 KB
Topological degree of communication	TDC	$\{1, 10, 20\}$ racks
Stability	ST	500 ms

Table 6

Performance of the algorithms in the control plane.



Algorithm	***Racks*** (***N***)	***TDC***	***K***	***L***	***Exec.T***
Routing and scheduling	$\forall N$	$\forall TDC$	5	5	<0.1 μs
			10	10	<0.1 μs
			20	20	<0.5 μs
			40	40	1.1 μs
Traffic matrix scheduling	102	1	$\forall K$	$\forall L$	5.6 μs
		10			23.6 μs
		20			42.9 μs
		101			204.9 μs
	512	1			27.7 *μs*
		10			116μs
		20			211.9μs
		511			7.3 ms
	1024	1			51.8 μs
		10			229.9 μs
		20			426.9 μs
		50			1.1 ms
		100			2.1 ms
		1023			29.2 ms