# Enhancing Performance of Cloud Computing Data Center Networks by Hybrid Switching Architecture

Xiaoshan Yu, Huaxi Gu, Kun Wang, and Gang Wu

***Abstract*—Cloud computing services have driven a new design of data center networks. Hybrid switching architecture is one of the promising solutions since it makes better tradeoff between the network performance and technical feasibility. However, as the existing hybrid networks only deploy one-hop optical circuit switching (OCS) in the top layer, the flexibility and scalability is limited. To address this problem, a distributed OCS model is proposed. To reduce the high blocking ratio, the WDM and SDM technologies are introduced to increase the connectivity of the optical network. Moreover a multi-wavelength optical switch based on microring resonators is designed to enable the fast switching. Based on this model, the multi-rooted tree based hybrid architecture with deep integration of optical connection is constructed. A new way to solve the mixed traffic scheduling problem is also provided by delivering the small flows and large flows through the different networks. The simulation results indicate that the multi-rooted tree based hybrid architecture achieves better performance under various traffic patterns. It also introduces less control overhead compared with the existing traffic scheduling schemes.**

***Index Terms*—Data center, flow scheduling, hybrid switching, optical network.**

## I. Introduction

In the past several years, cloud computing emerges as a new way to provide flexible services. As the fundamental infrastructure of cloud computing, data center faces various challenges. In cloud environment, the network plays a more important role than ever before. However, traditional data center network cannot satisfy the requirements of new cloud computing applications. Some novel electronic networks, such as fat tree [1], VL2 [2], BCube [3], DCell [4] etc., are proposed. Although notable improvements have been achieved, some issues, such as complicated management and unaffordable power consumption, cannot be well solved in electronic domain. Photonic technology provides great potential for cloud computing by exploiting the high bandwidth and energy efficiency. All-optical networks, such as DOS [5], Petabit [6], and MIMO-OFDM architecture [7], employ optical packet switching (OPS) to build high performance and agile network. As the key functional elements of OPS are not commercially available, these architectures promise a forward-looking solution for further data center networks [8]. The hybrid network architecture, which augments the electrical packet switching (EPS) with optical circuits, may fill the gap between the electronic architectures and all-optical architectures. Without making drastic changes to the existing network, the hybrid network incrementally upgrades the data center network with optical devices, thus maintaining high feasibility and reducing the cost. The existing hybrid networks, Helios [9] and C-through [10] employ OCS to deliver aggregation traffic between different PODs or racks. However, most of these research works employ one-hop OCS or OPS switching architecture for data center. A single layer of optical switches or a single optical switching fabric is deployed to interconnect huge number of racks. Although these architectures can greatly simplify the data center network, it is not easy to design a high performance and low cost optical switching fabric, which is capable of directly interconnecting thousands of racks. As a result, in practice the optical switches are often deployed in the core layer, thus limiting the scale of the network. To integrate a flexible and scalable optical network into data center, a distributed OCS over a multi-hop optical network is proposed in this paper. The multi-rooted tree based hybrid architecture is used to implement this OCS model. Low port count optical switches form a multi-layer optical network which can expand to larger scale. Furthermore, as the optical devices can be integrated into the lower layer, based on our proposed circuit switching, the optical connection can be established for each flow. To overcome the high blocking ratio of the OCS, three measures are proposed in this paper. Firstly, multiple wavelengths are introduced to enable better sharing of the optical paths. Secondly, an exploring scheme is designed to make better utilization of all available resources. Finally, a new optical switch based on microring resonators (MR) is designed to enable the fast switching configuration.

This paper also proposes a new way to schedule the mixed traffic flows in multi-rooted tree. As the natural traffic pattern in data center is characterized by a mixture of large and small flows [11]. Traditional multi-path routing algorithms cannot balance traffic load on multiple paths. New traffic scheduling schemes [12]–[14] may either introduce too much control overhead or potentially lead to packet disorder. As the multi-rooted tree based hybrid architecture consists of an optical network and an electronic network, a simple but effective traffic scheduling strategy is designed. A majority of large flows are delivered

Manuscript received December 15, 2013; revised February 6, 2014 and March 24, 2014; accepted March 24, 2014. Date of publication March 31, 2014; date of current version May 7, 2014. This work was supported by the National Science Foundation of China Grants 61070046 and 61334003, the Fundamental Research Funds for the Central Universities Grant K5051301003, and the 111 Project Grant B08038.

X. Yu, H. Gu, and G. Wu are with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: yuxiaoshan@stu.xidian.edu.cn; hxgu@xidian.edu.cn; xdantlks@163.com).

K. Wang is with the School of Computer Science, Xidian University, Xi'an 710071, China (e-mail: kwang@mail.xidian.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2014.2314693

through the optical network while the small flows are routed in the electronic network. The evaluation indicates that multi-rooted tree based hybrid architecture can achieve better performance while introduces less control overhead.

## II. Background

### A. The Integration of Optics in the Data Center

Some recent works also attempt at integrating optics into data center networks. An FSO (Free-Space Optics) based data center network is proposed in [15]. Visible or infra-red laser beams are used to implement the data links between different racks. The FSO transceivers are placed on the top of each rack (ToR). The inter-rack traffic will firstly be converted into the light beam. Then the light beam is directed to the ceiling mirrors and reflected to the destination rack. A centralized Topology Manager is proposed to dynamically reconfigure the inter-rack connection. After each reconfiguration, a Routing Manager is responsible for creating routing table entries for each ToR. This architecture can achieve better performance and greatly eliminate the cabling complexity. Our architecture, in contrast, uses the WDM fibers and microring-based optical switches to construct the optical network. As the optical switch can make flexible reconfigurations to direct the optical signal, an optical path can be established over multiple hops. In [16], an OCS prototype with the fast switching time is designed and implemented. A centralized scheduling approach is designed to control all optical switches. The total scheduling duration is divided into several time slots. After being informed the short scheduling of impending circuit configurations, each ToR can fill their traffic flows into the proper timeslots. This novel optical circuit scheduling approach can effectively increase the bandwidth utilization and make more traffic flows benefit from the short-lived circuits. In comparison, our hybrid architecture also adopts the fast switching fabric, but here a distributed control scheme is proposed to establish the optical connection. As each optical switch contains a control module to configure the switching fabric based on the control packets, the optical paths can be setup and be torn down independently.

### B. Scheduling the Mixed Traffic in Multi-Rooted Tree

Multi-rooted tree topologies, such as folded-Clos and fat tree, have been widely adopted in data center network. In these networks, multiple paths can be found between most source-destination pairs. To make full use of the aggregate bandwidth, flow-level Equal-Cost Multi-Path routing (ECMP) and Valiant Load Forwarding (VLB) are employed [12]. Unfortunately these routing algorithms cannot achieve anticipated performance. For each source-destination pair, flow-level ECMP and VLB just ensure that there are equal numbers of flows on all available paths. They are unaware of the traffic load of each flow. In cloud computing data center, the real traffic load consists of both mouse flows and elephant flows [2], [11]. In this case, substantial bandwidth is wasted because the multipath routing algorithms may lead to unbalanced load distribution. To solve this problem, several load-sensitive traffic scheduling schemes are proposed. A centralized scheduler is employed in Hedera [12] to make optimal scheduling on elephant flows. In contrast, DARD [13] adopts a distributed approach to schedule network traffic. In [14], a reactive reroute strategy based on Congestion Notification strategy is proposed. These solutions can effectively improve the network performance and make a better utilization of available bandwidth. However, some limitations, such as the limited scalability and too much control overhead, still exist. Besides, the delay for new path selection and setup should not be neglected. For example, it nearly takes NOX 10 ms to insert the flow entry into the switches [17].

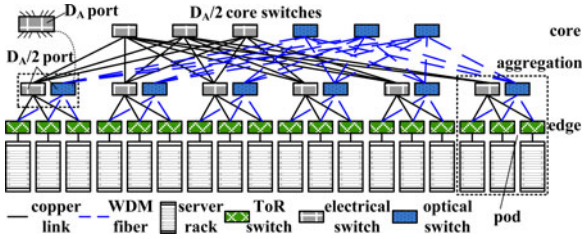


Fig. 1. The folded-Clos based hybrid switching architecture.

## III. The Hybrid Switching Architecture and Distributed OCS Model

As the proposed OCS model also utilizes some properties of the multi-rooted tree topologies, to make better illustration, we firstly describe the network architecture and then introduce the OCS model and its implementation.

### A. The Network Architecture

To embed an optical network in multi-rooted trees, the edge switches should be crammed with the optical transceivers to make E/O and O/E conversions. Then in the higher layers of the hierarchy, a portion of electrical switches are replaced by the optical switches. Finally, the optical fibers are used to connect the optical switches in different layers. Specifically, we take folded-Clos and fat tree as examples to show how to build the hybrid networks based on different topologies.

Traditional electronic folded-Clos, which is employed in VL2, consists of the edge layer, the aggregation layer and the core layer from bottom to top. In the top two layers, the $D_A$-port aggregation switches and $D_C$-port core switches are interconnected to form a complete bipartite graph. In the bottom layer, each ToR switch is connected to the two aggregation switches for redundancy. As shown in Fig. 1, to accommodate optical interconnections, the traditional architecture should make some adjustments. In the edge layer, one upstream port of the ToR switch is configured with the optical transceiver. Thus each ToR switch needs to be connected to one electrical switch and one optical switch in the upper layer. To meet this demand, original $D_A$-port aggregation switch is replaced by a pair of $D_A/2$-port switches, one electrical switch, and one optical switch. These two switches form an aggregation switch group. In the downstream direction, each aggregation switch group is connected to $D_A/4$ ToR switches. To further interconnect these aggregation

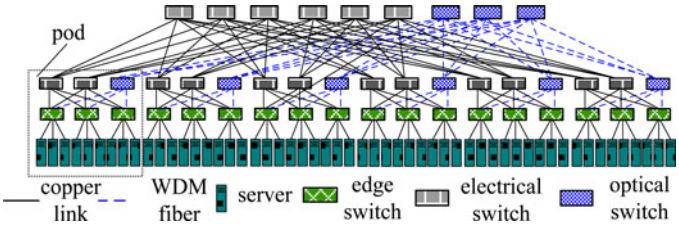


Fig. 2. The fat tree based hybrid switching architecture.

switch groups, half of the electrical core switches are replaced by the optical ones. In the upstream direction of the aggregation switch group, the electrical switch is connected to every electrical core switch and the optical switch is connected to every optical core switch. Fig. 1 illustrates an example of folded-Clos based hybrid architecture with $D_C = 6$ and $D_A = 12$.

Folded-Clos based hybrid switching architecture divides the network into the EPS portion and OCS portion. However, as various services are running in cloud computing data center, the proportion of the elephant flows may vary in different data centers [11]. To make better adaption, the proportion of OCS, responsible for delivering elephant flows, should be reconfigured according to the different requirements. Fat tree based hybrid switching architecture can achieve this flexibility. If new applications are deployed with more elephant flows, more optical devices can be employed. Fig. 2 gives an example of fat tree based hybrid switching architecture with the port number $k = 6$. The architecture consists of the same three layers: Edge layer, aggregation layer, and core layer. At the bottom, $k/2$ servers are connected to an edge switch to form a subnet. Then $k/2$ edge switches and $k/2$ aggregation switches are interconnected to form a pod. Each edge switch has one upstream port connected to each of $k/2$ aggregation switches in the pod. As each edge switch is configured with $m$ optical upstream ports (in Fig. 2 $m = 1$) and $(k/2 - m)$ electrical upstream ports, $m$ optical switches and $(k/2 - m)$ electrical switches are employed in the aggregation layer of each pod. Then in the top layer, core switches are introduced to deliver traffic between different pods. Totally each pod has $(k/2)^2$ upstream ports connected to the core layer, of which $(mk/2)$ ports are optical ones. As each upstream port is connected to one core switch, accordingly there are $(k/2)^2$ core switches, $(mk/2)$ optical ones and others electrical ones, deployed in the core layer.

### *B. The Communication Process and Distributed OCS Model*

In the hybrid architecture, two switching strategies, EPS and OCS, are employed to process different traffic flows. Thus the first issue is how to detect and classify traffic flows. The method proposed in [18] is introduced to identify the elephant flows in the servers. Then in the edge switch (or ToR switch), the elephant flows are further classified into three types. The *internal subnet E-flow* refers to the elephant flow with the source and the destination in the same subnet (or the same rack). The *internal pod E-flow* refers to the elephant flow with the source and the destination in different subnets of the same pod. The *external pod E-flow* refers to the elephant flow with the source and the destination in different pods. When new traffic flows are generated, the server directly sends them to the edge switch (or ToR switch). The edge switch (or ToR switch) then takes different actions based on the flow types. The mouse flows will be routed by ECMP or VLB in the electronic network. The *internal subnet E-flows* will be directly forwarded to the destinations. For the other two types of *E-flows*, as they need to be delivered in optical network, the edge switch (or ToR switch) will set up optical paths for them.

In OCS model the optical signal should be directly transmitted from the source to the destination, without O/E and E/O conversions in the intermediate switches. The control system is needed to manage the establishment and releasing of the optical connections. A distributed control system is designed as it is more scalable and flexible than the centralized control system in the large scale data center. As shown in Fig. 3(a), each optical switch contains a control module to configure the optical switching fabric. Moreover, low bandwidth copper links exist between the interconnected optical switches. These copper links combined with the control modules form an electronic control network which is overlapped with the optical data transmission network. The control packets are transmitted in the electronic control network, notifying the configuration information and building the optical path over multiple hops. As the control packets just need to take simple control information, they are small in size. The control module only needs a few buffers to temporarily store the blocked packets.

Traditional OCS is a circuit-switch-like technique. The source node will send a setup packet to reserve an optical path before data transmission. When the setup packet arrives at the destination, an ACK packet is sent back to inform the accomplishment of path setup. Finally, a teardown packet is sent to release the occupied links. However, as the data transmission network of the OCS model does not buffer any signal in the intermediate nodes, in traditional OCS, the optical links can only serve one source-destination pair. Other source-destination pairs which have the overlapped links with the established one have to wait until the required optical links are released, thus limiting the connectivity of the optical network and inducing a high block ratio. The WDM technique can be used to make several source-destination pairs share one optical link. Furthermore, the path diversity of multi-rooted tree facilitates forming multiple SDM channels for any source-destination pair. To effectively utilize the available resources, an improved distributed OCS model, combined with WDM and SDM, is proposed in this section. In this model, the source edge switch sends multiple setup packets to explore the idle wavelengths along all available paths. The destination edge switch is responsible for selecting one idle wavelength. Then an ACK packet is sent back to configure the corresponding optical switches and build an optical path for a certain elephant flow.

To be specific, for an *external pod E-flow*, assume there are $N_s$ optical paths. Each path is exactly bounded with one upstream port of the aggregation switches in the source pod. For fat tree topology, $N_s = (mk/2)$. For fold-Clos, $N_s = D_A/2$. To select one available path and one idle wavelength, $N_s$ setup packets are sent from the source edge switch. The $i$th setup packet is routed by the control modules along the $i$th optical path. At each hop,

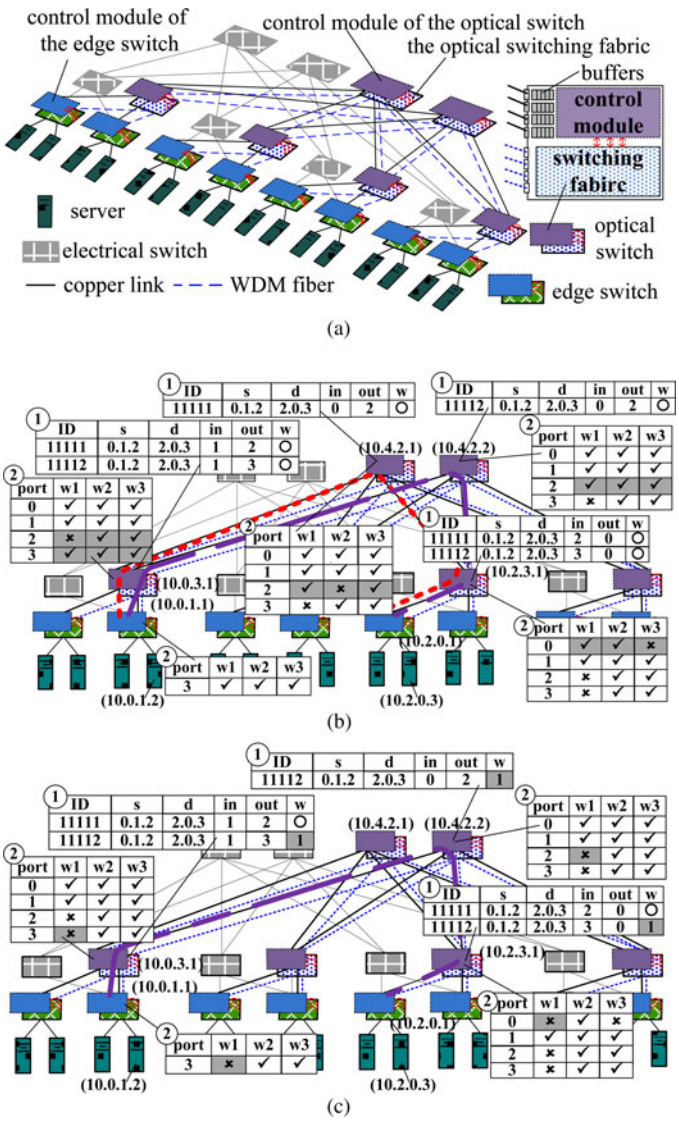


Fig. 3. An example of the optical path setup in fat tree based hybrid switching architecture (table ① is the trace table, s-source, d-destination, in-input port, out-output port, w-used wavelength; table ② is the wavelength state table, wi-wavelength i:_ Symbol × indicates wavelength is occupied. Symbol ✓ indicates wavelength is available. Symbol ○ indicates wavelength is not determined). (a) The electronic control network and optical data transmission network in the distributed OCS model. (b) The procedure of wavelength exploration. (c) The procedure of wavelength reservation.

it records the available wavelengths of the intermediate switch. These setup packets will arrive at the destination at different time. The destination edge switch checks the first arrived setup packet to find whether the corresponding optical path has idle wavelengths. If so, an idle wavelength is randomly selected and an ACK packet is sent back. The following arrived setup packets are dropped. If the first arrived setup packet shows that there is no idle wavelength along its path, the destination will wait for the next arrived setup packet. It does not send ACK packet until it finds an idle wavelength. In the worst case, the destination may find that all wavelengths are occupied when it has collected all $N_s$ setup packets. Then a NACK packet is sent back to notify the failure of path setup. The ACK packet is sent back along the original path. It reserves the required wavelength and configures the optical switching fabric hop by hop. Only when the required wavelength is occupied at a certain node, the ACK packet is blocked in the control module temporarily. It cannot be free from the blocking state until the required wavelength is released. When the source edge switch receives the ACK packet, it sends corresponding *E-flow* using the assigned wavelength. Finally, a teardown packet is sent to release the reserved path and wavelength. Once the source edge switch receives a NACK packet, it will restart the procedure of path setup after a random time interval.

A similar process is carried for the *internal pod E-flows*. However, in this case, only one hop exists between the source and the destination. The number of available optical paths also becomes less.

Fig. 3 uses fat tree based hybrid switching architecture to illustrate the implementation details of the improved OCS. This hybrid architecture adopts the same addressing rules as traditional fat tree [1]. The port numbers of each switch are labeled as $0, 1 \ldots, k-1$, from bottom to top, left to right. Each optical switch maintains a trace table and a wavelength state table in the control module. The trace table records corresponding information for optical path setup. The wavelength state table is used to identify the available wavelengths at each output port. As shown in Fig. 3(b), assume an elephant flow needs to be transmitted from (10.0.1.2) to (10.2.0.3). All the packets belonging to this flow are firstly sent to the edge switch (10.0.1.1). When these packets arrive at the edge switch, they are stored in a dedicated queue. At the same time, the edge switch begins to establish an optical path for this flow. Two setup packets with unique local IDs 11111 and 11112 are generated to find idle wavelengths along two optical paths. Optical path one, corresponding to setup packet 11111, passes through switches (10.0.1.1), (10.0.3.1), (10.4.2.1), and (10.2.3.1). Optical path two, corresponding to setup packet 11112, passes through switches (10.0.1.1), (10.0.3.1), (10.4.2.2), and (10.2.3.1). For optical path one, wavelength 1 is not available at the output port 2 of switch (10.0.3.1). Wavelength 2 is not available at the output port 2 of switch (10.4.2.1). Wavelength 3 is not available at the output port 0 of switch (10.2.3.1). Thus setup packet 1111 takes no idle wavelengths when it arrives at the destination edge switch. Setup packet 11112 will arrive at the destination with idle wavelengths 1 and 2. Then an ACK packet with ID 11112 is generated. It is forwarded back along optical path two, taking a randomly selected wavelength (assume wavelength 1). As shown in Fig. 3(c), at each hop, the switch checks the required wavelength. Then it configures the optical switching fabric and modifies corresponding information in two tables. When the ACK packet arrives at the source edge switch, an optical path is created. Finally, a teardown packet is sent to delete corresponding information and reset the wavelength state.

## IV. The Multi-Wavelength Optical Switch

Based on aforementioned communication strategy, a multi-wavelength optical switch is required to achieve flexible operation on each wavelength. Fig. 4 shows the basic structure of

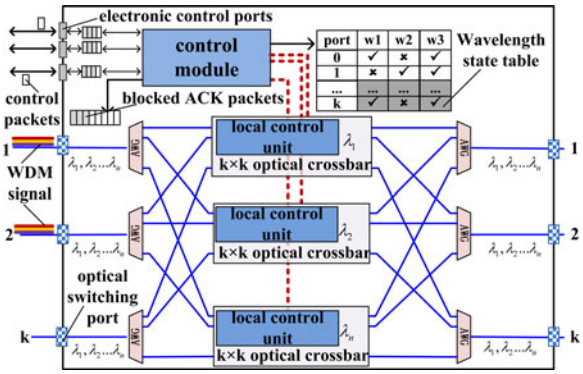


Fig. 4. The basic structure of the multi-wavelength optical switch $(k, n)$;$k$ is the number of switching ports, $n$ is the number of wavelengths.

the optical switch with $k$ switching ports and $n$ wavelengths. As stated before, each optical switch contains a control module to configure the switching fabric based on the control packets. Moreover, the optical switch contains electrical control ports to deliver these control packets from one to another. The main switching fabric consists of 2k AWG modules and $n$ $k \times k$ optical crossbar modules. At the input port, 1:$n$ AWG module separates the WDM signal into single wavelength signals. Then these signals are delivered into different crossbars and guided to the right output ports. Finally, these signals are multiplexed again by the $n$:1 AWG modules.

The $k \times k$ optical crossbar can be built with various photonic materials. However, to earn more benefits from the optical interconnection, the optical crossbar should enable fast reconfiguration time and maintains high quality of signal transmission. The switching time is crucial to the delay of path setup, which further determines the bandwidth utilization of an optical path. Specifically, the bandwidth utilization $\eta$ can be calculated by the following equation

$$\eta = \frac{t_{\text{setup}}}{t_{\text{setup}} + t_{\text{trans}}} \tag{1}$$

where $t_{\text{setup}}$ is the delay of path setup, $t_{\text{trans}}$ is the duration for the data transmission. The commercial MEMS based optical crossbar suffers from relative long reconfiguration time (10–100 ms). It will take at least 30 ms to establish an optical path if the fat tree based hybrid architecture deploys MEMS optical switches. Based on equation (1), only the size of the elephant flow is larger than 300M bits, the bandwidth utilization can be greater than 50%. This may greatly limits the optical benefits. SOA (semiconductor optical amplifier) is capable of fast reconfiguration time [19]. However, it is unfit for building high-radix switches since the SOA based switching fabric adopts broadcast-and-select structure. The power loss becomes higher as the port number increases.

In our design, the silicon photonic device MR is introduced to build the $k \times k$ optical crossbar. Its compact size, energy efficiency, CMOS compatibility and sub-nanosecond switching time enable a flexible and high performance switching behavior [20]. As shown in Fig. 5, MR-based optical switch consists of waveguides and MRs. The MR is placed at the intersection of two crossed waveguides. With a p-i-n junction embedded in,

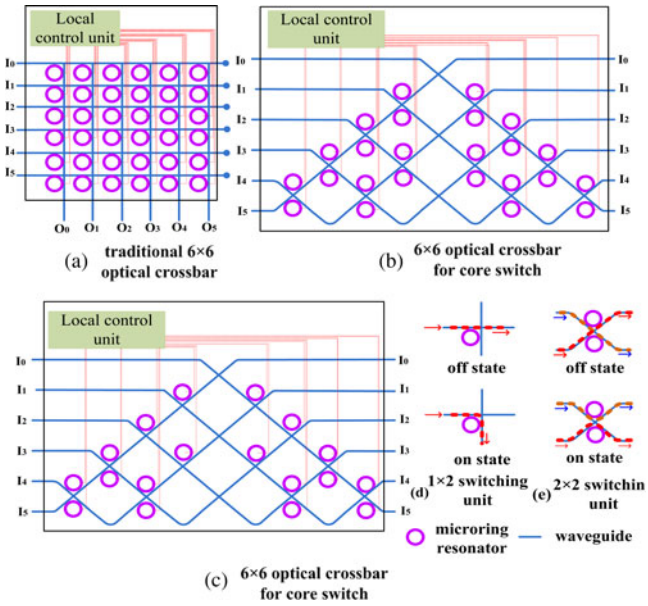


Fig. 5. The structure of MR based optical crossbar.

the refraction index of MR can be changed by the high carrier injection using a bias voltage. This operation can actively set MR in an off-state or on-state. When the MR is in on-state, it couples the optical signal into another waveguide. When the MR is in off-state, the optical signal passes by it without disturbance. The on-off configuration of MR can switch the optical signal to the desired output port. For example, an ACK packet requiring wavelength $\boldsymbol{m}$ arrives at the input port $\boldsymbol{i}$ of the optical switch. The control module calculates the output port $\boldsymbol{j}$ based on the source address. Then it checks whether the requiring wavelength is available or not. If yes, the control module notifies the local control unit of the $\boldsymbol{m}$-th $k \times k$ crossbar module to connect port $\boldsymbol{j}$ to port $\boldsymbol{i}$. Based on this notification, the $\boldsymbol{m}$-th local control unit sets corresponding MR to on-state. If the requiring wavelength is occupied, the ACK packet is blocked temporarily. The control module will not forward this ACK packet until the requiring wavelength is released and corresponding configurations are completed.

As shown in Fig. 5(a), traditional fully-connected crossbar consists of $\boldsymbol{k}^2$ MRs and $2\boldsymbol{k}$ waveguides. Although the fully-connected crossbar benefits from the regular structure and simple layout, too many waveguide crossings and MRs are introduced. To optimize the insertion loss, the improved optical crossbars are designed. As shown in Fig. 5(b) and (c), the $2 \times 2$ basic switching unit, with two MRs placed at the waveguide crossing, replaces the $1 \times 2$ switching unit, with one MR placed at the waveguide crossing. Furthermore, some MRs are removed from the crossbar as some switching cases are not needed. For example, a traffic flow never needs to be switched to its original input port. Additionally in the aggregation switch, there is no need for the traffic flows being forwarded between two upstream ports.

Table I makes a quantitative comparison of the basic component cost. As the multistage network can also be used to construct the optical crossbar, Benes network, which consists of the $2 \times 2$ basic unit, is chosen for the comparison.

TABLE I
THE COMPARISON AMONG THE BENES, TRADITIONAL FULLY CONNECTED AND THE IMPROVED SWITCHING FABRICS

| | | MRs | crossings | waveguides |
|---|---|---|---|---|
| Benes | | $k(2\log_2^k - 1)$ | $k^2/4 + k/2 + 2C_{n-1}$ | -- |
| fully-connected | | $k^2$ | $k^2$ | $k^2$ |
| improved | core switch | $k(k-2)$ | $k(k-1)/2$ | $k$ |
| | agg. switch | $3k(k-2)/4$ | $k(k-1)/2$ | $k$ |

$k$ is the number of switch ports and agg. switch is aggregation switch. For Benes, the number of waveguide crossing $C_n$ should be calculated iteratively. The parameter $n = \log_2^k$ and $C_2 = 8$.

TABLE II
THE INSERTION LOSS OF DIFFERENT 8 × 8 CROSSBARS (dB)

| | | maximum insertion loss | minimum insertion loss | average insertion loss |
|---|---|---|---|---|
| Benes | | 3.13 | 1.355 | 2.1625 |
| fully-connected | | 3.07 | 0.76 | 1.9150 |
| improved | core switch | 2.475 | 0.77 | 1.5612 |
| | agg. switch | 1.95 | 0.77 | 1.0937 |

Assuming an optical signal is switched from the port $i$ to port $j$. There are $m$ on-state MRs, $r$ off-state MRs, $p$ waveguide crossings, $q$ waveguide bends along the switching path form port $i$ to $j$. Then the insertion loss the optical signal experienced can be calculated by the following equation:

$$L(i,j) = m \cdot l_{drop} + r \cdot l_{trough} + p \cdot l_{crossing} + q \cdot l_{bend} \quad (2)$$

where $l_{drop} = 0.6dB, l_{trough} = 0.005dB, l_{crossing} = 0.16dB, l_{bend} = 0.005dB / 90^0$ [21]

Compared with Benes, the improved crossbar does not reduce the number of MRs. However, as an example shown in Table II, the insertion loss of the improved crossbar is less than Benes. The main reason is the insertion loss consists of the drop-loss, through-loss, waveguide-crossing loss and waveguide bending loss. The drop-loss, which denotes the loss of an optical signal being coupled into an on-state MR, is much larger than other types of loss. In the improved crossbar, the optical signal only experiences drop-loss once. But in Benes, the optical signal needs to be coupled into the microring several times.

## V. EVALUATIONS

As fat tree topology can provide better path diversity and flexibility, we simulate our proposed distributed OCS model on the hybrid fat tree architecture by OPNET. The electronic fat tree architecture is built for comparison. Each of these networks holds 128 servers. A set of continuously generated packets, with the same source and destination addresses, is used to imitate a traffic flow. The time interval between two sequential packets obeys the negative exponential distribution. Based on the analysis of the nature traffic characteristic in data center [2], the total size of a mouse flow is set to a few Kbytes while the size of an elephant flow is around 1M bytes. The proportions of elephant flows are set to 0%, 1% and 10% respectively. VLB routing algorithm is applied in the electronic fat tree network and the bandwidth of copper link is set to 1 Gbps. For fat tree based hybrid switching architecture, the parameter $m$ is set to 1 and each optical link carries four 10 Gbps wavelengths. Two metrics, the average end-to-end (ETE) delay and network throughput, are measured as a function of the injection rate. The average ETE delay is defined as the average time interval between packets generated at the source server and received at the destination server. The network throughput is defined as the steady bit rate at which all servers receive data from the network.

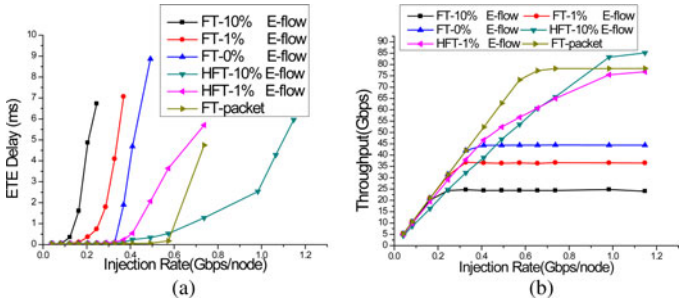


Fig. 6. (a) ETE delay and (b) throughput under random traffic pattern, FT: fat tree; HFT: fat tree based hybrid switching architecture; a% E-flow: traffic with a% elephant flows; FT-packet: fat tree with packet level VLB.

Since the traffic traces of cloud computing data center are not released to public for the reasons of safety and business secret, three synthetic traffic patterns, which introduced in [1], are applied in the simulation. These traffic patterns define the spatial distribution of the traffic flows. Each pattern is defined as follows: (1) Random, where a server sends traffic flows to a random selected server in the network. (2) Stride($i$), where a server with index $x$ constantly sends traffic to a server with index $((x + i)$ *mod* number_of_servers). The index $x$ is used to denote the server number (in [0, $k^3/4$ – 1], from left to right). (3) Staggered Prob ($p_{sub}$, $p_{pod}$), where a server sends traffic to another server in the same subnet with probability $p_{sub}$, and to another server in the same pod with probability $p_{pod}$.

Firstly, the network performance of flow-level and packet-level VLB routing algorithms is compared under the random traffic pattern. The result shown in Fig. 6 confirms that the flow-level VLB cannot achieve ideal performance. When the packet-level VLB routing algorithm is applied in fat tree, the saturation point, which indicates the maximum injection rate to keep the network stable and well-behaved, can approximately reach to 0.6 Gbps/node. However, flow-level VLB degrades the network performance. When network is full of mouse flows, the saturation point is early encountered at 0.3 Gbps/node. Worse performance is observed when elephant flows are introduced. It is reasonable since the packet collision will affect the queuing delay and further determines the ETE delay. In packet-level VLB, all packets randomly select the output ports. But in flow-level VLB, if the heads of two flows choose the same output port, the congestion will last for several packets and cannot be solved. Thus the probability of packet collision is many times increased. Furthermore, as flow-level VLB cannot achieve load balance when the large and small flows are mixed together, an irrational routing may easily increase the block ratio and reduce the bandwidth utilization. With a fixed injection rate 0.164 Gbps/node, Fig. 7 shows the amount of packet collision times at each switch. Compared with packet-level VLB, the value of packet collision times increases twice in edge switches when flow-level VLB with 0% elephant flows is introduced. With more elephant flows injected, collision becomes more serious. Furthermore, as the traffic load is unevenly distributed, the number of collision times varies largely in different switches.

Fat tree based hybrid switching architecture effectively improves the network performance. Compared with traditional

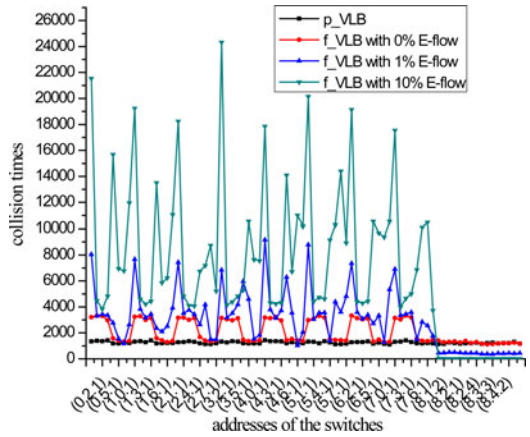


Fig. 7. The sum of packet collision times at each switch (the first byte of the ip address is omitted), p_VLB: packet level VLB; f_VLB: flow level VLB.

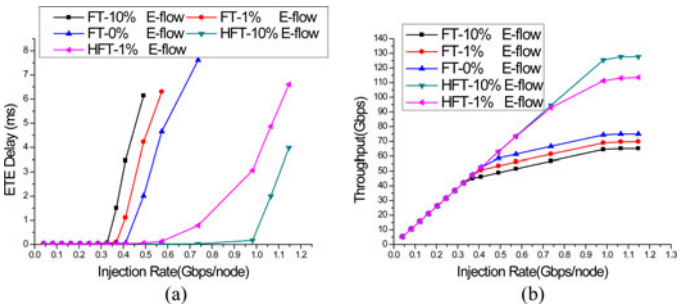


Fig. 8. (a) ETE delay and (b) throughput under stride(4) traffic pattern.

electronic fat tree with flow-level VLB, the saturation point and maximum network throughput increase about 1.96 times when 1% elephant flows are introduced. When 10% elephant flows are introduced, the two performance metrics are 3.35 times better than traditional fat tree. Three aspects contribute to this improvement. First, the large elephant flows are inherently suitable for OCS as the large volume of data chunks can effectively reduce the overhead for path setup. Second, the improved communication strategy achieves efficient utilization of all available resources. The multiple wavelengths and optical paths reduce the high blocking ratio of traditional OCS. Third, the MR based switching fabric enables fast reconfiguration behavior, which alleviates excess delay for optical path setup.

The network performance is also evaluated under other traffic patterns (as shown in Figs. 8–10). Three main observations are obtained from the simulation results. (1) The hybrid switching architecture can achieve better performance under various traffic patterns. The main reason is in fat tree topology the optical devices can be deeply embedded in each pod. Thus under most traffic patterns, the optical interconnection is effectively utilized to take the traffic burden. (2) Under local traffic patterns, as shown in Figs. 9 and 10, the hybrid switching architecture does not make significant improvements. It is reasonable because most local traffic flows are still processed by electrical switches. Although the *internal pod E-flows* are transmitted through optical aggregation switch, very limited paths and wavelengths are available in this case. (3) For traditional electronic fat tree, the

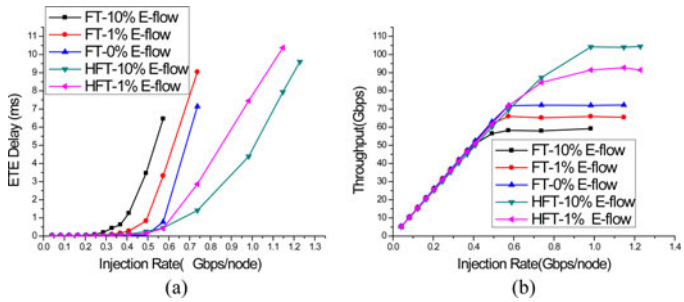


Fig. 9. (a) ETE delay and (b) throughput under staggered prob (50%,30%).

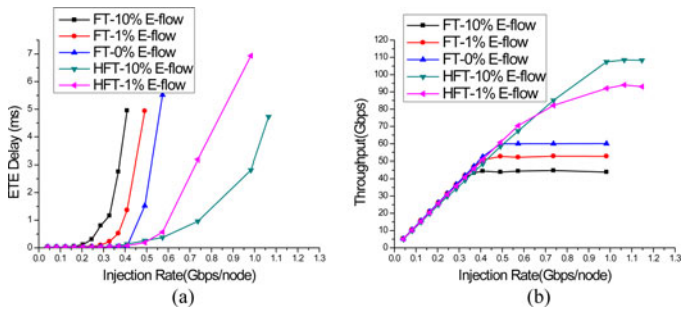


Fig. 10. (a) ETE delay and (b) throughput under staggered prob (30%,40%).

proportion of the elephant flows rather than the traffic patterns has significant influence on the network performance. This indicates that when multi-rooted trees face a mixture of elephant and mouse flows, the capacity for delivering elephant flows becomes the bottleneck of the network performance.

The control overhead can be evaluated in following three aspects: the length of the control loop, the bandwidth for delivering control messages, and the amount of control packets used for scheduling each elephant flow. As MR-based optical switch enables fast configuration time, thus in fat tree based hybrid switching architecture the length of the control loop is equal to the total transmission delay of three control packets. The size of all control packets is set to 64 bytes. For each packet, assume the average processing delay in each switch is about 10 $\mu$s and the transmission delay is around 0.512 $\mu$s/hop. It takes about 126.144 $\mu$s to set up an optical path, which is much less than Hedera and DARD. For Hedera, only the computation time in each control loop is estimated to be 100 ms [12].

As both Hedera and fat tree based hybrid switching architecture use out-of-band control system, the dedicated bandwidth for control can be calculated and compared. Hedera uses a single scheduler to communicate with all edge switches in each scheduling cycle. To maintain high quality of real-time control, higher link bandwidth, assume 10 Gbps, is needed to build the control channels. Thus the total control bandwidth of $k$-port Hedera is $10k^2$ Gbps, while that of fat tree based hybrid switching architecture is $mk^2$ Gbps. Generally, $m$ is less than 10, thus the control overhead of fat tree based hybrid switching architecture is less than that of Hedera.

As a centralized scheduling architecture, Hedera makes scheduling for multiple elephant flows concurrently. It is difficult to calculate the control cost for each flow. In DARD and

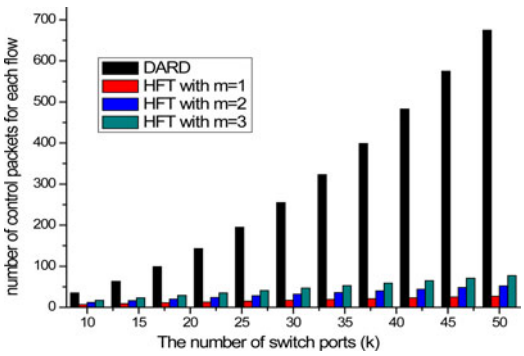


Fig. 11. The number of control packets sent for each flow, HFT: fat tree based hybrid switching architecture.

fat tree based hybrid architecture, the source node independently sends control packets for its own flows. The number of control packet sent for each flow can be estimated. For $k$-port fat tree topology, DARD needs to send $(k^2/4 + k)$ control packets for each flow while fat tree based hybrid architecture needs to send $(mk/2 + 2)$ control packets. The comparison shown in Fig. 11 indicates that fat tree based hybrid switching architecture largely reduces the control overhead.

## VI. Conclusion

Hybrid switching networks can effectively improve the network performance while maintain an incremental upgrade of an operating data center. To take more benefits from the optical interconnection, a distributed OCS model capable of building optical path over multiple hops is proposed. This new OCS model utilizes WDM technology and MR based fast switching fabric to increase the link utilization. It also enables building a multi-rooted tree based hybrid architecture which realizes flow-level optical circuit switching. The simulation results show the hybrid architecture achieves considerable network performance while maintains low control overhead.

## Acknowledgment

The authors are indebted to the associate editor and the anonymous reviews for their helpful comments on the earlier draft of this paper.

## References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM*, 2008, pp. 63–74.
[2] 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 *Proc. ACM SIGCOMM*, 2009, pp. 51–62.
[3] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM*, 2009, pp. 63–74.
[4] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: A scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 75–864, Aug. 2008.
[5] X. Ye, Y. Yin, S. J. B. Yoo, P. Mejia, R. Proietti, and V. Akella, "DOS: A scalable optical switch for datacenters," in *Proc.6th ACM/IEEE Symp. Archit. Netw. Commun. Syst.*, New York, NY, USA, 2010, pp. 1–12.
[6] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: A topology malleable data center network," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, New York, NY, USA, 2010, p. 8.
[7] P. N. Ji, Q. Dayou, K. Kanonakis, C. Kachris, and I. Tomkos, "Design and evaluation of a flexible-bandwidth OFDM-based intra-data center interconnect," *IEEE J. Sel. Topics Quantum Electron.*, vol. 19, no. 2, pp. 299–308, Mar./Apr. 2013.
[8] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1021–1036, Fourth Quarter 2012.
[9] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM*, New York, NY,, USA, 2010, pp. 339–350.
[10] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. Eugene Ng, M. Kozuch, and M. Ryan, "C-through: Part-time optics in data centers," in *Proc. ACM SIGCOMM* , New York, NY, USA, 2010, pp. 327–338.
[11] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2010, pp. 267–280.
[12] M. AI-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic how scheduling for data center networks," in *Proc. Netw. Syst. Design Implementation*, 2010, p. 19.
[13] X. Wu and X. Yang, "DARD: Distributed adaptive routing for datacenter networks," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst.*, Jun. 2012, pp. 32–41.
[14] A. S.-W. Tam, X. Kang, and H. J. Chao, "Leveraging performance of multiroot data center networks by reactive reroute," in *Proc. IEEE 18th Annu. Symp. High Perform. Interconnects*, Aug. 2010, pp. 66–74.
[15] N. Hamedazimi, H. Gupta, V. Sekar, and S. R. Das, "Patch panels in the sky: A case for free-space optics in data centers," in *Proc. 12th ACM Workshop Hot Topics Netw.*, 2013, p. 23.
[16] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, pp. 447–458, Oct. 2013.
[17] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying NOX to the datacenter," in *Proc. HotNets*, 2009.
[18] A. R. Curtis, K. Wonho, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *Proc. IEEE INFOCOM* , Apr. 2011, pp. 1629–1637.
[19] H. Wang and K. Bergman, "A bidirectional 2 × 2 photonic network building-block for high-performance data centers," in *Proc. Opt. Fiber Commun. Conf.*, 2011, pp. 1–3.
[20] A. Biberman, G. Hendry, J. Chan, H. Wang, K. Bergman, K. Preston, N. Sherwood-Droz, J. S. Levy, and M. Lipson, "CMOS-compatible scalable photonic switch architecture using 3D-integrated deposited silicon materials for high-performance data center networks," in *Proc. Opt. Fiber Commun. Conf.*, 2011, pp. 1–3.
[21] J. Chan, G. Hendry, K. Bergman, and L. P. Carloni, "Physical-layer modeling and system-level design of chip-scale photonic interconnection networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 10, pp. 1507–1520, Oct. 2011.

Authors' biographies not available at the time of publication.