Optical data center networks; architecture, performance, and energy-efficiency

Yuichi Ohsita, and Masayuki Murata
Osaka University, Japan

Abstract

In a data center, servers communicate with each other to store and handle a large amount of data. Thus, the network within a data center affects the performance of the data center; the lack of bandwidth between servers increases the time to obtain the required data from the other servers. The energy consumption is also one of the important problems in a data center. The energy consumed by the data center increases as the amount of data handled by the data center becomes large. The network within a data center consumes a non-negligible fraction of the energy consumed in a data center. Therefore, the data center network should provide the sufficiently large bandwidth between servers with a small energy consumption. One approach to achieve this is the *optical data center network*, which uses the optical switches, because optical switches provides a large bandwidth between their ports with a small energy consumption. There are two kinds of optical switches; *optical packet switches* and *optical circuit switches*. In this chapter, we explain the overview of the architectures of these optical switches. Then, we introduce two approaches of optical data center networks. The first approach is the application of the optical packet switches, and focuses on the bandwidth provided for all-to-all communication. In this approach, we introduce the network structure that uses the large bandwidth of optical packet switches. The other approach is the application of the optical circuit switches, whose energy consumption is much smaller than that of the packet switches, and aims to minimize the energy consumption. In this network, the optical circuit switches are used to construct the core of the data center network, and the packet switches are connected to the optical circuit switches. The connection between packet switches are changed by setting the optical circuit switches. By setting the optical circuit switch so as to minimize the number of required ports of the packet switches considering the current traffic, we reduce the energy consumption.

1 Intoroduction

In recent years, online services such as cloud computing have become popular, and the amount of data, required to be processed by such online services, is increasing. To handle such a large amount of data, large data centers with hundreds of thousands of servers have been built.

In a large data center, a large amount of data is stored in the memories or storages of a large number of servers by using distributed file systems such as Google File System [1]. Then, such a large amount of data is handled by using the distributed computing frameworks such as MapReduce [2]. The distributed file systems or distributed computing frameworks require the communication between servers within a data center. Thus, the data center network plays an important role in a data center, and affects the performance of the data center.

To avoid the network being a bottleneck of the data center, the data center network should provide communication with sufficiently high bandwidth between communicating server pairs. The lack of bandwidth between servers may prevent the communication between servers, and increases the time to obtain the required data. This degrades the performance of the data center. However, the traditional data center network, which is constructed as a tree topology, cannot provide communication with sufficiently large bandwidth between servers; in the traditional data center network, the root of tree topology becomes the bottleneck, and the number of hops between servers becomes large as the number of servers in a data center increases.

Another important problem in a data center is the energy consumption. As the amount of data handled by data centers becomes large, the energy consumption of the data centers increases. Energy consumption of the data center network occupies a non-negligible fraction of the total energy consumption in the data center [3], and becomes large as the size of the network increases. Thus, to reduce energy consumption of a large data center, the energy consumption of the data center network should be reduced.

There are many researches to construct a data center with sufficient performance or small energy consumption [4–12]. For example, Al-Fares et al. have proposed the topology called *FatTree* [4], that provides sufficient bandwidth between all server pairs. The FatTree is a tree with multiple root nodes. In this topology, each node uses a half of its ports to connect it to the nodes of the upper layer, and the other half of its ports to connect it to the nodes of the lower layer. Another topology to provide sufficient bandwidth has been proposed by Kim et al. [5]. This topology is called the *flattened butterfly*. The flattened butterfly provides enough bandwidth between servers, and makes the number of hops between servers small, by using nodes with a large number of ports instead of constructing the tree.

Though many data center network structures constructed of electronic switches have been proposed as described above, it is difficult to achieve both of the sufficient bandwidth and the small energy consumption by using only the electronic switches. The electronic switch with a large number of ports that provides communication with a large bandwidth consumes a large energy. Though the electronic switches with a small number of ports consume less energy than the switches with a large number of ports, we require a large number of switches to connect all servers in a large data center if we construct the data center network by using the switches with a small number of ports.

One approach to provide enough bandwidth with a small energy consumption is the *optical data center networks*, which uses optical switches. Optical switches consume much less energy than electronic switches, and provides communication with a large bandwidth between their ports. There are two kinds of optical switches; *optical packet switches* and *optical circuit switches*. Optical packet switches relay the packets constructed of optical signals without converting optical signals to electronic signals. The destination port at each optical packet switch is determined based on the labels of the packets. And multiple packets from the different input ports share the same output port of each optical packet switch by waiting the output port to become free at the buffer when the output port is busy. Optical circuit switches connect each input port with one of the output ports based on the configuration. Unlike the optical packet switches, the output ports of the optical circuit switch cannot be shared by multiple flows from the different input ports. However, the energy consumption of the optical circuit switch is much smaller than the optical packet switches, because the optical circuit switches do not require label processing and so on.

In this chapter, we introduce two approaches using the above optical switches. First approach aims to provide a large bandwidth between all servers. In this approach, we use the optical packet switches, because a large number of ports are required to provide communication between all servers at once if the packets from different input ports cannot share the same output port. The other approach aims to minimize the energy consumption. To minimize the energy consumption, the optical circuit switches are useful because their energy consumption is small. In this approach, optical circuit switches are deployed at the core of the data center, and packet switches are connected to the optical circuit switches. In this network, the connection of the packet switches can be changed by configuring the optical circuit switches. Thus, by configuring the connection between packet switches so as to satisfy the current requirements and shutting down the unused ports of packet switches, we provide a sufficient bandwidth with a small energy consumption.

The rest of this chapter is organized as follows. Section 2 explains the overview of the optical switch architectures. Section 3 introduces the approach to provide a large bandwidth to all-to-all communication by using optical packet switches. Section 4 introduces the approach to achieve the small energy consumption.

In Section 5 provides a conclusion.

2 Optical Switches Used in Optical Data Center Networks

In optical data center networks, two kinds of switches, optical packet switches and optical circuit switches are used. In this section, we introduce the overview of their architectures.

2.1 Optical Packet Switch

Optical packet switch relays optical packets constructed of optical signals without converting optical signals to electronic signals. Each optical packet includes a label indicating the destination. An optical packet switch receiving an optical packet relays the packet to the output port based on the label.

Figure 4 shows the model of an optical packet switch. As shown in this figure, an optical packet switch is constructed of label processors, controllers, switching fabrics and buffers. In an optical packet switch, the label processors identify the labels of the optical packets. Then, based on the label of the incoming optical packet, the controller determine the destination port of the optical packet, and configure the switching fabric. After the configuration of the switching fabric, the incoming optical packet is relayed to the output port.

In the packet switch, multiple packets to the same output port may arrive at the same time. To avoid packet loss, buffers are deployed in the optical packet switch.

The buffers may be constructed by the fiber delay lines (FDLs) or the electronic memories. In the case of the FDL-based buffer, the optical packets can be stored without converting into electronic packets. However, it is difficult to construct a large buffer. On the other hand, in the case of the electronic buffer, the large buffers can be easily implemented, though the optical packets must be converted into electronic packets before storing the packets.

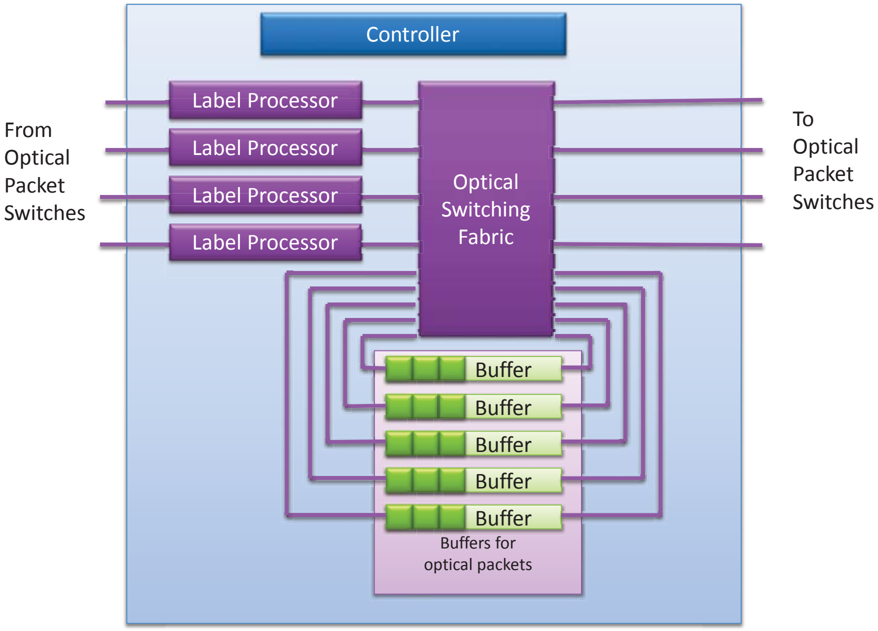


Figure 1: The model of the optical packet switch

The switching fabric relays the incoming optical packets to the desired destination ports without converting optical signals into electronic signals. The switching fabric can be constructed by the arrayed waveguide grating router (AWGR) [13] or the broadcast and select switch (B&S) [14]. The AWGR is a passive switching fabric where the output port of the input signal depends on the wavelength of the input signal. Thus, in the switching fabric constructed by the AWGR, the packets are relayed to the destination port by changing the wavelength of the input signal according to the desired output port. To change the wavelength, the wavelength converters or the tunable lasers are deployed at all input ports of the AWGR. The B&S is based on the WDM star coupler. In the B&S, the input signals are broadcast through a splitter to all of output ports. Then, by setting each output port to select the signal corresponding to the port, the optical signals are relayed to their output ports. Both types of the switching fabrics can change the destination port immediately by setting the wavelength of input signal or setting the selector. Thus, we can change the configuration of the switching fabric each time a packet arrives.

Optical packet switch provides a large bandwidth between its port with a small energy consumption, compared with the electronic packet switch, because it relays the optical packet without converting optical signals into electronic signals. Thus, one approach to provide a large bandwidth with a small energy consumption is to use the optical packet switches.

2.2 Optical Circuit Switch

Optical circuit switch is a switch where the input optical signals are relayed to the output ports based on the configuration.

One of the most popular optical circuit switch is the Micro-Electronic-Mechanical Systems (MEMS) based optical circuit switch shown in Figure 2 [15]. In the MEMS based optical circuit switch, the micro mirrors are deployed. The input optical signals are reflected by the micro mirrors to the output ports. Each mirrors is attached to tiny motors, and the angles of the mirrors can be changed. By changing the angles of the micro mirrors, the output port of the input signal can be changed.

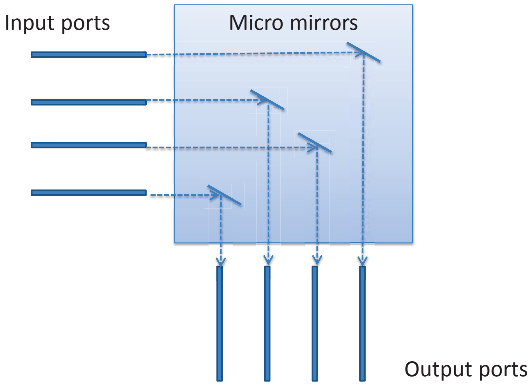


Figure 2: MEMS Optical Circuit Switch

The MEMS optical circuit switch can be configured by the commands from the remote node. The configuration commands indicate the output port corresponding to each input port. The controller within the MEMS optical switch controls the tiny motors to set the angles of the micro mirrors so that the input signals are reflected to the corresponding output ports.

The optical circuit switch consumes only little energy, because it only reflect the optical signals to their output ports by the micro mirrors. However, the change of the angles of the mirrors takes a time. Thus, the MEMS based optical circuit switch cannot be used as a switching fabric of the optical packet switch.

One of the important application of the optical circuit switch is to construct the virtual network. In this approach, the core network is constructed of optical circuit switches. Then, packet switches are connected to the ports of the optical circuit switches. By configuring the optical circuit switches, the lightpaths called *optical paths* are established between the packet switches. The set of the optical paths and packet switches forms the virtual network. The virtual network can be changed by reconfiguring the optical circuit switches based on the current traffic and so on.

3 Approach 1: Optical Data Center Network to Provide a Large Bandwidth for All-to-All Communication

In a data center handling a large amount of data, a large number of servers cooperate with each other. To enable the cooperation between any server pairs, the data center network should accommodate all-to-all communication. In addition, the lack of the bandwidth between communication may increase the time to obtain the required data from the other servers. Thus, the data center network should provide a large bandwidth communication between all server pairs.

All-to-all communication requires packet switches, because circuit switch network cannot accommodate all-to-all communications since multiple flows from different input ports cannot share the same output port of the circuit switch. In this section, we discuss the network structure that uses the optical packet switches to provide a large bandwidth between all server pairs based on our research [16].

Optical packet switches provide a large bandwidth with a small energy consumption. Several optical packet switch architectures for data centers have been proposed [17–19]. Some of them are the optical packet switch with a large number of ports [18, 19]. However, the network using the optical packet switches with a large number of ports is vulnerable to the failure of the optical packet switch, because most of the traffic between servers traverses the optical packet switch.

In this section, we introduce the network structure using the optical packet switches with a small number of ports that can provide enough bandwidths between all server pairs even when failures occur. In this network structure, the optical packet switches are used to construct the core network of the data center. Figure 3 shows the image of the data center network using optical packet switches. In this network structure, optical packet switches are used to construct the core network so as to use the large bandwidth of the optical packet switches.

Similar to the traditional data center, we deploy the top of rack (ToR) switch in each server rack. All servers in a server rack are connected to one ToR switch. The ToR switches are connected to the core network by connecting them to optical packet switches. Each optical packet switch is connected to multiple ToR switches, and aggregates traffic from them to efficiently use the large bandwidth between optical packet switches. Each ToR switch is also connected to multiple optical packet switches to keep the connectivity even when optical packet switches fail.

In this network, packets from a server rack are converted into optical packets at the first optical packet switch connected to the source server rack. Then, the optical packets are relayed in the core network constructed of optical packet switches. Finally, the optical packet is converted into the electronic packets at the optical packet switch connected to the destination server rack, and is relayed to the destination server rack. In this network, each ToR switch relays only the electronic packets from or to the correspondent server rack, and does not relay the packets from or to the other server racks.

The details of the network structure suitable to the data center network using optical packet switches are discussed in the rest of this section.

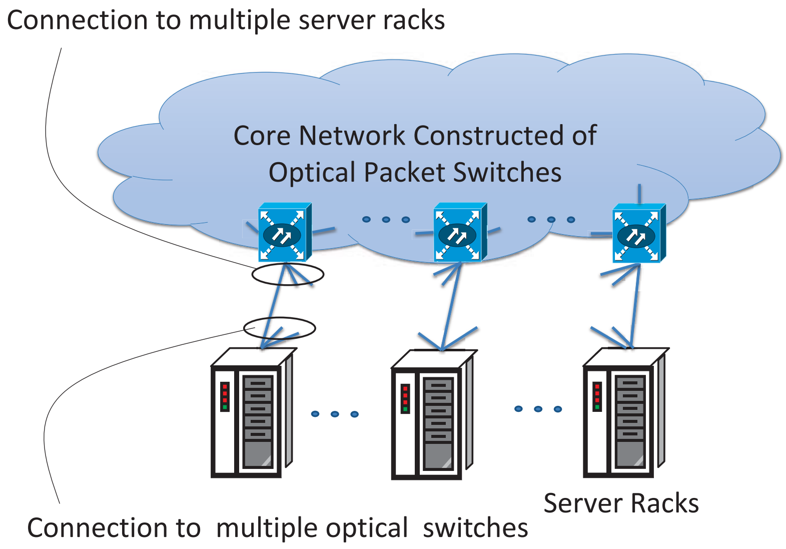

Connection to multiple server racks
Core Network Constructed of Optical Packet Switches
Server Racks
Connection to multiple optical switches

Figure 3: Data Center Network Using Optical Packet Switches

3.1 Optical Packet Switches with a Large Bandwidth

One approach to provide a large bandwidth between server racks is to use the optical packet switches with a large bandwidth. In this subsection, we introduce an optical packet switch architecture that provides a large bandwidth between its ports, which is used as an example of optical packet switch in this section.

In an optical network, a large bandwidth is provided by using multiple wavelength. Using multi-wavelength packets is one of the approaches using multiple wavelength. Multi-wavelength packets are constructed by dividing a packet into multiple wavelength signals. By using multiple wavelengths, we can provide a large bandwidth for each port.

Urata et al. [17] proposed and implemented an optical packet switch based on the multi-wavelength packet technology. Figure 4 shows the optical packet switch architecture. In this architecture, optical packets, constructed of multiple wavelengths, are relayed between optical packet switches. The optical packets from other optical packet switches are demultiplexed into optical signals of each wavelength. Then, after label processing, the optical signals are relayed to the destination port and multiplexed into optical packets. In case of collision, the optical packets are stored in the shared buffer constructed with CMOS after serial-to-parallel conversion. Then, we try to relay the packets again after parallel-to-serial conversion.

This optical packet switch also has electronic ports. In the data center, the electronic ports can be used to connect them to the ToR switches. Packets from ToR switches are aggregated to the optical packets and stored in the shared buffer. Then, the packets are relayed after parallel-to-serial conversion. Optical packets whose destination is the ToR switches connected to the optical packet switch are also stored in the shared buffer. Then the packets are sent to the destination ToR switches after demultiplexing the optical packet into the packets to each ToR switch.

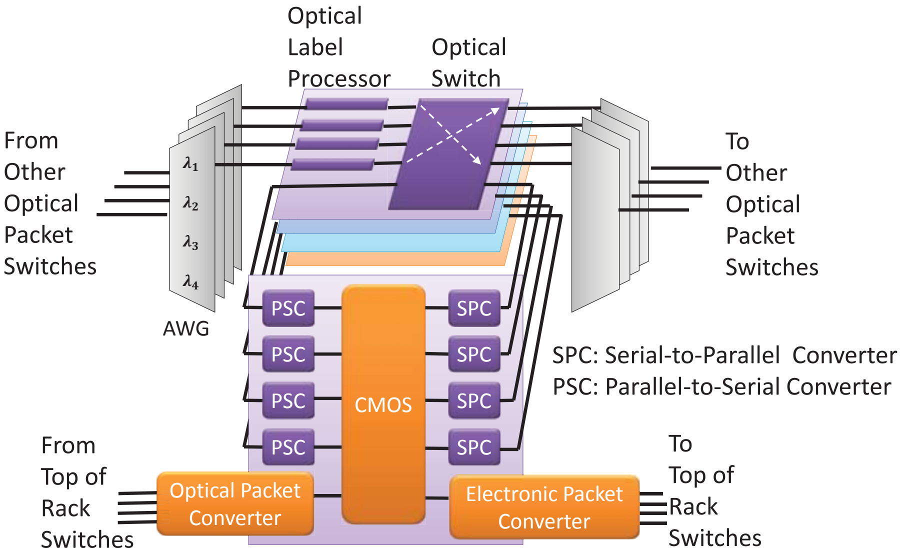

Optical Label Processor
Optical Switch
From Other Optical Packet Switches
λ_1
λ_2
λ_3
λ_4
AWG
To Other Optical Packet Switches
PSC
PSC
PSC
PSC
CMOS
SPC
SPC
SPC
SPC
SPC: Serial-to-Parallel Converter
PSC: Parallel-to-Serial Converter
From Top of Rack Switches
Optical Packet Converter
Electronic Packet Converter
To Top of Rack Switches

Figure 4: Opt-Electronic Packet Switch

3.2 Data Center Network Structure using Optical Packet Switches

In this section, we introduce a network structure satisfying the following points; (1) we efficiently use the links between optical packet switches, whose bandwidths are much larger than those of ports of ToR switches, by aggregating traffic from multiple ToR switches, and (2) we keep the connectivity between all servers even when optical packet switches fail by connecting each ToR switch to multiple optical packet switches.

In our topology, we divide the data center network into multiple groups. By connecting each ToR switch to optical packet switches belonging to the same group, we avoid long links between optical packet switches and ToR switches. We denote the number of ToR switches in each group, the number of optical packet switches in each group, and number of groups as $N_{\text{in}}^{\text{tor}}$, $N_{\text{in}}^{\text{opt}}$, and G respectively. Each optical packet uses P_{in} ports to connect optical packet switches belonging to the same group, and P_{gr} ports to connect optical packet switches belonging to other groups. We also denote the number of servers connected to each ToR switch as $P_{\text{tor}}^{\text{svr}}$. The number of the ToR switches connected to each optical packet switch is denoted as $P_{\text{tor}}^{\text{opt}}$, and the number of optical packet switches connected to each ToR switch is denoted as $P_{\text{opt}}^{\text{tor}}$.

We also divide each group into $P_{\text{opt}}^{\text{tor}}$ subgroups. Each ToR switch is connected to optical packet switches belonging to different subgroups. All of P_{in} ports of each optical packet switch are used to connect optical packet switches belonging to the same subgroup. No links are constructed between optical packet switches belonging to different subgroups as shown in Figure 5.

In this topology, we have $P_{\text{opt}}^{\text{tor}}$ distinct paths between all ToR switch pairs. Thus, we can keep the connectivity between all ToR switch pairs even when optical packet switches fail.

In addition, this topology effectively uses the ports of optical packet switches. The set of ToR switches connected to each subgroup is the same. Thus, the links between optical packet switches belonging to different subgroups are not required. By using all of P_{in} ports of each switch to connect optical packet switches of the same subgroup, we make the number of hops between ToR switches and optical packet switches small.

We assign the unique ID to the groups, the subgroups in each group and the optical packet switches in each subgroup. We denote the group ID, the subgroup ID and the optical packet switch ID of optical packet switch s as $D^{\mathrm{gr}}(s)$, $D^{\mathrm{sub}}(s)$ and $D^{\mathrm{opt}}(s)$ respectively.

The rest of this subsection, we explain the details of the connection within a group and between the groups.

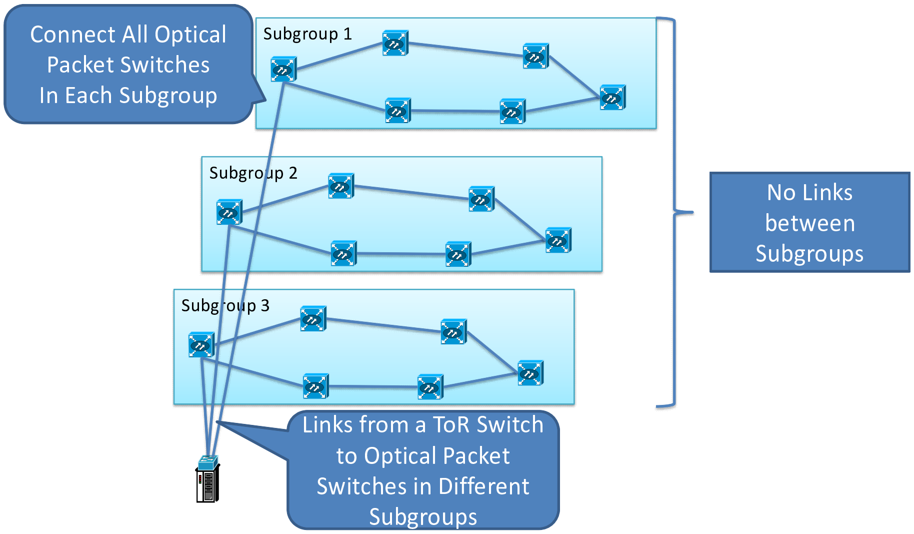

Connect All Optical Packet Switches In Each Subgroup
Subgroup 1
Subgroup 2
Subgroup 3
No Links between Subgroups
Links from a ToR Switch to Optical Packet Switches in Different Subgroups

Figure 5: Connection withing a Group

3.2.1 Connection within a Group

We first connect the optical packet switches belonging to the same subgroups. Then, we connect each ToR switch to $P_{\mathrm{opt}}^{\mathrm{tor}}$ optical packet switches belonging to different subgroups.

The optical packet switches belonging to the same subgroup are connected by the following steps. First, we construct a ring topology by connecting the optical packet switches of the nearest optical packet switch IDs. Then, we add links between optical packet switches S_1 and S_2 if the following constraint is satisfied;

$$D^{\mathrm{opt}}(S_2) = \lfloor D^{\mathrm{opt}}(S_1) + iN_{\mathrm{sub}}/(P_{\mathrm{in}} - 1) \rfloor \bmod N_{\mathrm{sub}}, \tag{1}$$

where N_{sub} is the number of optical packet switches belonging to each subgroup and i is a positive integer. If the optical packet switch S_2 satisfying Eq. (1) does not have enough ports used for the connection within a group, we connect S_1 to the optical packet switch that has enough ports and has the optical packet switch ID close to S_2.

3.2.2 Connection between Groups

We connect groups by adding links between optical packet switches belonging to different groups. The number of links used to connect a group to other groups is $N_{\mathrm{in}}^{\mathrm{opt}} P_{\mathrm{gr}}$. If $N_{\mathrm{in}}^{\mathrm{opt}} P_{\mathrm{gr}} \geq G - 1$, we can add links between all group pairs. In this section, we assume that we can add links between all group pairs.

To connect groups, we select the optical packet switches on the both ends of links between the groups. We select the optical packet switch S_1 as the optical packet switch to be connected to the Kth link between groups $D^{\mathrm{gr}}(S_1)$ and $D^{\mathrm{gr}}(S_2)$ if the following constraint is satisfied;

$$D^{\mathrm{in}}(S_1) = \begin{cases} \left\lfloor \frac{D^{\mathrm{gr}}(S_2) + K(G-1)}{P_{\mathrm{gr}}} \right\rfloor & (D^{\mathrm{gr}}(S_1) \geq D^{\mathrm{gr}}(S_2)) \\ \left\lfloor \frac{D^{\mathrm{gr}}(S_2) + K(G-1) - 1}{P_{\mathrm{gr}}} \right\rfloor & (\text{Otherwise}) \end{cases}, \tag{2}$$

where D^{in} is the number defined by

$$D^{\mathrm{in}}(S_1) = D^{\mathrm{sub}}(S_1)\frac{N_{\mathrm{in}}^{\mathrm{opt}}}{P_{\mathrm{opt}}^{\mathrm{tor}}} + D^{\mathrm{opt}}(S_1).$$

3.2.3 Routing in the Topology

In our topology, we can calculate routes from ToR switches to optical packet switches and from optical packet switches to ToR switches by using the ID assigned to optical packet switches, $[D^{\mathrm{gr}}(s), D^{\mathrm{sub}}(s), D^{\mathrm{opt}}(s)]$, without exchanging any route information.

Routes from ToR Switches to Optical Packet Switches The routes from ToR switch in the group $D^{\mathrm{gr}}(s)$ to the optical packet switch d are calculated by the following steps.

If the destination optical packet switch d belongs to $D^{\mathrm{gr}}(s)$, the source ToR switch first sends the packet to the optical packet switch that is directly connected to the source ToR switch and belongs to the same subgroup as the destination optical packet switch d, (i.e., the subgroup $D^{\mathrm{sub}}(d)$). The intermediate optical packet switch selects the next hop by calculating $H(d, a)$, defined by Eq. (3), for all neighbor optical packet switches a.

$$H(d, a) = |D^{\mathrm{opt}}(d) - D^{\mathrm{opt}}(a)| \tag{3}$$

The optical packet switch a having the smallest $H(d, a)$ is close to the destination optical packet switch d. Thus, we select the optical packet switch having the smallest $H(d, a)$ as the next-hop optical packet switch. If there are multiple optical packet switches having the smallest $H(d, a)$, we regard all optical packet switches having the smallest $H(d, a)$ as the candidates of the next hop, and balance the load by selecting the next-hop optical packet switch randomly from the candidates.

If the destination optical packet switch does not belong to $D^{\mathrm{gr}}(s)$, we first select the intermediate optical packet switch in the group $D^{\mathrm{gr}}(s)$ having a link to an optical packet switch belonging to the subgroup $D^{\mathrm{sub}}(d)$ in the group $D^{\mathrm{gr}}(d)$. The intermediate optical packet switch is selected by the following steps. First we calculate the range of $\tilde{k}$ in Eq. (2) where $\tilde{k}$th link between the groups $D^{\mathrm{gr}}(s)$ and $D^{\mathrm{gr}}(d)$ are connected to optical packet switches belonging to the subgroup $D^{\mathrm{sub}}(d)$ in the group $D^{\mathrm{gr}}(d)$ by solving the following inequation;

$$D^{\mathrm{sub}}(d)\frac{N_{\mathrm{in}}^{\mathrm{opt}}}{P_{\mathrm{opt}}^{\mathrm{tor}}} \leq \tilde{D}^{\mathrm{in}}(d) < (D^{\mathrm{sub}}(d)+1)\frac{N_{\mathrm{in}}^{\mathrm{opt}}}{P_{\mathrm{opt}}^{\mathrm{tor}}}, \tag{4}$$

where

$$\tilde{D}^{\mathrm{in}}(d) = \begin{cases} \lfloor \frac{D^{\mathrm{gr}}(s)+\tilde{k}(G-1)}{P_{\mathrm{gr}}} \rfloor & (D^{\mathrm{gr}}(d) \geq D^{\mathrm{gr}}(s)) \\ \lfloor \frac{D^{\mathrm{gr}}(s)+\tilde{k}(G-1)-1}{P_{\mathrm{gr}}} \rfloor & \text{(Otherwise)} \end{cases}.$$

Then, we identify the optical packet switch s' connected to an optical packet switch belonging to the subgroup $D^{\mathrm{sub}}(d)$ in the group $D^{\mathrm{gr}}(d)$, by substituting $\tilde{k}$ to K, s' to S_1, and d to S_2 in Eq. (2).

After selecting the intermediate optical packet switch, we calculate the routes from the source ToR switch to the intermediate optical packet switch and from the intermediate optical packet switch to the destination optical packet switch by the same steps as the case that the destination optical packet switch d belongs to $D^{\mathrm{gr}}(s)$.

Routes from optical packet switches to ToR switches If the destination ToR switch belongs to the same group as the source optical packet switch, we first select the intermediate optical packet switch d^{opt} that belongs to the same subgroup as the source optical packet switches and is directly connected to the destination ToR switch. In this section, we assume that each optical packet switch knows the connections between all optical packet switches and all ToR switches within its group. Thus, each optical packet switch can calculate d^{opt}. Then, we calculate the routes from the source optical packet switch to the intermediate optical packet switch d^{opt} by using $H(d^{\mathrm{opt}}, a)$ in the same manner as the case of routes from the ToR switch to the optical packet switch.

If the destination ToR switch does not belong to the same group as the source optical packet switch, we select the intermediate optical packet switch having a link to the group of the destination ToR switch. The intermediate optical packet switches having a link to the group of the destination ToR switch are obtained by Eq. (2). Then, we calculate the routes from the source optical packet switch to the intermediate optical packet switch, and from the intermediate optical packet switch to the destination ToR switch, by the same steps as the case that the destination ToR switch belongs to the same group as the source optical packet switch.

Routes between ToR Switches We can calculate routes between ToR switches by selecting an intermediate optical packet switch and calculating the routes from the source ToR switch to the intermediate optical packet switch and from the intermediate optical packet switch to the destination ToR switch. By selecting the intermediate optical packet switch at an end of a link between the group of the source ToR switch and the group of the destination ToR switch based on Eq. (2), we can avoid large hop counts between ToR switches.

Handling Failures If the optical packet switch S_1 cannot find no suitable next-hop optical packet switch for the destination d because of failures, it returns the packet to the previous-hop optical packet switch S_2. By receiving the returned packet, S_2 identifies that S_1 has no suitable path to the destination d. Thus, S_2 removes S_1 from the candidates of the next-hop optical packet switches to d, and relays the packet to one of the other candidates. If S_2 cannot also find no suitable next-hop optical packet switch after removing S_1 from the candidates, S_2 also returns the packet to the previous hop of S_2. By continuing the above steps, all optical packet switches can remove the switches having no suitable routes to d from their candidates of the next-hop switch to d.

3.3 Parameter Settings

Our topology has three kinds of parameters, P_{gr}, P_{in} and connection between the ToR switches and the optical packet switches. In this subsection, we set these parameters so that our topology can accommodate any traffic without limiting the bandwidth between servers.

In this section, we assume each server has a link with 1 Gbps connected to a ToR switch. The bandwidth of each link between optical packet switches is B^{opt} Gbps.

When setting parameters, we assume that traffic is balanced by Valiant Load Balancing (VLB) [20]. In the VLB, we select the intermediate nodes randomly regardless of the destination to avoid the concentration of traffic on certain links even when traffic volume of certain node pairs is large.

Applying the VLB to this topology, we select an intermediate optical packet switch randomly with the probability of $\frac{1}{N_{\mathrm{in}}^{\mathrm{opt}} G}$. Then, traffic is sent via the selected intermediate optical packet switch. Applying the VLB, the traffic volume from a ToR switch to an optical packet switch, $T_{\mathrm{tor,opt}}$ and the traffic volume from an optical packet switch to a ToR switch, $T_{\mathrm{opt,tor}}$ satisfy the following conditions;

$$T_{\mathrm{tor,opt}} \leq \frac{P_{\mathrm{tor}}^{\mathrm{svr}}}{N_{\mathrm{in}}^{\mathrm{opt}} G}, \tag{5}$$

$$T_{\mathrm{opt,tor}} \leq \frac{P_{\mathrm{tor}}^{\mathrm{svr}}}{N_{\mathrm{in}}^{\mathrm{opt}} G}. \tag{6}$$

Thus, we set the parameters of our topology so as to accommodate traffic of $T^{\mathrm{max}}_{\mathrm{tor,opt}} = T^{\mathrm{max}}_{\mathrm{opt,tor}} = \frac{P^{\mathrm{svr}}_{\mathrm{tor}}}{N^{\mathrm{opt}}_{\mathrm{in}} G}$ between all ToR switch and optical packet switch pairs.

3.3.1 Parameter of Connection Between Groups

By applying the VLB, the sum of traffic sent between a certain group pair T^{gr} is constrained by

$$T^{\mathrm{gr}} \leq (T^{\mathrm{max}}_{\mathrm{tor,opt}} + T^{\mathrm{max}}_{\mathrm{opt,tor}}) N^{\mathrm{opt}}_{\mathrm{in}} N^{\mathrm{tor}}_{\mathrm{in}}.$$

We have $\frac{P_{\mathrm{gr}} N^{\mathrm{opt}}_{\mathrm{in}}}{G-1}$ bidirectional links between each group pair whose bandwidths are B^{opt} Gbps. Thus, to avoid congestion on the links between groups, we set P_{gr} so as to satisfy the following condition;

$$\frac{2B^{\mathrm{opt}} P_{\mathrm{gr}} N^{\mathrm{opt}}_{\mathrm{in}}}{G-1} \geq (T^{\mathrm{max}}_{\mathrm{tor,opt}} + T^{\mathrm{max}}_{\mathrm{opt,tor}}) N^{\mathrm{opt}}_{\mathrm{in}} N^{\mathrm{tor}}_{\mathrm{in}}. \tag{7}$$

3.3.2 Parameters of Connection within a Group

We denote the traffic amount on link l as X_l and the set of links between optical packet switches within a group as L. We also denote the set of traffic from a ToR switch to an optical packet switch as $F^{\mathrm{tor}}_{\mathrm{opt}}$, and the set of traffic from an optical packet switch to a ToR switch as $F^{\mathrm{opt}}_{\mathrm{tor}}$.

The sum of the traffic amounts traversing the links within a certain group $\sum_{l \in L} X_l$ satisfies the following condition;

$$\sum_{l \in L} X_l \leq \sum_{i \in F^{\mathrm{tor}}_{\mathrm{opt}}} M_i T^{\mathrm{max}}_{\mathrm{tor,opt}} + \sum_{i \in F^{\mathrm{opt}}_{\mathrm{tor}}} M_i T^{\mathrm{max}}_{\mathrm{opt,tor}},$$

where M_i is the number of links within the group passed by traffic i.

We have $\frac{P_{\mathrm{in}} N^{\mathrm{opt}}_{\mathrm{in}}}{2}$ bidirectional links between optical packet switches within a group. Thus the sum of the bandwidth of the links within a group is $B^{\mathrm{opt}} P_{\mathrm{in}} N^{\mathrm{opt}}_{\mathrm{in}}$. Therefore, Eq. (8) should be satisfied to provide enough bandwidth between all ToR switches.

$$B^{\mathrm{opt}} P_{\mathrm{in}} N^{\mathrm{opt}}_{\mathrm{in}} \geq \sum_{i \in F^{\mathrm{tor}}_{\mathrm{opt}}} M_i T^{\mathrm{max}}_{\mathrm{tor,opt}} + \sum_{i \in F^{\mathrm{opt}}_{\mathrm{tor}}} M_i T^{\mathrm{max}}_{\mathrm{opt,tor}} \tag{8}$$

Eq. (8) indicates that one approach to provide enough bandwidth between ToR switches is to reduce the average number of hops between ToR switches and optical packet switches. Thus, we connect ToR switches to optical packet switches so as to minimize the average number of hops between the ToR switches and the optical packet switches. Then, we check whether the condition of Eq. (8) is satisfied. If the condition of Eq. (8) is not satisfied, we add more links between optical packet switches within the group.

We set the parameter P_{in} and connection between the ToR switches and the optical packet switches by the following steps.

Step 1 Initialize P_{in} to 2.

Step 2 Construct the topology between optical packet switches including both intra- and inter-group connection based on the current parameter, P_{in}.

Step 3 Connect ToR switches to optical packet switches so that the average number of hops between ToR switches and optical packet switches is minimized.

Step 4 Check whether Eq. (8) is satisfied for all groups. If Eq. (8) is satisfied, go to Step 5. Otherwise, go back to Step 2 after incrementing P_{in} by 1.

Step 5 End.

At Step 2 mentioned above, it is required to minimize the average number of hops between ToR switches and optical packet switches. However, it is difficult to obtain the optimal connection between ToR switches and optical packet switches among all possible solutions. In this section, we select one optical packet switch to be connected to a certain ToR switch so as to minimize the average number of hops from the ToR switch to all optical packet switches at each step, instead of finding the optimal solution among all possible solutions. By continuing this step, we connect all ToR switches to optical packet switches.

3.4 Evaluation

3.4.1 Topologies

In this subsection, we evaluate our topology by comparing it with the topologies shown in Table 1.

Table 1: Topologies Used in Our Evaluation

	# of Servers	# of Optical Packet SW	# of Links between Optical SW
Our Topology	2400	24	48
Full Torus	2400	24	48
Parallel Torus	2400	24	48
FatTree (3 layer)	2400	20	32
FatTree (4 layer)	2400	56	140
Switch-based DCell	2400	30	60

Our Topology In our evaluation, we set the number of optical packet switches connected to one ToR switch, $P^{\mathrm{tor}}_{\mathrm{opt}}$ to 2, and the number of ToR switches connected to one optical packet switch, $P^{\mathrm{opt}}_{\mathrm{tor}}$ to 10. Each ToR switch is connected to 20 servers within a rack. We set the number of optical packet switch within a group $N^{\mathrm{opt}}_{\mathrm{in}}$ to 6, and the number of groups G to 4. Thus, the number of optical packet switches in our topology is 24. We set the parameters P_{group} and P_{in} by the steps described in Section 3.3, setting B^{opt} to 100 Gbps. As a result, P_{group} and P_{in} are set to two.

Full Torus We construct the torus topology using the same number of optical packet switches and the same number of links as our topology. In this evaluation, each optical packet switch of our topology has four ports. Thus, in the full torus topology, we also use the optical packet switches with four ports, and we connect optical packet switches as the 4×6 torus. Similar to our topology, we connect each ToR switch to two optical packet switches and each optical packet switch to ten ToR switches.

Parallel Torus We construct $P^{\mathrm{tor}}_{\mathrm{opt}}$ torus topologies without links between the different torus topologies. We connect each ToR switch to optical packet switches in the different torus topologies. We use the same number of optical packet switches and the same number of links as our topology. That is, in this evaluation, we use 24 optical packet switches with four ports, and construct two 3×4 torus topologies. Similar to our topology, we connect each ToR switch to two optical packet switches and each optical packet switch to ten ToR switches.

FatTree We construct the FatTree topology using optical packet switches with four ports by method proposed by Al-Fares et al. [4]. This topology is the tree topology with multiple roots, where the half of the ports of an optical packet switch are used to connect it to nodes of the upper layer and the other half of the ports of an optical packet switch are used to connect it to nodes of the lower layer.

Though the method proposed by Al-Fares et al. [4] constructs the 3-layer FatTree, which is constructed of root switches and the pods containing two layers of switches, we can construct higher-layer FatTree topologies. The k-layer FatTree constructed of optical packet switches with four ports includes $(2k-1)2^{k-1}$ optical packet switches.

For our evaluation, we construct two kinds of the FatTree topologies; the 3-layer FatTree topology and the 4-layer FatTree topology using optical packet switches with four ports. We connect ToR switches to the optical packet switches at the lowest layer only. We connect the same number of ToR switches as our topology to both topologies. We set the number of optical packet switches connected to each ToR switch to 2. The number of ToR switches connected to each optical packet switch is 30 and 15 for the 3-layer and 4-layer FatTree topologies, respectively.

Switch-based DCell DCell is the topology for data center networks proposed by Guo et al. [6]. Since the original DCell is constructed by connecting server ports directly, we modify the DCell so as to be used for the connection between optical packet switches. We call the modified version of the DCell *switch-based DCell*.

In the switch-based DCell, a high-layer DCell is constructed from low-layer DCells. We denote the number of optical packet switches in one layer-k DCell as N_k^{DCell}. The switch-based DCell is constructed by the following steps. First, layer-0 DCells are constructed by adding links between all pairs of N_0^{DCell} optical packet switches. Then, layer-k DCells are constructed from $N_{k-1}^{\mathrm{DCell}}+1$ layer-$k-1$ DCells so that each layer-$k-1$ DCell is connected to all other layer-$k-1$ DCells with one link.

In our evaluation, we construct the layer-1 switch-based DCell with $N_0^{\mathrm{DCell}}=5$. Thus, the number of optical packet switches is 30 and the number of ports per optical packet switch is 5, which are larger than our topology. We connect the same number of ToR switches as our topology and set the number of optical packet switches connected to each ToR switch to 2. Thus, the number of ToR switches connected to one optical packet switch is 8. Comparing our topology with this topology, we clarify that our topology can accommodate more traffic than the switch-based DCell even though the switch-based DCell has more links.

3.4.2 Properties of Topologies

We compare the topologies by the following metrics.

Edge Betweenness The edge betweenness of the link l, C_l is defined by

$$C_l = \sum_{s,d \in V, l \in L} \frac{|F_{s,l,d}|}{|F_{s,d}|},$$

where V is the set of nodes which are the source or destination nodes of traffic, L is the set of links, $F_{s,l,d}$ is the set of the shortest paths from nodes s to d passing the link l, and $F_{s,d}$ is the set of the shortest paths from nodes s to d. The edge betweenness indicates the expected number of traffic passing the link. Thus, the topology having the large edge betweenness is easy to be congested. In our evaluation, we calculate the maximum edge betweenness for the traffic between ToR switches.

Minimum Cut The minimum cut indicates the smallest number of link failures to make the source node unable to reach the destination node. In our evaluation, we calculate the minimum cut for all ToR switch pairs. In all topologies used in our evaluation, each ToR switch is connected to two optical packet switches. Thus, the minimal cut is at most 2.

Table 2 shows the results. From this table, the minimum cuts of all topologies are 2. That is, all server pairs can communicate with each other even when one link fails in all topologies.

The FatTree topologies have large edge betweenness regardless of the number of layers. Especially, even though the 4-layer FatTree uses more than double optical packet switches and links between optical packet switches compared with other topologies, its edge betweenness is larger than our topology and the parallel torus, and is similar to the full torus. This is caused by the large average number of hops between ToR switches. In the FatTree topologies, a large amount of traffic passes the root optical packet switches, which causes the large average number of hops. The large average number of hops leads to the large expected number of traffic passing a link.

The switch-based DCell also has large edge betweenness, even though the switch-based DCell used in our evaluation has more links than our topology and torus topologies. This is because the switch-based DCell has only one link between each layer-0 DCell pair. In the switch-based DCell, we connect many layer-0 DCell pairs by limiting the number of links between each layer-0 DCell pair to one. This makes the number of hops between optical packet switches small. One link between each layer-0 DCell pair, however, cannot provide enough bandwidth.

Compared with the full torus, the parallel torus has smaller edge betweenness. This is caused by close connection between optical packet switches connected to different ToR switches. The parallel torus has more links between optical packet switches connected to different ToR switches instead of connecting optical packet switches connected to the same ToR switch, while the full torus has links between optical packet switches connected to the same ToR switch. This close connection between optical packet switches connected to different ToR switches makes the number of links passed by the traffic between ToR switches small, and reduces the number of traffic between ToR switches passing each link.

Among the topology used in our evaluation, our topology has the smallest edge betweenness. Similar to the parallel torus, our topology uses more links between optical packet switches connected to different ToR switches instead of connecting optical packet switches connected to the same ToR switches. In addition, the parameters in our topology are set by the steps described in Section 3.3, which aims to avoid concentration of traffic on certain links. As a result, the parameters of our topology are set so as to make the maximum edge betweenness small.

Table 2: Properties of Topologies

	Edge Betweenness	Minimum Cut
Our Topology	1000	2
Full Torus	1600	2
Parallel Torus	1200	2
FatTree (3 layer)	2700	2
FatTree (4 layer)	1575	2
Switch-based DCell	2065	2

We also compare the maximum edge betweenness when the randomly selected optical packet switches fail. In this comparison, we generate 100 patterns of random failures, and calculate the average of the maximum edge betweenness for the cases that all servers can communicate with each other. By using this metric, we compare the possibility that congestion occurs when some optical packet switches fail. Figure 6 shows the results. In Fig. 6, the horizontal axis indicates the failure rate of the optical packet switches, and the vertical axis indicates the maximum edge betweenness.

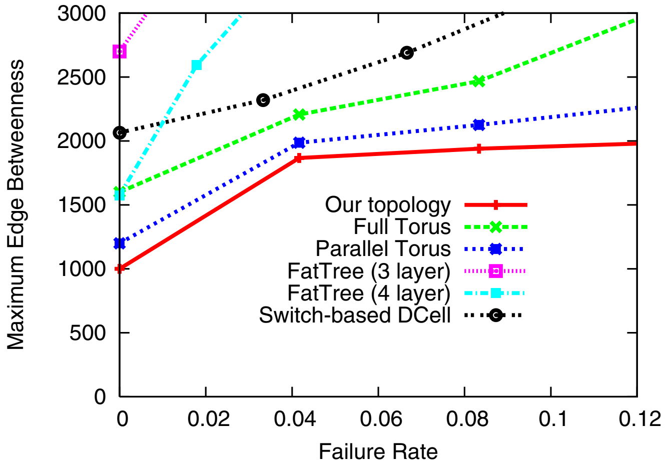


Figure 6: Edge Betweenness in Case of Failure

As shown in Fig. 6, the maximum edge betweennesses of the FatTree topologies increase faster than other topologies as the failure rate increases. In the FatTree topologies, because the number of shortest paths between ToR switches passing each link is large, the failure of each optical packet switch affects many ToR switch pairs. Moreover, the paths between the ToR switch pairs affected by the failure also pass many links. As a result, the failure of each optical packet switch has large impacts on the edge betweennesses of many links.

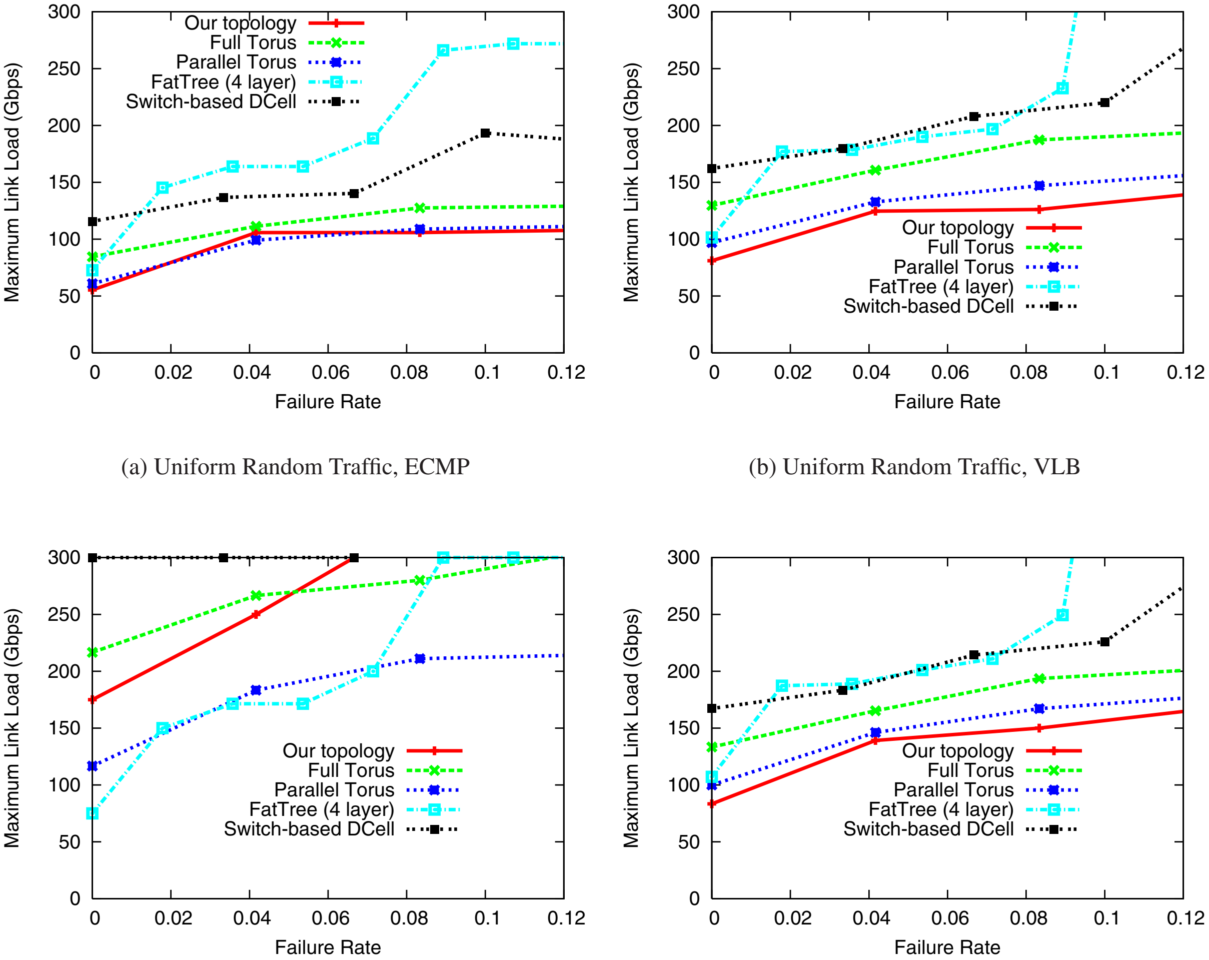
Fig. 6 also indicates that our topology has the smallest edge betweenness even when some optical packet switches fail. As discussed above, our topology has the smallest edge betweenness in the case of no failures. In addition, unlike the FatTree topologies, because the number of shortest paths between ToR switches passing each link and the average number of hops between ToR switches are small, the failure of each optical packet switch affects only few paths between ToR switch pairs, and few links. As a result, the edge betweenness of our method remains the smallest even when some optical packet switches fail.

We have also confirmed that the edge betweenness of our topology does not become large even when the failure rate becomes more than 0.12. However, the probability that ToR switch pairs unable to communicate with each other exist becomes large as the failure rate increases in our topology. In our topology, any optical packet switch has important links that connects different groups. Thus, as the failure rate increases, the number of redundant paths between groups decreases. Finally, when the number of paths between groups becomes 0 due to the failures, the ToR switches belonging to the different groups become unable to communicate with each other. However, as shown in Table 2, considering the worst case of failure, no topologies are more robust to failures than our topology. In addition, by setting $P_{\mathrm{opt}}^{\mathrm{tor}}$ to a large value, we can make our topology more robust to failures.

3.4.3 Maximum Link Load

In this subsection, we define the link load as the sum of traffic volume passing the link, and we compare the maximum link load without limiting the sum of traffic volume passing each link. In this evaluation, we generate the following two kinds of traffic.

Uniform Random Traffic is generated between all server pairs. We add the traffic, whose volume is randomly generated, between the randomly selected server pairs until the NICs of all servers have no remaining bandwidth.



(a) Uniform Random Traffic, ECMP

(b) Uniform Random Traffic, VLB

(c) Certain SW Pair Traffic, ECMP

(d) Certain SW Pair Traffic, VLB

Figure 7: Maximum Link Load

Certain SW Pair All of servers connected to the same ToR switch communicate with the servers connected to a certain ToR switch.

For each type of traffic, we randomly generate 20 patterns of traffic and calculate the maximum link load. In our evaluation, routes of traffic between ToR switches are calculated by the following policies.

ECMP Traffic between ToR switch is equally divided among all shortest paths.

VLB One intermediate optical packet switch is selected randomly regardless of the destination. Then the traffic is sent from the source ToR switch to the selected intermediate optical packet switch, and from the intermediate optical packet switch to the destination ToR switch.

In this evaluation, similar to Fig. 6, we generate the random failure of optical packet switches and investigate the maximum link utilization in the case that all server can communicate with each other. Figure 7 shows the results. In Fig. 7, the horizontal axis indicates the failure rate of the optical packet switches, and the vertical axis indicates the maximum link loads.

Figs. 7(a) and 7(b) indicate that our topology has the smallest link loads in the case of the uniform random traffic regardless of the routing. In the case of the uniform random traffic, link loads are proportional to the edge betweennesses. Thus, our topology, having the smallest edge betweenness as shown in Fig. 6, has the smallest link loads.

In the case of the certain switch pair traffic, our topology using the ECMP has much larger link loads than the parallel torus. This is caused by the number of distinct shortest paths. While the torus has many distinct shortest paths, the number of distinct shortest paths in our topology is small, which causes concentration of traffic on certain links.

By calculating routes with VLB, however, our topology achieves the smallest link loads even in the case of the certain switch pair traffic. This is because the parameters of our topology are set so as to avoid concentration of traffic on certain links when the routes are calculated by VLB. As shown in Fig. 7, among all pairs of the topologies and routing methods used in our evaluation, only the 4-layer FatTree topology using the ECMP achieves slightly smaller link loads than our topology in the case of no failures. The 4-layer FatTree, however, uses more than double optical packet switches and links between optical packet switches of our topology. In addition, similar to the edge betweenness shown in Fig. 6, the link loads of the 4-layer FatTree increase fast as the failure rate increases. Therefore, our topology is the most suitable topology for accommodating traffic between ToR switches when some optical packet switches fail.

4 Approach 2: Network to Achieve Low Energy Consumption

In this section, we focus on the energy consumption of the data center network, while the previous section focuses on the bandwidth provided between servers. Energy consumption of the data center network occupies a non-negligible fraction of the total energy consumption in the data center [3], and becomes large as the size of the network increases. Thus, to reduce energy consumption of a large data center, the energy consumption of the data center network should be reduced.

One approach to reduce the energy consumption is to construct the data center network that accommodate the traffic within the center with a small energy. The network suitable to the data center network depends on the applications and the current load of the data center. For the application where the servers exchange a large amount of data with each other, the network should provide large bandwidth between all servers related to the application. On the other hand, in the case of the application, where the servers exchange only a small amount of data, the small bandwidth is sufficient, and the network structure with only a small number of devices is preferable to reduce the energy consumption.

However, the traffic demands in the data center changes in time [21]. Moreover, servers related to the application, which suddenly become popular, may start exchanging a large amount of data. Additional servers related to the application may be implemented to handle the suddenly increased demand for the application. As a result, the data center network becomes no longer suitable for the current applications and loads. Though we can avoid the lack of bandwidth or large delay by constructing a redundant network, this approach consumes a large energy.

Therefore, the reconfiguration of the network structure based on the current traffic demands within a data center is required to accommodate the current traffic with only a small energy consumption. The network using the optical circuit switches enables the reconfiguration of the network structures. In this network, the core of the data center network is constructed by using the optical circuit switches and optical fibers. Then, the electronic packet switches, deployed in each server rack, are connected to the core network by connecting them to optical circuit switches. An optical path is established between two packet switches by configuring the optical circuit switches along the route between the electronic switches. A set of the optical paths and electronic packet switches forms a virtual network. Traffic between electronic switches is carried over the virtual network.

In this network, the energy consumption of the data center network is minimized by minimizing the number of ports of electronic switches used in the virtual network and shutting down the unused ports, because energy consumption of electronic switches is much larger than that of optical circuit switches. In the cases of the changes of demands, we keep the sufficiently large bandwidth, small delay between servers and low energy consumption by reconfiguring the virtual network.

Dynamic reconfiguration of the virtual network constructed over the optical network has also been discussed in many papers [22–26]. However, most of them aim to optimize the virtual network for the monitored or estimated current traffic demand, and are not applicable to the data center network, where the traffic changes within a second [27], because their calculation time is too large for a large data center.

Therefore, we have proposed a method to reconfigure the virtual network suitable for a large data center network [28]. In this section, we introduce this method proposed in Ref. [28]. In this method, the traffic changes in a short period are handled by the load balancing [20] over the virtual network. We design the virtual network so as to achieve sufficiently large bandwidth and small delay with low energy consumption, considering the load balancing. Then, if the current virtual network is not suitable to the current demands, the virtual network is reconfigured. Our method reconfigures the virtual network by setting parameters of a topology so as to avoid large calculation time in a large data center. As the topology used in the virtual network configuration, we introduce the topology called *Generalized Flattened Butterfly (GFB)*. We also introduce a method to set the parameters so as to suit the current condition.

4.1 Overview

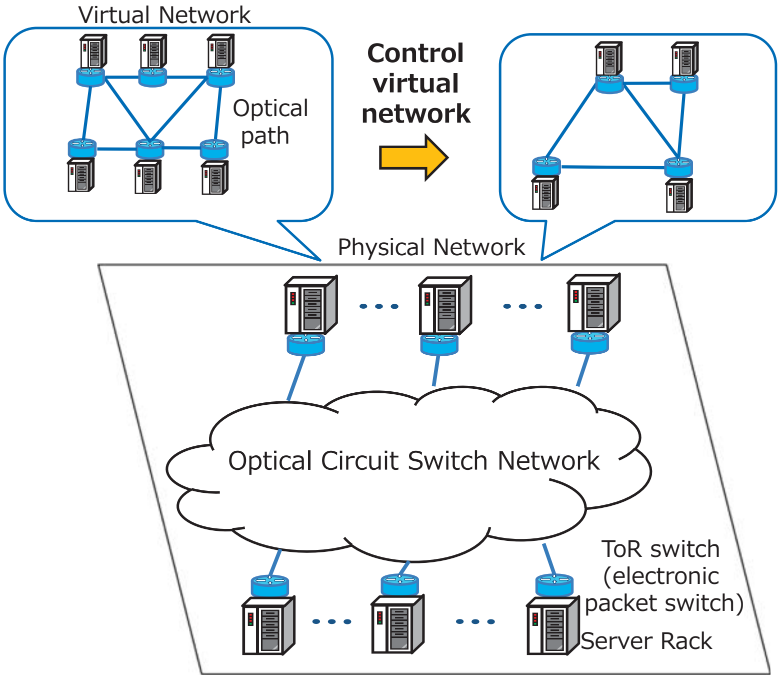


Figure 8: Data Center Network Using Optical Circuit Switches

In this section, we introduce the virtual network configured over the data center network constructed of the optical circuit switches and the electronic switches. In this network, the core of the data center network is constructed by using the optical circuit switches and optical fibers. Each ToR switch, which is an electronic packet switch, is connected to the core of the data center by connecting it to one of the port of optical circuit switches. An optical path is established between two electronic switches by configuring the optical packet switches along the route between the electronic switches. A set of the optical paths and ToR switches forms a virtual network, where each optical path is regarded as a link and each ToR switch is regarded as a node in the virtual network. Traffic between electronic switches is carried over the virtual network. In this network, the energy consumption of the data center network can be minimizing by minimizing the number of ports of electronic switches used in the virtual network and shutting down the unused ports, because energy consumption of electronic switches is much larger than that of optical circuit switches.

The virtual network can be easily reconfigured by adding or deleting optical paths if the current virtual network is no longer suitable. In the cases of the changes of demands, we keep the sufficiently large bandwidth, small delay between servers and low energy consumption by reconfiguring the virtual network. Moreover, servers related to the application, which suddenly become popular, may start exchanging a large amount of data in a data center. The method to reconfigure the virtual network is required to handle such traffic changes in a short period. However, existing methods to reconfigure the virtual network [23, 24] cannot be applicable in a large data center network, because these methods require the large calculation time for optimizing the virtual network in a large data center network.

Therefore, we introduce the method to reconfigure the virtual network by setting parameters of a topology so as to avoid large calculation time in a data center. As the topology used in the virtual network configuration, we introduce the topology called *Generalized Flattened Butterfly (GFB)*. We also introduce a method to set the parameters so as to suit the current condition.

4.2 Virtual Network Topologies Suitable to Optical Data Center Netwoks

Because it is difficult to obtain the optimal topology for a large data center network in a short time, our virtual network reconfiguration method constructs the virtual network by setting parameters of a topology, which is suitable for data center networks, instead of calculating the optimal topology which achieves the sufficiently large bandwidth, small delay between servers and low energy consumption.

In this subsection, we discuss the requirements for the topology used by our virtual network reconfiguration. Then, we introduce a new topology called *generalized flattened butterfly*, which can construct various data center networks by setting parameters.

4.2.1 Requirement

The virtual network should satisfy the following requirements.

Low Energy Consumption Energy consumption of the network occupies a non-negligible fraction of the total energy consumption in the data center. To reduce energy consumption of the data center, energy consumption of the data center network should be reduced.

In the data center network used in this section, most of energy is consumed by the ToR switches, because energy consumption of optical circuit switches is much smaller than that of ToR switches, and the energy consumption of the ToR switches can be reduced by shutting down their unused ports. Thus, by constructing the virtual network using the smallest number of ports of ToR switches, the energy consumption of the data center network is minimized.

Large Bandwidth between Servers In some applications such as distributed file system, a large amount of data is exchanged between servers. The bandwidth provided between servers is important for such applications; the lack of bandwidth increases the time required to transport data. Therefore, the virtual network should provide sufficient bandwidth between servers.

Small Delay between Servers A data center handles a large amount of data by using the distributed computing frameworks. In the distributed computing frameworks, a large number of servers communicate with each other. If the delays between servers are large, it takes time to obtain the required data from other servers, and the performance of the data center is degraded. Thus, the delay should be kept small enough for the application of the data center.

The delays between servers are difficult to forecast when constructing the virtual network, because the delays are affected by traffic load. In this section, we keep the small delay between servers by constructing the virtual network which can provide sufficient bandwidth and make the number of hops between servers small.

4.2.2 Existing Network Structures for Data Centers

Before introducing the our network topology, we introduce the existing network structures for data centers as the candidate topology used as the virtual network.

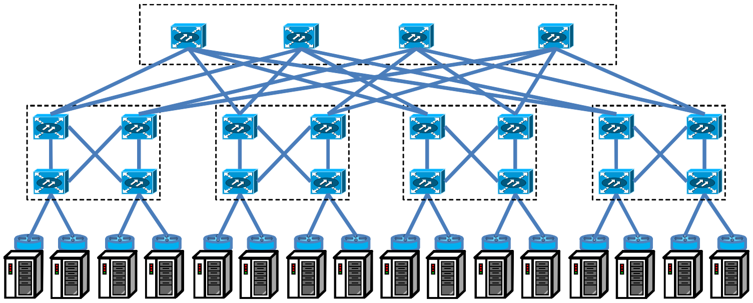


Figure 9: FatTree

FatTree One of the popular network structures for the data center is the topology called *FatTree*. The FatTree using switches with small number of ports has been proposed by Al-Fares et al. [4]. The FatTree is a tree topology constructed of multiple roots and multiple pods containing multi-layer of switches as shown in Figure 9.

Each pod is regarded as the virtual switch having a large number of ports constructed by multiple switches having a small number of ports. Pods are constructed as the *butterfly topology*, where each switch uses a half of its ports to connect it to the switches of the upper layer, and the other half of its ports to connect it to the switches of the lower layer. The switches at the lowest layer are connected to the servers.

Though the method proposed by Al-Fares et al. [4] constructs the 3-layer FatTree, which is constructed of root switches and the pods containing two layers of switches, we can construct the higher-layer FatTree topologies. The k-layer FatTree constructed of switches with n ports includes $(2k-1)\frac{n}{2}^{k-1}$ switches.

In the FatTree, the number of links from the lower-layer switch equals the number of links to the upper-layer switch at each switch. That is, the sum of bandwidth from a switch to the upper layer equals that from the lower layer to the switch. Therefore, any switch does not become a bottleneck, and we can provide sufficiently large bandwidth between all servers.

However, the FatTree is not suitable to the virtual network constructed of ToR switches. In the FatTree, the switches at the upper layer are not connected to servers. This means that the ToR switches that are not connected to servers should be powered on, which leads to large energy consumption.

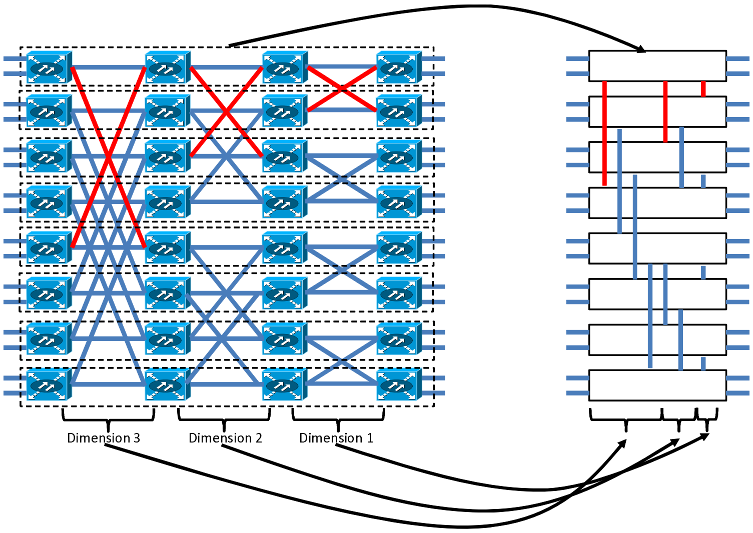


Figure 10: Flattened Butterfly

Flattened Butterfly Kim et al. [5] have proposed the data center network topology called *flattened butterfly*. The flattened butterfly is constructed by *flattening* the butterfly topology as shown in Figure 10; we combine the switches in each row of the butterfly topology into a single switch.

The flattened butterfly provides sufficiently large bandwidth between all servers with lower energy consumption than the FatTree [4]. In addition, all switches in the flattened butterfly are connected to servers. Thus, unlike the FatTree, all ToR switches, which are not connected to any working servers, can be shut down if the flattened butterfly is constructed as the virtual network. However, the flattened butterfly requires the switches with a large number of ports to construct a large data center network. Thus, the flattened butterfly is not preferable when the traffic demands are small.

DCell Guo et al. have proposed a data center network called *DCell*, which is constructed from a small number of switches and servers with multiple ports as shown in Figure 11 [6]. DCell uses a recursively-defined structure; the level-0 DCell is constructed by connecting one switch with n ports to n servers, and the level-k DCell is constructed by connecting servers belonging to different level-$k-1$ DCells.

By directly connecting server ports, DCell reduces the number of switches required to construct a large data center network. However, DCell is not used as the topology of the virtual network introduced in this section, which is constructed of ToR switches.

Therefore, we introduce the topology called *switch-based DCell*, where the level-0 DCell is replaced with the fully-connected network constructed of switches as shown in Figure 12. Similar to the DCell, the switch-based DCell can construct a large data center network by using the switches with small number of ports. That is, the switch-based DCell achieves low energy consumption. However, the switch-based DCell cannot provide large bandwidth between all servers, because the switch-based DCell has only one link between each lower-level DCells.

4.2.3 Generalized Flattened Butterfly

As discussed above, the flattened butterfly [5] provide sufficient bandwidth between all server pairs but requires large energy. DCell [6] can construct the topology which includes a large number of servers using a small number of ports but cannot provide sufficient bandwidth.

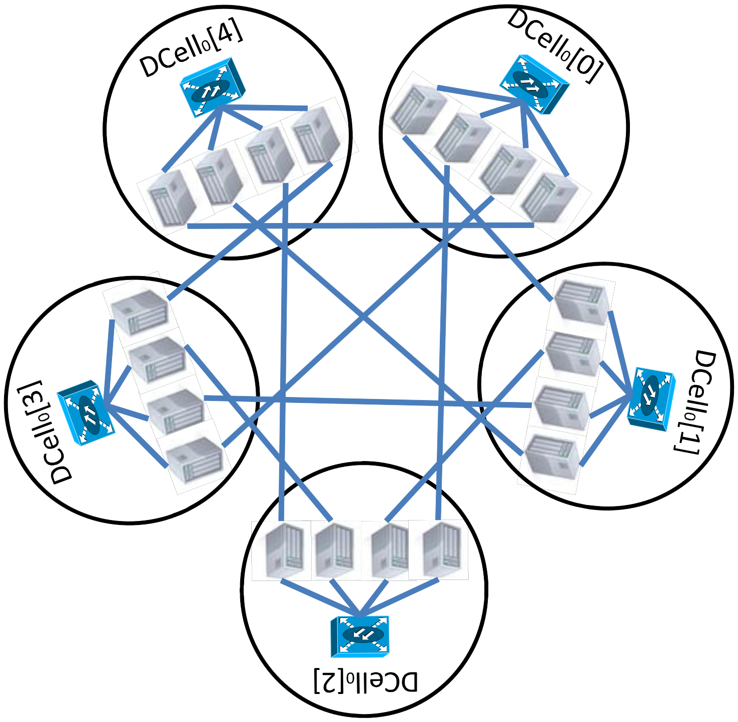


Figure 11: DCell

In this subsection, we introduce a topology called *Generalized Flattened Butterfly (GFB)*. In the GFB, the number of required ports, the maximum number of hops and the bandwidth provided between servers can be changed by setting the parameters. The GFB is constructed hierarchically; the upper-layer GFB is constructed by connecting multiple lower-layer GFBs. The GFB has the following parameters.

- Number of layers: k
- Number of links per node used to construct layer-k GFB: L_k
- Number of layer-$k-1$ GFBs used to construct layer-k GFB: N_k

By setting these parameters, we can construct various topologies including the flattened butterfly and the switch-based DCell.

Steps to Construct the Generalized Flattened Butterfly The layer-k GFB is constructed by the following two steps.

Step I Construct the connections between the layer-$k-1$ GFBs.

Step II Select the switches connected to the links between each layer-$k-1$ GFB pair

In these steps, we use the ID assigned for the GFBs of each layer. The switch can be identified by the set of IDs of the GFBs the switch belongs to. We denote the ID of the layer-k GFB the switch s belongs to as

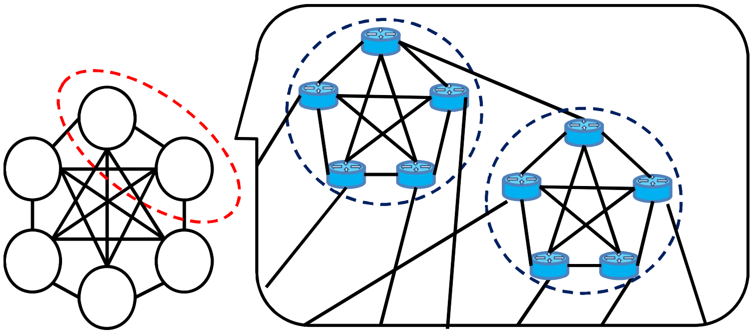


Figure 12: Switch-based DCell

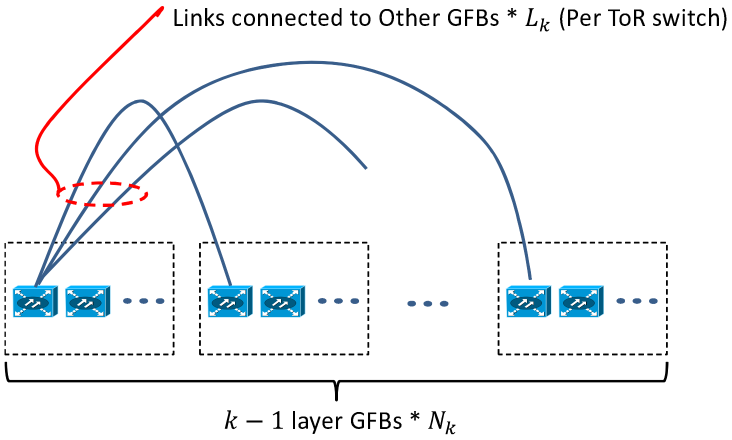


Figure 13: Generalized Flattened Butterfly

$D_k^{GFB}(s)$. We define the ID of the switch s in the layer-k GFB by

$$D_k^{sw}(s) = \sum_{1 \leq i \leq k} \left(D_i^{GFB}(s) \prod_{j=1}^{i-1} N_j \right).$$

Connections between layer-$k-1$ GFBs We construct the connections between the layer-$k-1$ GFBs by the following steps.

Step I.I Calculate the number of links used to connect one layer-$k-1$ GFB to the other layer-$k-1$ GFBs, L_k^{GFB}, by

$$L_k^{GFB} = L_k \prod_{i=1}^{k-1} N_i \tag{9}$$

Step. I.II If L_k^{GFB} is larger than $(N_k - 1)$, we can connect all layer-$k-1$ GFB pairs. Otherwise, construct the ring topology by connecting the GFBs having the nearest ID.

Step I.III Calculate the number of the residual links $L_k^{'GFB}$ which can be used to connect one layer-$k-1$ GFB to the other layer-$k-1$ GFBs by

$$L_k^{'GFB} = L_k^{GFB} - \bar{L}_k^{GFB} \tag{10}$$

where $\bar{L}_k^{GFB}$ is the number of links per layer-$k-1$ GFB constructed at Steps I.II.

Step I.IV Check whether layer-$k-1$ GFBs have residual links to be used connect layer-$k-1$ GFBs. If yes, connect the GFB of ID $D_{k-1}^{GFB}(a)$ to the GFB of ID $D_{k-1}^{GFB}(b)$ where the following equation is satisfied.

$$D_{k-1}^{GFB}(b) = (D_{k-1}^{GFB}(a) + \lceil p_k \rceil + C\lfloor p_k \rfloor) \bmod N_k. \tag{11}$$

C is the integer value and p_k is the value which defines the distance of the connected layer-$k-1$ GFBs, and calculated by following equation;

$$p_k = \frac{N_k}{L_k^{'gfb} + 1}. \tag{12}$$

In the GFB, the links are connected at the equal distance of the ID of the layer-k GFB so as to minimize the maximum number of hops between the layer-k GFBs.

Selection of the switches used to connect layer-$k-1$ GFBs After constructing the connections between layer-$k-1$ GFBs, we select the switches that are used to connect the layer-$k-1$ GFB pair. The switch $D^{sw}(s)$ included in the GFB of ID $D_{k-1}^{GFB}(a)$ is connected to the GFB of ID $D_{k-1}^{GFB}(b)$ when the following condition is satisfied.

$$D^{sw}(s) = D_{k-1}^{gfb}(b) + \left\lfloor \frac{Cn_{D_{k-1}^{gfb}(a)}}{l_{(D_{k-1}^{gfb}(a), D_{k-1}^{gfb}(b))}} \right\rfloor$$

where C is a integer value, $n_{D_{k-1}^{gfb}(a)}$ is the number of switches in the GFB of ID $D_{k-1}^{gfb}(a)$, and $l_{(D_{k-1}^{gfb}(a), D_{k-1}^{gfb}(b))}$ is the number of links to be constructed between GFBs of IDs $D_{k-1}^{gfb}(a)$ and $D_{k-1}^{gfb}(b)$. By connecting switches using the above condition, the intervals of switch connected to the same GFB become constant, and we can avoid the large number of hops from a switch to the other GFB.

Properties of the Generalized Flattened Butterfly In the GFB, the maximum number of hops or the number of paths passing each link can be calculated from the parameters as described below.

Maximum Number of Hops The maximum number of hops between switches in the layer-k GFB, H_k is calculated by

$$H_k = (h_k + 1)H_{k-1} + h_k, \tag{13}$$

where h_k is the largest number of links between layer-$k-1$ GFBs passed by the traffic between layer-$k-1$ GFBs. H_k is obtained by calculating h_k. In the rest of this paragraph, we discuss how to calculate h_k from the parameters of the GFB.

If L_k^{gfb} defined by Eq. (9) is larger than $M_k(N_k-1)$, we add links between all pairs of layer-$k-1$ GFBs. Thus, $h_k = 1$.

If L_k^{GFB} is smaller than $N_k - 1$ and $L_k^{'GFB}$ defined by Eq. (10) is zero, the connections between layer-$k-1$ GFBs form a ring topology. In this case, h_k is $\lceil \frac{N_k}{2} \rceil$.

If L_k^{GFB} is smaller than $(N_k - 1)$ and $L_k^{'GFB}$ is a positive value, we add links to the GFBs satisfying Eq. (11). In this case, we discuss the calculation of h_k by dividing the topology constructed of layer-$k-1$ GFBs into modules so that each module includes the GFB whose ID is within the range from Cp_k to $(C+1)p_k$ where C is a integer variable and p_k is defined by Eq. (12). Then, we calculate the maximum number of hops from the source GFB whose ID is zero. Since all low-layer GFBs play the same role in the high-layer GFB, h_k is calculated by calculating the maximum number of hops from the GFB whose ID is zero.

If $L_k^{'GFB}$ is one, the topology constructed of layer $k-1$ GFBs are divided into two modules as shown in Figure 14. In this case, each module becomes a ring topology whose number of nodes is p_k. That is, h_k is $\lceil \frac{p_k}{2} \rceil$ in this case.

If $L_k^{'GFB}$ is more than one, the topology constructed of layer $k-1$ GFBs are divided into more than two modules. In this case, at lease one module does not include the source GFB, and the module without the source GFB includes the GFB whose number of hops from the source GFB is the largest. The modules without the source GFB includes $\lfloor p_k \rfloor$ GFBs. The GFBs at the both edges of the module are connected to the source GFB as shown in Figure 15. As shown in this figure, the source GFB and the GFB included in the modules form a ring topology with $\lfloor p_k \rfloor + 2$ nodes. Thus, h_k is $\lceil \frac{\lfloor p_k \rfloor + 2}{2} \rceil$ in this case.

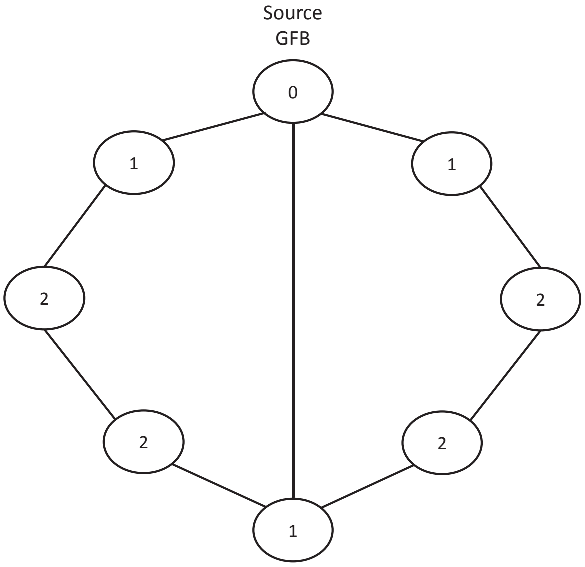


Figure 14: Example of number of hops in the topology constructed of low-layer GFBs ($L_k^{'GFB} = 1$)

Summarizing the above discussion, h_k is calculated by

$$h_k = \begin{cases} 1 & (L_k^{GFB} \geq (N_k - 1)) \\ \lceil \frac{p_k}{2} \rceil & (L_k^{GFB} < (N_k - 1) \text{ and } L_k^{'GFB} = 0, 1) \\ \lceil \frac{\lfloor p_k \rfloor + 2}{2} \rceil & (Otherwise) \end{cases} \tag{14}$$

h_k is defined by constructing the layer-k GFB in Step. I, h_k can be calculated by the parameters. If the links are added between all pair of layer-$k-1$ GFBs, h_k is 1. Otherwise, the links are added to the layer-$k-1$ GFB satisfying Eq. (11). In this case, h_k is defined by p_k.

Number of Flows through a Link The number of layer-$k-1$ GFB pairs whose traffic passes the link l between the layer-$k-1$ GFBs x_l^k is obtained by calculating the number of flows passing the link in the

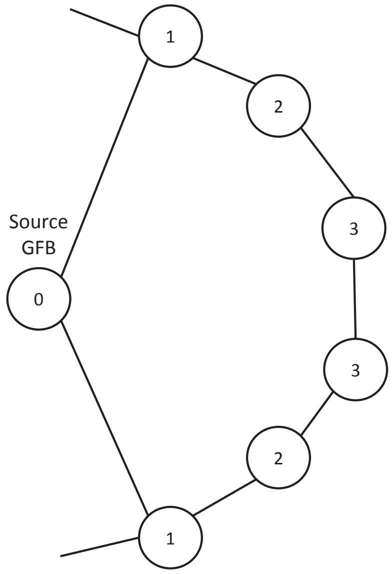


Figure 15: Example of number of hops in the topology constructed of low-layer GFBs ($L_k^{\prime GFB} > 1$)

abstracted topology where the layer-$k-1$ GFB is regarded as a single node. Then, by multiplying it with the number of flows passing the layer-$k-1$ GFB pairs , we obtain the number of flows passing each link. Since all layer-k GFB play the same role, the number of flows passing between the layer-$k-1$ GFB pair is independent of the ID of the GFB.

Thus, the number of flows passing the link l between layer-$k-1$ GFBs X_l^k is obtained by

$$X_l^k = F_k x_l^k,$$

where F_k is the number of flows from a layer-$k-1$ GFB to the other layer-$k-1$ GFB. Hereafter, we calculate x_l^k and F_k.

We first calculate x_l^k. In the abstracted topology where the lower-layer GFB is regarded as a single node, there are two kinds of links; one is the link on the ring topology (hereafter we call this link *ring link*), and the other is the link added to shortcut the ring topology (hereafter we call this link *shortcut link*).

Since all layer-$k-1$ GFBs play the same role in the layer-k GFB, the number of flows passing each ring link is independent of the GFBs connected to the link. Similarly, the number of flows passing each shortcut link is also independent of the GFBs connected to the link. Therefore,

$$x_l^k \quad = \quad \begin{cases} \frac{M_k^{\text{ring}}}{2\prod_{i=1}^{k} N_i} & (l \text{ is a ring link}) \\ \frac{M_k^{\text{shortcut}}}{(L_k-2)\prod_{i=1}^{k} N_i} & (l \text{ is a shortcut link}) \end{cases} . \tag{15}$$

where M_k^{ring} is the total of the ring links passed by the traffic between layer-$k-1$ GFB pairs, and M_k^{shortcut} is the total of the shortcut links passed by the traffic between layer-$k-1$ GFB pairs. $2\prod_{i=1}^{k} N_i$ is the number of ring links between layer-$k-1$ GFBs, and $(L_k-2)\prod_{i=1}^{k} N_i$ is the number of shortcut links between layer-$k-1$ GFBs.

The traffic between layer-$k-1$ GFBs passes at most one shortcut links , because the interval of the IDs of the GFBs connected to a certain GFB is constant. The number of flows that does not pass the shortcut link is $2h_k \prod_{i=1}^{k} N_i$. Thus,

$$M^{\text{shortcut}} = \prod_{i=1}^{k} N_i(\prod_{i=1}^{k} N_i - 1) - 2h_k \prod_{i=1}^{k} N_i.$$

In addition, M^{ring} is obtained by subtracting M^{shortcut} from the total number of links passed by the traffic between layer-$k-1$ GFBs ;

$$M^{\text{ring}} = \sum_{i=1}^{h_k} i s_k(i) - M^{shortcut}.$$

where $s_k(i)$ is the number of layer $k-1$ GFB pairs whose traffic passes i links in the abstracted topology.

$s_k(i)$ is obtained as follows. $s_k(1)$ is the same value as the number of links in the layer-k GFB. That is,

$$s_k(1) \quad = \quad \begin{cases} N_k(N_k - 1) & (L_k^{GFB} \geq (N_k - 1)) \\ N_k L_k \prod_{i=1}^{k-1} N_i & (\text{otherwise}) \end{cases} . \tag{16}$$

$s_k(i)$ for $i > 1$ is calculated by dividing the topology constructed of layer-$k-1$ GFBs into groups similar to the case of calculating h_k. By dividing the topology, $s_k(i)$ is calculated by the sum of the number of the layer-$k-1$ GFBs i hops away from source layer-$k-1$ GFB in each group. Thus, $s_k(i)$ is calculated by

$$s_k(i) = N_k \sum_{m_j \in M} U_{(k,m_j)}(i), \tag{17}$$

where $U_{(k,m_j)}(i)$ is the number of the layer-$k-1$ GFBs i hops away from the source layer-$k-1$ GFB in the group m_j. Since the GFBs included in each group, the source GFB and the GFBs directly connected to the source GFB form a ring topology,

$$U_{(k,m_j)}(i) \quad = \quad \begin{cases} 0 & \left(i > \left\lceil \frac{m_j+2}{2} \right\rceil\right) \\ 1 & \left(i = \left\lceil \frac{m_j+2}{2} \right\rceil \text{ and } |m_j| \text{ is odd}\right) \\ 2 & (Otherwise) \end{cases} \cdot \tag{18}$$

We calculate the number of flows between each layer-$k-1$ GFB pair, F_k. The number of flows between each layer-$k-1$ GFB pair is independent of the ID of the source or destination GFB. Thus, we calculate the number of flows passing between layer-$k-1$ GFBs s and d, $F_k^{s \to d}$.

$F_k^{s \to d}$ is calculated by

$$\begin{aligned} F_k^{s \to d} = f_k^{s \to s \to d \to d} + \sum_{n \in G} f_k^{n \to s \to d \to d} \\ + \sum_{n \in G} f_k^{s \to s \to d \to n} + \sum_{n_1, n_2 \in G} f_k^{n_1 \to s \to d \to n_2}, \end{aligned} \tag{19}$$

where $f^{a \to b \to c \to d}$ is the number of flows whose source and destination switches belong to the layer-$k-1$ GFBs a and d and that traverse the layer $k-1$ GFBs b and c. G is the set of switches that do not belong to the layer-k GFB including the layer-$k-1$ GFBs s and d.

$f_k^{s \to s \to d \to d}$ is calculated by the product of the number of switches included in the layer-$k-1$ GFB s and that included in the layer-$k-1$ GFB d. That is,

$$f_k^{s \to s \to d \to d} = \prod_{i=1}^{k-1} (N_i)^2. \tag{20}$$

$\sum_{n \in G} f_k^{s \to s \to d \to n}$ indicates the number of flows from the layer-$k-1$ GFB s to the outside of the layer k GFB via the layer $k-1$ GFB d. Because all layer-$k-1$ GFBs play the same role in the GFB, $\sum_{n \in G} f_k^{s \to s \to d \to n}$ is calculated by dividing the number of flows whose source switches belong to the layer-$k-1$ GFB s and destination switches belong to the different layer-k GFB by the number of layer-$k-1$ GFBs in the layer-k GFB.

$$\sum_{n \in G} f_k^{s \to s \to d \to n} = \frac{(\prod_{i=1}^{k-1} N_i)(\prod_{i=1}^{K_{\max}} N_i - \prod_{i=1}^{k} N_i)}{N_k}. \tag{21}$$

Similarly, $\sum_{n \in G} f_k^{n \to s \to d \to d}$ is calculated by

$$\sum_{n \in G} f_k^{n \to s \to d \to d} = \frac{(\prod_{i=1}^{k-1} N_i)(\prod_{i=1}^{K_{\max}} N_i - \prod_{i=1}^{k} N_i)}{N_k}. \tag{22}$$

$\sum_{n_1, n_2 \in G} f_k^{n_1 \to s \to d \to n_2}$ indicates the number of flows that come from the outside of the layer-k GFB via the layer-$k-1$ GFB s and go to the outside of the layer-k GFB via the layer-$k-1$ GFB d. The number of flows coming from the outside of the layer-k GFB via the layer-$k-1$ GFB s is the sum of flows on the links that connect switches in the layer-$k-1$ GFB s and the switches outside the layer-k GFB, which is calculated by

$$\prod_{j=1}^{k-1} N_j \sum_{i=k+1}^{K} (X_l^i L_i). \tag{23}$$

We obtain the number of flows that come from the outside of the layer-k GFB via the layer-$k-1$ GFB s and are sent to the layer-$k-1$ GFB d by dividing Eq. (23) by the number of the layer-$k-1$ GFBs in the layer-k GFB. This calculated value includes the flows whose destination switches belong to the layer-$k-1$ GFB d, whose number is $\sum_{n_1 \in G} f_k^{n_1 \to s \to d \to d}$. Therefore, $\sum_{n_1, n_2 \in G} f_k^{n1 \to s \to d \to n_2}$ is calculated by

$$\begin{aligned} \sum_{n_1, n_2 \in G} f_k^{n_1 \to s \to d \to n_2} = & \frac{\prod_{j=1}^{k-1} N_j \sum_{i=k+1}^{K} (X_l^i L_i)}{N_k} \\ & - \sum_{n_1 \in G} f_k^{n_1 \to s \to d \to d}. \end{aligned} \tag{24}$$

4.3 Virtual Network Topology Control to Achieve Low Energy Consumption

4.3.1 Outline

In our method, the virtual network is constructed so as to minimize the number of used ports considering two kinds of requirements; bandwidths and delay between servers.

One approach to provide sufficient bandwidths between servers is to construct the virtual network that can accommodate the current traffic demands between servers. However, in a data center, traffic may change within a second [27]. Thus, if the virtual network is optimized for the current traffic demands, the virtual network may be required to be reconfigured every second. However, in a large data center, the calculation time to optimize the virtual network for the current traffic demands becomes too large. Therefore, the virtual network cannot be constructed to accommodate the current traffic demands between servers.

In our method, the traffic changes in a short period are handled by the load balancing [20] over the virtual network. And, we design the virtual network so as to achieve sufficiently large bandwidth and small delay with small energy consumption, considering the load balancing.

In this section, we use one of the load balancing technique called *Valiant Load Balancing (VLB)* [20]. In the VLB, we select the intermediate nodes randomly regardless of the destination to avoid the concentration of traffic on certain links even when traffic amount of a certain node pair is large. Then, traffic is sent from the source node to the intermediate node and from the intermediate node to the destination node. By applying the VLB, the amount of traffic between each ToR switch pair T is calculated by the following equation.

$$T \leq \frac{T^{SWto} + T^{SWfrom}}{N_{\text{all}}}. \tag{25}$$

In this equation, T^{SWto} is the maximum traffic amount to a ToR switch, T^{SWfrom} is the maximum traffic amount from a ToR switch, and N_{all} is the number of ToR switches in the virtual network. Thus, we provide sufficient bandwidth by making the number of flows passing a link less than a threshold, which is calculated by dividing the capacity of an optical path by the traffic amount between each switch pair calculated by Eq.(25).

The delay is also hard to forecast when designing the virtual network. In this section, we avoid too large delay by providing enough bandwidth and making the maximum number of hops less than a threshold.

4.3.2 Topology Control to Satisfy the Requirements

In this subsection, we introduce a method to set the parameters of the GFB so as to minimize the number of used ports and satisfy the requirements of the bandwidth and the maximum number of hops between servers.

In our method to set the parameters, the number of switches connected in the virtual network N_{all}, the acceptable maximum number of hops H_{max}, the maximum traffic amount from a ToR switch T^{SWfrom}, and the maximum traffic amount to a ToR switch T^{SWto} are given. Our method sets the parameters by the following steps.

First, we calculate the candidates of the number of layers. Because we cannot make the maximum number of hops of the GFB less than the case that $h_k = 1$ in Eq. (13) for all layers, to make the maximum number of hops less than H_{max}, the number of layers $K_{\max}$ must satisfy the following condition.

$$2^{K_{\max}} - 1 \leq H_{max} \tag{26}$$

We consider the all $K_{\max}$ satisfying the above condition as the candidates of the number of layers. For each candidate, we set suitable parameters by the following steps.

Step 1 Set the parameters considering the acceptable number of hops.

Step 2 Modify the parameters so as to provide the sufficient bandwidth.

Then, we construct the topology which uses the smallest number of virtual links among the candidates. The details of the above steps are described in the following paragraphs.

Parameter Settings considering the acceptable number of hops We set parameters N_k and L_k so as to make the maximum hops less than H_{max}. In this steps, to reduce the number of variables, we set N_k to $\prod_{i=1}^{k-1} N_i + 1$ for $1 < k < K_{\max}$. By doing so, h_k becomes 1 even when $L_k = 1$.

To connect N_{all} switchs, $N_{K_{\max}}$ must satisfy the following equation.

$$N_{K_{\max}} = \left\lceil \frac{N_{\text{all}}}{\prod_{i=1}^{k-1} N_i} \right\rceil. \tag{27}$$

In this steps, we also set $L_{K_{\max}}$ so that $h_{K_{\max}}$ becomes 1 to reduce the number of variables. To make $h_{K_{\max}}$ 1, $L_{K_{\max}}$ should satisfy the following equation.

$$L_{K_{\max}} = \left\lceil \frac{N_{K_{\max}}}{\prod_{i=1}^{k-1} N_i} \right\rceil . \tag{28}$$

To make the maximum number of hops less than H_{max}, h_1 must satisfy the following conditions, according to Eq. (14).

$$h_1 \leq \left\lceil \frac{H_{\max}+1}{2^{K-1}} - 1 \right\rceil . \tag{29}$$

To satisfy Eq. (29), L_1 should satisfy the following equation.

$$L_1 = \begin{cases} N_1 - 1 & (h_1 = 1) \\ 2 & (h_1 \geq \lfloor \frac{N_1}{2} \rfloor) \\ \lfloor \frac{N_1}{2h_1} + 1 \rfloor & (Otherwise) \end{cases} . \tag{30}$$

In the above condition, all N_k ($k > 1$) and L_k are calculated by N_1. The objective of our parameter setting is to minimize the number of used ports of ToR switches. That is, we minimize $\sum_{1 \leq k \leq K} L_k$. Since $\sum_{1 \leq k \leq K} L_k$ is the convex function of N_1, we find the N_1 that minimizes $\sum_{1 \leq k \leq K_{\max}} L_k$ by incrementing N_1 as long as $\sum_{1 \leq k \leq K_{\max}} L_k$ decreases.

Parameter Modifications to Provide the Sufficient Bandwidth If the GFB with the parameters set at Steps 1 cannot provide the sufficient bandwidth, we add the links to the layer where the sufficient bandwidth cannot be provided. To detect the lack of bandwidth, we check whether the following condition is satisfied for each layer k.

$$TX_l^k \leq B \tag{31}$$

where B is the bandwidth of one link, and T is calculated by Eq. (25). If Eq. (31) is not satisfied, we add L_k until Eq. (31) is satisfied.

4.4 Evaluation

We investigate the number of ports of ToR switches required to achieve the requirements. In this comparison, all topologies include 420 ToR switches. We compare the topology constructed by our method with four topologies, FatTree, Torus, Switch-based DCell [6] and Flattened Butterfly [5]. Unlike the FatTree topology proposed by Al-Fares et al. [4], we assume that the traffic is generated not only from the switches at the lowest layer but also from the switches at the upper layer in the FatTree used in this evaluation, since powering up additional switches consumes more energy. In our evaluation, the parameters of each topology are set so as to minimize the number of ports required by the topology under the constraint that it can provide the sufficient bandwidth and the maximum number of hops is less than H_{max}.

In this evaluation, we assume that the number of wavelengths on optical fibers is sufficient. We set the bandwidth of one optical path to 10 Gbps.

We compare the number of virtual optical paths per ToR switch required to achieve the requirements by changing the maximum traffic amount from or to each ToR switches. In this comparison, we set the acceptable maximum number of hops to a sufficiently large value. That is, the bandwidth provided for each ToR switch is the requirement for the virtual network.

Figure 16 shows the results. In this figure, the horizontal axis indicates the maximum traffic amount from or to ToR switches that is required to be accommodated, and the vertical axis indicates the number of virtual links per ToR switch required to satisfy the requirement. As shown in this figure, the switch-based

DCell cannot accommodate traffic more than 1 Gbps, and the FatTree and the torus cannot accommodate traffic more than 6 Gbps per ToR switch. In the switch-based DCell, the link between level-0 DCells becomes bottleneck, which cannot be solved by the parameter settings. In the FatTree topologies, we cannot construct the topology having more links than the 3-layer FatTree. Thus, the FatTree cannot accommodate more traffic that cannot be accommodated in the 3-layer FatTree. Similarly, the torus cannot accommodate more traffic that cannot be accommodated in the torus whose dimension is the largest among the torus constructed by 420 switches.

Though the flattened butterfly can accommodate a large amount of traffic, it requires a large number of virtual links. This figure indicates that our method uses the smallest number of virtual links to accommodate traffic regardless of the amount of traffic. This is because our method to set parameters of the GFB adds only links that are necessary to accommodate the traffic. Therefore, the topology constructed by our method satisfies the requirement of the bandwidth with the smallest energy consumption.

We also compare the number of virtual links per ToR switch required to achieve the requirements by changing the acceptable maximum number of hops. In this comparison, we assume that the capacity of each virtual link is sufficient. That is, the acceptable maximum number of hops is the only requirement for the virtual network.

Figure 17 shows the results. In this figure, the horizontal axis indicates the maximum number of hops, and the vertical axis indicates the number of virtual links per ToR switch required to satisfy the requirement. As shown in this figure, the flattened butterfly requires a large number of virtual links even if the acceptable maximum number of hops is large.

On the other hand, the switch-based DCell and Fattree can construct the topology by using the small number of links when the acceptable maximum number of hops is large. However, these topologies require the large number of virtual links when the acceptable maximum number of hops is small. In these topologies, the maximum number of hops is defined by the number of layers. These topologies require the small number of layers when the acceptable maximum number of hops is small. However, the small number of layers requires the large number of links in theses topologies constructed by 420 switches. The torus constructed by 420 switches cannot accommodate the acceptable number of hops less than 7 hops.

As shown in this figure, in all cases of the acceptable maximum number of hops, the topology constructed by our method uses the smallest number of virtual links to satisfy the requirements. This is because our method to set parameters of the GFB adds only links that are necessary to achieve the maximum number of hops. Therefore, the topology constructed by our method satisfies the requirement of the maximum number of hops with the smallest energy consumption.

5 Conclusion

The optical data center network is one approaches to construct the data center network that provides a large bandwidth between servers with a small energy consumption. In the optical data center network, we can use two type of optical switches; optical packet switches and optical circuit switches.

In this chapter, we introduced two approaches from our researches [28,29]. The first approach was introduced as an application of the optical packet switches. This approach focuses on the bandwidth provided for all-to-all communication. In this approach, we deploy the optical packet switches and construct the network structure that efficiently use the large bandwidth provided by the optical packet switches.

The other approach was introduced as an application of the optical circuit switches. This approach aims to minimize the energy consumption. In this approach, the optical circuit switches are placed at the core of the data center network. The ToR switches are connected to one of the ports of the optical circuit switches. The virtual network constructed of ToR switches are constructed by setting the optical circuit switches. The topology of the virtual network is changed by reconfiguring the optical circuit switches. By setting the

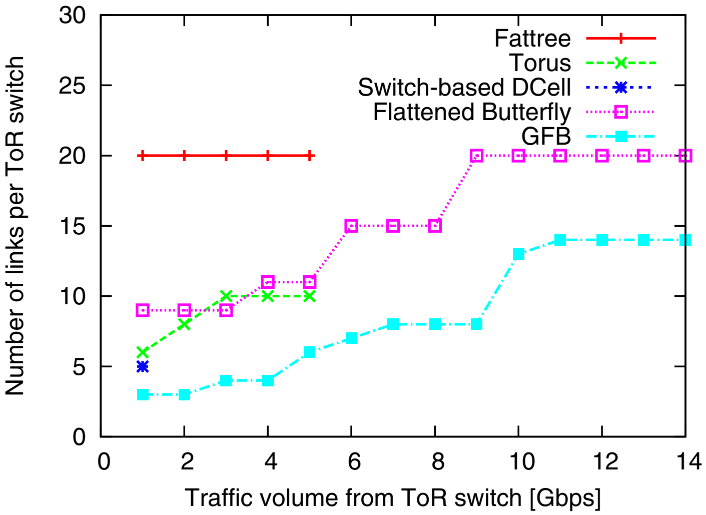


Figure 16: Number of virtual links required to accommodate the traffic from ToR switches

optical circuit switch so as to minimize the number of required ports of the packet switches, we reduce the energy consumption. For this approach, we also introduced a method to calculate the suitable settings of optical circuit switches.

One of the future direction is to combine these two approaches. Though the optical packet switch consumes less energy than the electronic packet switch, the optical packet switch consumes more energy than the optical circuit switch because the optical packet switch requires the label processing, buffers and so on. Thus, similar to the second approach explained in this chapter, the method to construct a virtual network of optical packet switches by setting optical circuit switches may accommodate more traffic with less energy consumption.

Acknowledgement

Our researches introduced in this chapter are done as a part of “Research & Development of Basic Technologies for High Performance Opto-electronic Hybrid Packet Router” supported by National Institute of Information and Communications Technology (NICT).

References

[1] S. Ghemawat, H. Gobioff, and S. Leung, “The google file system,” in *Proceeding of ACM SIGOPS Operating Systems Review*, vol. 37, pp. 29–43, ACM, Dec. 2003.

[2] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[3] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu, “Energy proportional datacenter networks,” *ACM SIGARCH Computer Architecture News*, vol. 38, pp. 338–347, June 2010.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in *Proceedings of ACM SIGCOMM*, vol. 38, pp. 63–74, Aug. 2008.

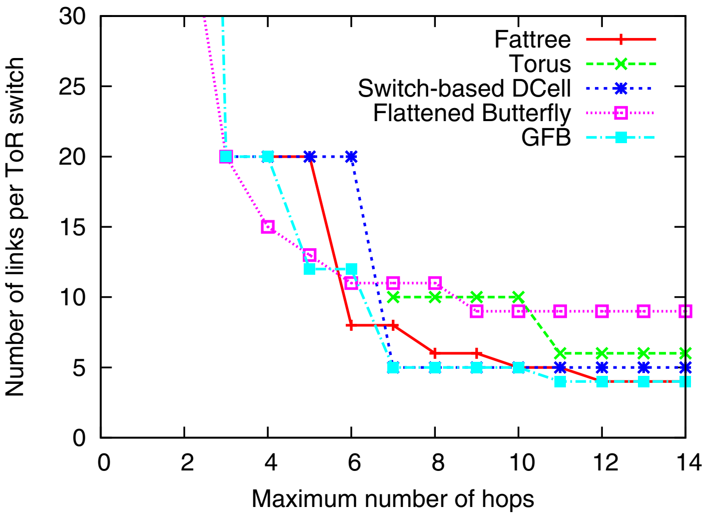


Figure 17: Number of virtual links required to make the maximum number of hops less than the target value

[5] J. Kim, W. Dally, and D. Abts, “Flattened butterfly: a cost-efficient topology for highradix networks,” in *Proceedings of ISCA*, vol. 35, pp. 126–137, June 2007.

[6] 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 Computer Communication Review*, vol. 38, pp. 75–86, Aug. 2008.

[7] 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,” *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 63–74, Aug. 2009.

[8] D. Guo, T. Chen, D. Li, Y. Liu, X. Liu, and G. Chen, “BCN: expansible network structures for data centers using hierarchical compound graphs,” in *Proceedings of INFOCOM*, pp. 61–65, Apr. 2011.

[9] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu, “Scalable and cost-effective interconnection of data-center servers using dual server ports,” *IEEE/ACM Transactions on Networking*, vol. 19, pp. 102–114, Feb. 2011.

[10] Y. Liao, D. Yin, and L. Gao, “Dpillar: Scalable dual-port server interconnection for data center networks,” in *Proceedings of ICCCN*, pp. 1–6, Aug. 2010.

[11] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, “VL2: A scalable and flexible data center network,” *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 51–62, Aug. 2009.

[12] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, “PortLand: A scalable fault-tolerant layer 2 data center network fabric,” *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 39–50, Aug. 2009.

[13] K. A. McGreer, “Arrayed waveguide gratings for wavelength routing,” *IEEE Communications Magazine*, vol. 36, Dec. 1998.

[14] B. Li, Y. Quin, X. R. Cao, K. M. Sivaligam, and Y. Danziger, “Photonic packet switching: Architecture and performance,” *Optical Networks Magazine*, vol. 2, pp. 27–39, Jan. 2001.

[15] P. Beebe, J. M. Ballantyne, and M. F. Tung, “An introduction to mems optical switches.” `https://courses.cit.cornell.edu/engrwords/final_reports/Tung_MF_issue_1.pdf`, Dec. 2001.

[16] Y. Ohsita and M. Murata, “Data center network topologies using optical packet switches,” in *Proceedings of IEEE DCPerf*, pp. 57–64, June 2012.

[17] R. Urata, T. Nakahara, H. Takenouchi, T. Segawa, H. Ishikawa, A. Ohki, H. Sugiyama, S. Nishihara, and R. Takahashi, “4x4 optical packet switching of asynchronous burst optical packets with a prototype, 4x4 label processing and switching sub-system,” *Optics Express*, vol. 18, pp. 15283–15288, July 2010.

[18] H. J. Chao and K. Xi, “Bufferless optical clos switches for data centers,” in *Proceedings of OFC*, Mar. 2011.

[19] K. Xi, Y. H. Kao, M. Yang, and H. J. Chao, “Petabit optical switch for data center networks.” Technical Report, Polytechnic Institute of New York University, `http://eeweb.poly.edu/~chao/publications/petasw.pdf`.

[20] M. Kodialam, T. V. Lakshman, and S. Sengupta, “Efficient and robust routing of highly variable traffic,” in *Proceedings of HotNets*, Nov. 2004.

[21] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild,” in *Proceedings of Internet Measurement Conference*, Nov. 2010.

[22] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, “Fast accurate computation of large-scale IP traffic matrices from link loads,” in *Proceedings of ACM SIGMETRICS Performance Evaluation Review*, vol. 31, pp. 206–217, June 2003.

[23] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, and M. Murata, “Gradually reconfiguring virtual network topologies based on estimated traffic matrices,” *IEEE/ACM Transactions on Networking*, vol. 18, pp. 177–189, Feb. 2010.

[24] M. Zhang, C. Yi, B. Liu, and B. Zhang, “GreenTE: power-aware traffic engineering,” in *Proceedings of ICNP*, pp. 21–30, Oct. 2010.

[25] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, “Proteus: a topology malleable data center network,” in *Proceedings of ACM SIGCOMM Workshop on Hot Topics in Networks*, pp. 8–13, Oct. 2010.

[26] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in *Proceedings of ACM SIGCOMM Computer Communication Review*, pp. 339–350, Oct. 2010.

[27] T. Benson, A. Anand, A. Akella, and M. Zhang, “MicroTE: Fine Grained Traffic Engineering for Data Centers,” in *Proceedings of ACM CoNEXT*, pp. 1–12, Dec. 2011.

[28] Y. Tarutani, Y. Ohsita, and M. Murata, “A virtual network to achieve low energy consumption in optical large-scale datacenter,” in *Proceedings of IEEE International Conference on Communication Systems (ICCS 2012)*, Nov. 2012.

[29] Y. Ohsita and M. Murata, “Data center network topologies using optical packet switches,” in *Proceedings of DCPerf*, pp. 57–64, June 2012.