A Virtual Network to Achieve Low Energy Consumption in Optical Large-scale Datacenter

Yuya Tarutani, Yuichi Ohsita, and Masayuki Murata
Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
{y-tarutn, y-ohsita, murata}@ist.osaka-u.ac.jp

***Abstract*—A data center network should provide communication with sufficiently large bandwidth and small delay between all servers. On the other hand, energy consumption of the data center network should be minimized. To satisfy all of the above requirements, in this paper, we introduce the virtual network configured over the data center network constructed of the optical cross connects (OXCs) and the electronic switches. We design the virtual network topology (VNT) so as to achieve sufficiently large bandwidth and small delay with small energy consumption. To calculate the suitable VNT in a short period, we propose the topology called *Generalized Flattened Butterfly* and a method to set the parameters so as to suit the current condition. In our evaluation, we clarify that our method achieves the sufficient bandwidth and the target maximum number of hops between *top-of rack(ToR)* switches with small energy consumption.**

***Index Terms*—Data Center; Energy Consumption; Virtual Network Topology; Optical Network;**

I. INTRODUCTION

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 data center, servers handle a large amount of data by communicating with each other.

The data center network should provide communication with large bandwidth and small delay between all servers so that the data center can handle a large amount of data efficiently. The lack of bandwidth or large delay between servers may prevent the communication between servers, and degrade the performance of the application of the data center handling a large amount of data. However, the traditional data center network, which is constructed as a tree topology, cannot provide communication with sufficiently large bandwidth and small delay between servers, because 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.

On the other hand, energy consumption is another problem in a data center network. Energy consumption of the data center network occupies a non-negligible fraction of the total energy consumption in the data center [1], and becomes large as the size of the network increases. Thus, to reduce energy consumption of a data center, the energy consumption of the data center network should be reduced.

There are many researches to construct a data center network topology (e.g., [2, 3]). However, no single network topology can achieve large bandwidth and small delay with small energy consumption, and the suitable network topology depends on the application running in the data center and the current demands, which may change frequently. Heller et al. [4] has proposed a method to shut down the port of switches in the multi-root tree based on the current demands of the data center. However, this method cannot satisfy the requirements that cannot be achieved by the static base topology, because this method only activates the subsets of switches and links of the static base topology. Thus, a method to reconfigure the network topology more flexibly is required to keep the requirements satisfied with small energy consumption.

One approach to enable flexible reconfiguration of topologies is to configure the virtual network over the data center network constructed of the optical cross connects (OXCs) and the electronic switches. In this network, the core of the data center network is constructed by using the OXCs and optical fibers. We deploy an electronic switch called *top-of rack (ToR) switches* in each server rack, and all servers in a server rack are connected to a ToR switch in the rack. ToR switches are connected to the core network by connecting them to OXCs. A lightpath is established between two electronic switches by configuring the OXCs along the route between the electronic switches. A set of the lightpaths forms a virtual network topology (VNT). Traffic between electronic switches is carried over the VNT.

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 VNT and shutting down the unused ports, because energy consumption of electronic switches is much larger than that of OXCs. 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 VNT.

VNT reconfiguration methods for the core network of the internet service providers have been proposed[5, 6]. However, they are not suitable for the data center network, where the number of switches is significantly large and significant traffic changes occur frequently, because of large calculation time.

In this paper, we propose a method to reconfigure the VNT with a small calculation time even in a large data center. Our method calculates the suitable VNT by setting parameters of a topology so as to avoid large calculation time. As the topology used in the VNT configuration, we propose the topology called *Generalized Flattened Butterfly (GFB)*. We also propose a method to set the parameters so as to suit the current condition.

The rest of this paper is organized as follows. In Section II, we propose the GFB. In Section III, we propose a method to control the VNT by setting the parameters of the GFB. Then, we evaluate our method in Section IV. Finally, Section V provides a conclusion.

II. VIRTUAL NETWORK TOPOLOGIES SUITABLE TO OPTICAL DATA CENTER NETWOKS

Our VNT reconfiguration method constructs the VNT by setting parameters of a topology instead of calculating the optimal topology. In this section, as the topology used in our VNT reconfiguration, we propose a new topology called *Generalized Flattened Butterfly* (GFB).

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: $K_{\max}$
- Number of links per switch 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 change the number of required ports, the maximum number of hops and the bandwidth provided between servers.

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

$$D^{sw}(s) = \sum_{1 \le i \le K_{\max}} \left(D^{GFB}_i(s) \prod_{j=1}^{i-1} N_j \right).$$

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

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

$$L^{GFB}_k = L_k \prod_{i=1}^{k-1} N_i. \qquad (1)$$

Step I-2 If L^{GFB}_k is larger than $(N_k - 1)$, connect all layer-$k-1$ GFB pairs. Otherwise, construct a ring topology by connecting the GFBs having the nearest ID.

Step I-3 Calculate the number of the residual links L'^{GFB}_k which can be used to connect one layer-$k-1$ GFB to the other layer-$k-1$ GFBs by

$$L'^{GFB}_k = L^{GFB}_k - \bar{L}^{GFB}_k \qquad (2)$$

where $\bar{L}^{GFB}_k$ is the number of links per layer-$k-1$ GFB constructed at Steps I-2.

Step I-4 Check whether layer-$k-1$ GFBs have residual links to be used connect layer-$k-1$ GFBs. If yes, connect the GFB of ID $D^{GFB}_{k-1}(a)$ to the GFB of ID $D^{GFB}_{k-1}(b)$ which the following equation is satisfied.

$$D^{GFB}_{k-1}(b) = (D^{GFB}_{k-1}(a) + \lceil p_k \rceil + C \lfloor p_k \rfloor) \bmod N_k \qquad (3)$$

where C is an integer value, and p_k is the variable that indicates the interval of the IDs of the layer-$k-1$ GFBs connected to the same layer-$k-1$ GFBs. p_k is calculated by

$$p_k = \frac{N_k}{L'^{gfb}_k + 1}. \qquad (4)$$

By connecting the layer-$k-1$ GFBs by using the condition of Eq. 4, we can avoid large number of hops between layer-$k-1$ GFBs. In addition, the links constructed between layer-$k-1$ GFBs play the same role. Thus, the number of flows passing each link depends only on the layer of the GFBs the link connects, and is easily calculated, which is discussed in Section II-2.

b) 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 $D^{GFB}_{k-1}(a)$ is connected to the GFB of ID $D^{GFB}_{k-1}(b)$ when the following condition is satisfied.

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

where C is a integer value, $n_{D^{gfb}_{k-1}(a)}$ is the number of switches in the GFB of ID $D^{gfb}_{k-1}(a)$, and $l_{(D^{gfb}_{k-1}(a), D^{gfb}_{k-1}(b))}$ is the number of links to be constructed between GFBs of IDs $D^{gfb}_{k-1}(a)$ and $D^{gfb}_{k-1}(b)$. 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.

2) 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.

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

$$H_k = (h_k + 1) H_{k-1} + h_k \qquad (5)$$

where h_k is the largest number of links between layer-$k-1$ GFBs passed by the traffic between layer-$k-1$ GFBs.

First, we calculate h_k. If all layer-$k-1$ GFBs are directly connected, $h_k = 1$. In other cases, we calculate h_k by calculating the largest number of links between layer-$k-1$ GFBs passed by the traffic from the source layer-$k-1$ GFB whose ID is 0, because all GFBs play the same role. From the viewpoint of the source GFB, the topology constructed of layer-$k-1$ GFBs is the ring topology where some shortcut links are added directly from the source GFB. To calculate h_k, we divide the set of the GFBs, which are not directly connected to the source GFB, into groups so that the short cut links from the source GFB become the border of the group. m_j denotes the set of switched within the j-th group, and M denotes the set of the groups. Based on the steps to construct the connection between layer-$k-1$ GFBs, $|m_j|$ is calculated by

$$m_j = \begin{cases} \lceil p_k \rceil - 3 & (j = 1 or |M|) \\ \lfloor p_k \rfloor - 2 & (Otherwise) \end{cases} . \tag{6}$$

The GFBs included in each group, the source GFB and the GFBs directly connected to the source GFB form a ring topology. Thus, the maximum number of links passed by the traffic from the source GFB or the GFB belonging to the group m_j is obtained by $\left\lceil \frac{|m_j|+2}{2} \right\rceil$. Since at least one group includes the GFB whose number of hops from the source GFB is the largest, h_k is the maximum of $\left\lceil \frac{|m_j|+2}{2} \right\rceil$ among all groups. That is,

$$h_k = \begin{cases} 1 & (L_k^{GFB} \geq (N_k - 1)) \\ \lceil \frac{p_k}{2} \rceil & (L_k^{GFB} < (N_k - 1) \text{ and } L_k'^{GFB} \leq 1) \\ \lfloor \frac{p_k}{2} \rfloor + 1 & (Otherwise) \end{cases} . \tag{7}$$

b) Number of Flows through a Link: We calculate the number of flows passing each link when one flow is generated between each switch pair. In the GFB, all links constructed between layer-$k-1$ GFBs, play the same role. Thus, the number of flows passing a link depends only on the layer of the GFBs the link connects. X_k denotes the number of flows passing the link between layer-$k-1$ GFBs.

The flows between layer-$k-1$ GFBs are balanced among all links between layer-$k-1$ GFBs. Thus, X_k is the sum of the number of flows passing the links between layer-$k-1$ GFBs divided by the number of links. The sum of the number of flows passing the links between layer-$k-1$ GFBs is calculated by the product of the number of flows between each layer-$k-1$ GFB pair and the sum of the number of layer-$k-1$ GFB pairs whose flow passes links between layer-$k-1$ GFBs. That is,

$$X_k = \frac{F_k \sum_{i=1}^{h_k} i s_k(i)}{L_k \prod_{i=1}^{k} N_i} \tag{8}$$

where F_k is the number of flows between each layer-$k-1$ GFB pair, and $s_k(i)$ is the number of layer-$k-1$ GFB pairs whose flow passes i links between layer-$k-1$ GFBs. Thus, X_k is obtained by calculating $s_k(i)$ and F_k.

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

$$s_k(1) = \begin{cases} N_k(N_k - 1) & (L_k^{GFB} \geq (N_k - 1)) \\ N_k L_k \prod_{i=1}^{k-1} N_i & (\text{otherwise}) \end{cases} . \tag{9}$$

$s_k(i)$ 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

$$s_k(i) = N_k \sum_{m_j \in M} U_{(k,m_j)}(i). \tag{10}$$

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,

$$U_{(k,m_j)}(i) = \begin{cases} 0 & \left(i > \left\lceil \frac{m_j+2}{2} \right\rceil\right) \\ 1 & \left(i = \left\lceil \frac{m_j+2}{2} \right\rceil \text{ and } |m_j| \text{ is odd}\right) \\ 2 & (Otherwise) \end{cases} . \tag{11}$$

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

To make the calculation simple, in this paper, we assume that at least one link is constructed between all layer-i GFB pairs for $i \geq 1$, because we can easily connect all layer-i GFB pairs directly if $i \geq 1$ even if $L_i = 1$ since each layer-i GFB can use $L_k \prod_{i=1}^{k-1} N_i$ links. By this assumption, no GFBs relay the traffic whose source and destination switch are not included in the GFB. Thu, $F_k^{s \to d}$ is calculated by

$$F_k^{s \to d} = f_k^{s \to s \to d \to d} + \sum_{n \in G} f_k^{n \to s \to d \to d} + \sum_{n \in G} f_k^{s \to s \to d \to n}, \tag{12}$$

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,

$$f_k^{s \to s \to d \to d} = \prod_{i=1}^{k-1} (N_i)^2. \tag{13}$$

$\sum_{n \in G} f_k^{s \to s \to d \to n}$ 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, $\sum_{n \in G} f_k^{n \to s \to d \to d}$ 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.

$$\sum_{n \in G} f_k^{s \to s \to d \to n} = \frac{(\prod_{i=1}^{k-1} N_i)(\prod_{i=1}^{K_{\max}} N_i - \prod_{i=1}^{k} N_i)}{N_k}. \tag{14}$$

Similarly, F_k^{inward} is calculated by

$$\sum_{n \in G} f_k^{n \to s \to d \to d} = \frac{(\prod_{i=1}^{k-1} N_i)(\prod_{i=1}^{K_{\max}} N_i - \prod_{i=1}^{k} N_i)}{N_k}. \tag{15}$$

III. VIRTUAL NETWORK TOPOLOGY CONTROL TO ACHIEVE LOW ENERGY CONSUMPTION

A. Outline

In this section, we propose the method to set parameters of the GFB 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 VNT that can accommodate the current traffic demands between servers. However, in a data center, traffic may change within a second [7]. Thus, even if the VNT is optimized for the current traffic demands, the VNT may be required to be reconfigured every second. However, too frequent reconfiguration may cause significant packet reordering or large jitters between servers, which may degrade the throughput of the communication between servers.

In our method, the traffic changes in a short period are handled by the load balancing [8] over the VNT. And, we design the VNT so as to achieve sufficiently large bandwidth and small delay with small energy consumption, considering the load balancing.

In this paper, we use one of the load balancing technique called *Valiant Load Balancing (VLB)* [8]. In the VLB, we select the intermediate switches randomly regardless of the destination to avoid the concentration of traffic on certain links even when traffic amount of a certain switch pair is large. Then, traffic is sent from the source switch to the intermediate switch and from the intermediate switch to the destination switch. By applying the VLB, the amount of traffic between each ToR switch pair T is calculated by the following equation.

$$T \le \frac{T^{SWto} + T^{SWfrom}}{N_{\mathrm{all}}}. \tag{16}$$

where 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 a link by the traffic amount between each switch pair calculated by Eq.(16).

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

B. Topology Control to Satisfy the Requirements

In this subsection, we propose 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. (5) 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.

$$2^{K_{\max}} - 1 \le H_{max}$$

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 parahraphs.

1) 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 our parameter settings, N_k is set to $\prod_{i=1}^{k-1} N_i + 1$ for $1 < k < K_{\max}$ so as to make h_k is 1 even when $L_k = 1$.

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

$$N_{K_{\max}} = \left\lceil \frac{N_{\mathrm{all}}}{\prod_{i=1}^{k-1} N_i} \right\rceil. \tag{17}$$

In this paper, we consider the cases that $h_{K_{\max}}$ is 1. To make $h_{K_{\max}}$ 1, $L_{K_{\max}}$ should satisfy the following equation.

$$L_{K_{\max}} = \left\lceil \frac{N_{K_{\max}}}{\prod_{i=1}^{k-1} N_i} \right\rceil. \tag{18}$$

To make the maximum number of hops less than H_{max}, h_1 must satisfy the following conditions, according to Eq.(7).

$$h_1 \le \left\lceil \frac{H_{\max} + 1}{2^{K-1}} - 1 \right\rceil. \tag{19}$$

To satisfy Eq.(19), L_1 should satisfy the following equation.

$$L_1 = \begin{cases} N_1 - 1 & (h_1 = 1) \\ 2 & (h_1 \ge \lfloor \frac{N_1}{2} \rfloor) \\ \lfloor \frac{N_1}{2h_1} + 1 \rfloor & (Otherwise) \end{cases}. \tag{20}$$

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 $\sum_{1 \le k \le K} L_k$. Since $\sum_{1 \le k \le K} L_k$ is the convex function of N_1, we find the N_1 that minimizes $\sum_{1 \le k \le K_{\max}} L_k$ by incrementing N_1 as long as $\sum_{1 \le k \le K_{\max}} L_k$ decreases.

2) 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.

$$TX_k \le B \tag{21}$$

where B is the bandwidth of one link, and T is caluculated by Eq. 16. If Eq. (21) is not satisfied, we add L_k until Eq. (21) is satisfied.

IV. EVALUATION

We investigate the number of ports of ToR switches required to achieve the requirements in the GFB by comparing the existing data center network topologies; FatTree, Torus, Switch-based DCell [9] and Flattened Butterfly [3]. In this evaluation, unlike the FatTree topology proposed by Al-Fares et al. [2], 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, since powering up additional switches consumes more energy. In our evaluation, the parameters of the GFB are set by the steps described in Section III, and the parameters of the other topologies 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 comparison, all topologies include 420 ToR switches, and the bandwidth of one link is set to 10 Gbps. We assume that the number of wavelengths on optical fibers is sufficient.

First, we investigate the number of required port when the amount of traffic required to be accommodated is changed. Figure 1 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 used ports per ToR switch required to satisfy the requirement.

As shown in this figure, our method uses the smallest number of ports of ToR switches to accommodate traffic regardless of the amount of traffic, while other topologies require a large number of ports, or cannot accommodate the required amount of traffic with any parameter settings. 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 used port of ToR switch required to achieve the requirements of the acceptable maximum number of hops. In this comparison, we assume that the capacity of each virtual link is sufficient. Figure 2 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, 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 make the maximum number of hops less than the required value.

V. CONCLUSION

In this paper, we introduced the virtual network configured over the data center network constructed of the OXCs and the electronic switches, and proposed a method to reconfigure the VNT in a short period of time by setting parameters of the topology. Through numerical evaluations, we clarified that our method constructs the topology satisfying the requirements with small energy consumption.

One of our future research topics is a method to control the VNT that also considers the structure of physical topology so as to reduce the energy consumption more.

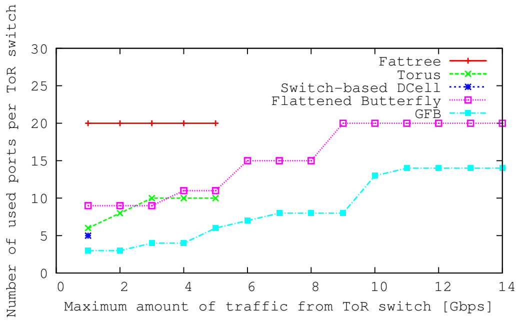

Fig. 1. Number of virtual links required to accommodate the traffic from ToR switches

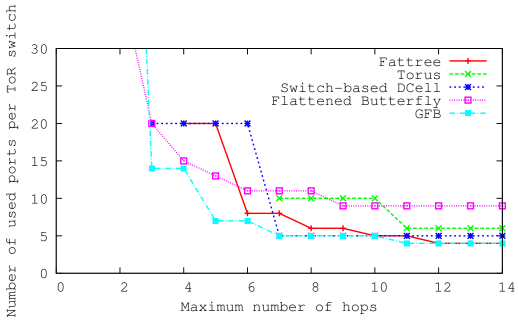

Fig. 2. Number of virtual links required to make the maximum number of hops less than the target value

ACKNOWLEDGMENTS

This work was partly supported by the National Institute of Information and Communications Technology (NICT).

REFERENCES

[1] 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.

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

[3] J. Kim, W. Dally, and D. Abts, “Flattened butterfly: a cost-efficient topology for high-radix networks,” in *Proceedings of ACM SIGARCH Computer Architecture News*, vol. 35, pp. 126–137, June 2007.

[4] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “ElasticTree: saving energy in data center networks,” in *Proceedings of USENIX*, pp. 17–32, Apr. 2010.

[5] 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.

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

[7] 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.

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

[9] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, “DCell: a scalable and fault-tolerant network structure for data centers,” in *Proceedings of ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 75–86, Aug. 2008.