Order/Radix Problem: Towards Low End-to-End Latency Interconnection Networks

Ryota Yasudo*, Michihiro Koibuchi†, Koji Nakano‡, Hiroki Matsutani*, Hideharu Amano*

*Keio University
Hiyoshi 3–14–1, Kohoku-ku,
Yokohama, JAPAN 223–8522
yasudo@am.ics.keio.ac.jp

†National Institute of Informatics
Hitotsubashi 2–1–2, Chiyoda-ku,
Tokyo, JAPAN 101–8430
koibuchi@nii.ac.jp

‡Hiroshima University
Kagamiyama 1–4–1, Higashihiroshima-shi,
Hiroshima, JAPAN 739–8527
nakano@cs.hiroshima-u.ac.jp

***Abstract*—We introduce a novel graph called a host-switch graph, which consists of host vertices and switch vertices. Using host-switch graphs, we formulate a graph problem called an order/radix problem (ORP) for designing low end-to-end latency interconnection networks. Our focus is on reducing the host-to-host average shortest path length (h-ASPL), since the shortest path length between hosts in a host-switch graph corresponds to the end-to-end latency of a network. We hence define ORP as follows: given order (the number of hosts) and radix (the number of ports per switch), find a host-switch graph with the minimum h-ASPL. We demonstrate that the optimal number of switches can mathematically be predicted. On the basis of the prediction, we carry out a randomized algorithm to find a host-switch graph with the minimum h-ASPL. Interestingly, our solutions include a host-switch graph such that switches have the different number of hosts. We then apply host-switch graphs to interconnection networks and evaluate them practically. As compared with the three conventional interconnection networks (the torus, the dragonfly, and the fat-tree), we demonstrate that our networks provide higher performance while the number of switches can decrease.**

***Index Terms*—Network topology, interconnection network, graph theory, average shortest path length, optimization.**

1. Introduction

Graph theory is not merely mathematical study; it is useful for designing networks. In the design of networks for computer systems such as telecommunications and multiprocessors, there are certain requirements and limitations. In particular, requirements include the number of nodes, and limitations include the degrees and the diameter. Hence the three parameters above have been studied in graph theory. A classical problem for such studies is the degree/diameter problem (DDP). DDP is the problem of finding the largest number of vertices in a graph of given maximum degree Δ and diameter D. The known upper bound—called the Moore bound [1]—on the number of vertices of an undirected graph is $1+\Delta\sum_{i=0}^{D-1}(\Delta-1)^i$. Near-optimal/optimal solutions of DDP are applied to topologies of interconnection networks [2], [3].

However, DDP solutions may not directly usable for building network topologies in supercomputers and high-end data centers for two reasons. First, DDP requires the specific number of vertices and the degree, and hence we cannot meet the technical requirement such as the number of nodes and the number of links. To cover this shortcoming, we can consider the order/degree problem (ODP) instead of DDP. Although less attention is given to ODP as compared with DDP, ODP is recently studied by designers of interconnection networks [4]. For ODP, we can obtain the lower bound on the diameter as with the Moore bound of DDP.

Second, in conventional graph theory, one kind of vertex is considered on a graph, though two types of nodes—hosts (end points) and switches—exist in typical interconnection networks. Hence the mapping between vertices and physical nodes is not obvious; if we regard vertices as switches, we have no information for hosts. Because the mapping strongly affects the network performance (we show this in Section 5), this is a serious issue. This shortcoming cannot be covered even though we tackle ODP; we should radically change both a model of interconnection networks and a graph problem.

We therefore introduce a novel graph called a *host-switch graph* and formulate a graph problem called an *order/radix problem* for designing low end-to-end latency interconnection networks. A host-switch graph consists of two types of vertices, hosts and switches. Our focus is on reducing the host-to-host average shortest path length (*h-ASPL*), since the shortest path length between hosts in the host-switch graph corresponds to the end-to-end latency of an interconnection network. The main technical requirement is the number of hosts and the number of ports per switch, and thus they become constraint conditions. Note that, most importantly, the number of switches is unlimited. One might think the more switches provide the lower h-ASPL, but it is not always true. We show the optimal number of switches can be predicted from the Moore bound in Section 5.

The rest of the paper is organized as follows. In Section 2 we describe related work. In Section 3 we define the host-switch graph and the order/radix problem. In Section 4 we provide the lower bound on the diameter and the h-ASPL of host-switch graphs. In Section 5 we present a randomized algorithm to find host-switch graphs with the low h-ASPL, and discuss the optimal number of switches. In Section 6 we compare the topologies given by our algorithm with conventional topologies used in supercomputers listed in Top500 [5]. Finally, in Section 7, we conclude the paper.

2. Related work

2.1 Graphs Inspired by Complex Networks

In the field of network science, researchers find that complex networks such as social networks provide the low diameter and ASPL. Thus, some models are proposed, e.g, a cycle plus a random matching [6], the Erdős-Rényi model (purely random graph) [7], and the Watts-Strogatz model (so-called small-world networks) [8]. Random graphs have not only low diameter/ASPL but also high bisection width [9], which is also important for interconnection networks. Random/small-world graphs are hence applied for computer systems, including high-performance computing (HPC) systems [10], data centers [11], and on-chip networks [12]. To apply such complex topologies to practical networks, physical layouts [13] and routing algorithms [14] are also studied. However, latest studies [15]–[17] demonstrate that local search algorithms enable us to construct better (i.e., closer to the optimal) graphs than naive random topologies. This paper follows these studies, but establishes different network models and optimization problems that consider both hosts and switches.

Researchers on network science also find that the degree-distributions of real-world networks tend to follow a power law. Networks follow this property are called scale-free networks. The ASPL of scale-free networks is *ultrasmall* [18], that is, the ASPL is $\mathcal{O}(\ln\ln N)$ while the ASPL of small-world networks is $\mathcal{O}(\ln N)$, where N denotes the number of vertices. Gyarmati *et al.* apply scale-free networks to data center networks [19]. When we apply scale-free networks, however, we must carefully evaluate the cost of various switches with different number of ports. In interconnection networks, technical requirements include the number of switches, and hence scale-free networks would not be appropriate for practical interconnection networks.

2.2 Topology of Interconnection Networks

The study on topologies of interconnection networks for parallel computer systems has a long history. In the 1970s, hypercubes were used in many systems such as Cosmic Cube [20]; in the 1980s, 2-D/3-D tori and meshes became a main stream due to short cables that provide high bandwidth and cost-efficiency; from the 1990s to 2000s, as the number of nodes becomes over 10 thousand, high-radix networks such as the dragonfly [21] are researched for reducing communication overhead; and now, in the 2010s, the high-radix interconnection networks are used in commercial high-performance computers [22], [23].

All of the networks above are *direct networks*, which mean networks such that a given number of hosts are connected to each switch. In addition to direct networks, *indirect networks* are also used, which mean networks such that some switches are connected with a given number of hosts while the other switches are connected with no hosts. Above all, the fat-tree [24] is widely used in parallel computer systems from generation to generation though technology for each generation is different (e.g., both CM-5 [25] in the 1980s and Tianhe-2 [26] in the 2010s use the fat-tree). In this respect, indirect networks contrast with direct networks. For this reason, the question of our interest is how we should uniformly discuss direct and indirect networks. Host-switch graphs enable us to answer this question. Moreover, host-switch graphs include new type of networks such that the number of connected hosts is different according to the switch, which correspond to neither direct nor indirect networks. Interestingly, the graph generated by our method is such a graph. We compare the network given by our graph and conventional network topologies in Section 6.

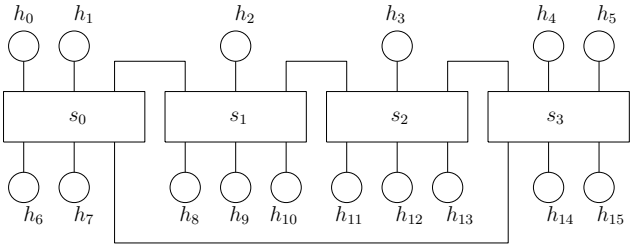


Fig. 1. An example of a host-switch graph ($n = 16, m = 4, r = 6$). A circle and a rectangle represent a host and a switch, respectively.

3. Model and Optimization Problem

3.1 A Host-Switch Graph

A *host-switch graph* is a 3-tuple $G = (H, S, E)$ with integer parameters $n \geq 3$, $m \geq 1$, and $r \geq 3$ where

- $H = \{h_0, h_1, ..., h_{n-1}\}$ is a set of n elements called *host vertices* (or simply *hosts*),
- $S = \{s_0, s_1, ..., s_{m-1}\}$ is a set of m elements called *switch vertices* (or simply *switches*), and
- $E = \{\{s_i, s_j\} \mid s_i, s_j \in S\} \bigcup \{\{h_i, s_j\} \mid (h_i \in H) \wedge (s_j \in S)\}$ is a set of elements called *edges*.

The number n of hosts is called the *order* of G. Each host must be connected with exactly one edge while each switch is connected with not greater than r edges. The number r of ports per switch is called the *radix* of G. For simplicity, we assume that the host-switch graphs dealt in this paper are connected and have no redundant switches. In other words, there exists a path along edges between any pair of two hosts, and any switches is on at least one of such paths. In Fig. 1 we illustrate an example of a host-switch graph. Throughout this paper, as with Fig. 1, a circle and a rectangle represent a host and a switch, respectively.

We can regard a host-switch graph as the undirected graph $G = (V, E)$ extended by partitioning V into H and S, and redefining elements of E. Thereby, we can think that a host-switch graph represents interconnection networks with m r-port switches, n 1-port host computers, and communication links connecting a pair of two switches or a pair of one host and one switch. In this sense, we shall regard a host-switch graph as a theoretical model of interconnection networks. Note that we cannot distinguish between hosts and switches

when we use an ordinary undirected graph as a model of interconnection networks.

3.2 Order/Radix Problem

For any two vertices $x, y \in H \bigcup S$, let $\ell(x,y)$ denote the number of edges of the shortest path between x and y. Let $D(G)$ and $A(G)$ be the *diameter* and the *host-to-host average shortest path length (h-ASPL)* of a host-switch graph G, respectively. For example, $\ell(h_0, h_{15})$ of a host-switch graph shown in Fig. 1 is 3, because the shortest path between them is (h_0, s_0, s_3, h_{15}). By using $\ell(x,y)$, we formally define the diameter and the h-ASPL as follows:

$$D(G) := \max\{\ell(h_i, h_j) \mid 0 \le i < j < n\},$$
$$A(G) := \sum_{0 \le i < j < n} \ell(h_i, h_j) / \binom{n}{2}.$$

The h-ASPL is essentially different from the average shortest path length (ASPL) of an undirected graph in that the considered path is between hosts rather than switches.

Our goal is to find a host-switch graph with the minimum h-ASPL in all host-switch graphs that connect n hosts with any number of r-port switches, because the h-ASPL determines the ideal all-to-all communication latency of interconnection networks. The diameter is also important because it determines the maximum latency; however, it is not so critical as the h-ASPL, and hence the problem focuses on minimizing the h-ASPL. Our proposed optimization problem is defined as follows: given order n and radix r, find a host-switch graph with the minimum h-ASPL. We name this problem, an *order/radix problem (ORP)*. The distinguishing characteristic of ORP is that the number of switches is variable, and hence we have a wide design space and should explore the preferred number of switches. Now, let us discuss it briefly.

If $n \le r$, then all the hosts can be connected with a switch. In this case, the h-ASPL clearly takes the minimum value for given n and r. If $r < n \le m(r-m+1)$, then all the switches can constitute a m-vertex clique. Also in this case, the h-ASPL takes the minimum value for given n and r. It is non-trivial, but we can prove it (see Appendix). If $m(r-m+1) < n$, however, there exist no trivial constructions of the host-switch graph with the minimum h-ASPL for given n and r. Because $n \gg r$ holds in the design of practical interconnection networks, it is difficult to construct a host-switch graph with the minimum h-ASPL.

4. Lower Bounds on the Diameter and h-ASPL of Host-Switch Graphs

First, we provide the lower bound on the diameter.

Theorem 1 (Lower bound on the diameter) *For any host-switch graph $G = (H, S, E)$ with fixed parameters n and r, the diameter is not less than $\lceil \log_{r-1}(n-1) \rceil + 1$.*

Proof: Consider any fixed host $h_s \in H$. The host h_s reaches exactly one switch along one edge, and hence h_s can reach at most $r-1$ hosts along two edges. In general, h_s can reach at most $(r-1)^{i-1}$ hosts along i edges. Thus, we have $n-1 \le (r-1)^{D(G)-1}$. Solving the inequality above for $D(G)$, we obtain $D(G) \ge \lceil \log_{r-1}(n-1) \rceil + 1$. ■

Next, we provide the lower bound on the h-ASPL of a host-switch graph $G = (H, S, E)$ with fixed parameters n and r. Let $A_{h_s}(G)$ and $D_{h_s}(G)$ denote a single-source h-ASPL from $h_s \in H$ and a single-source diameter from h_s, respectively. The lower bound on $A_{h_s}(G)$ is trivially the lower bound on $A(G)$. For any source host h_s, we can partition all the hosts in H into subsets $H_0, H_1, \ldots$ such that $H_i = \{h_d \in H \mid \ell(h_s, h_d) = i\}$. Similarly, we can partition all the switches in S into subsets $S_1, S_2, \ldots$ such that $S_i = \{s_d \in S \mid \ell(h_s, s_d) = i\}$. Any host-switch graph with a source host h_s clearly satisfies $H_0 = \{h_s\}$, $H_{D_{h_s}(G)} \ne \emptyset$, and $S_{D_s(G,h_s)-1} \ne \emptyset$. Thus, we have:

Lemma 1 *For any host-switch graph G with a source host h_s and a switch $s_a \in S_{D_{h_s}(G)-1}$ connected with exactly one host, there exists a host-switch graph G' with a source host h_s such that $A_{h_s}(G') < A_{h_s}(G)$.*

Proof: Let G be host-switch graph with a source host h_s and a switch $s_a \in S_{D_{h_s}(G)-1}$ connected with exactly one host h_a. By converting s_a to h_a in G, we can construct a host-switch graph G' such that $A_{h_s}(G') = A_{h_s}(G) - \frac{1}{n-1} < A_{h_s}(G)$. ■

Let a *balanced host-switch graph* G with a source host h_s denote the host-switch graph such that $H \setminus H_0 = H_{D_{h_s}(G)} \bigcup H_{D_{h_s}(G)-1}$ holds. A host-switch graph is said to be *imbalanced* if it is not a balanced host-switch graph. Using these notations, we obtain the following lemma:

Lemma 2 *Any host-switch graph G with a source host h_s such that $A_{h_s}(G) \le A_{h_s}(G')$ for any host-switch graph G' with a source host h_s is a balanced host-switch graph.*

Proof: Suppose that there exists an imbalanced host-switch graph G with a source host h_s such that $A_{h_s}(G) \le A_{h_s}(G')$ for any host-switch graph G' with a source host h_s. From Lemma 1, we can assume there exists a switch $s_a \in S_{D_{h_s}(G)-1}$ connected with at least two hosts h_a and h_b, without loss of generality. Since G is an imbalanced host-switch graph, there exists $h_c \in H_{D_{h_s}(G)-\varepsilon} (2 \le \varepsilon < D_{h_s}(G))$. We can convert h_c to a switch $s_b \in S_{D_{h_s}(G)-\varepsilon}$ connected with $h'_c \in H_{D_{h_s}(G)-\varepsilon+1}$. If $r > 3$, then we can construct G' such that $A_{h_s}(G') = A_{h_s}(G) - \frac{2(\varepsilon-1)-1}{n-1} < A_{h_s}(G)$ by reconnecting h_a and h_b with s_b. If $r = 3$, then we can reconnect only one host h_a with s_b. Here, s_a is connected with only a switch in $S_{D_{h_s}(G)-2}$ and h_c. Hence, we can construct G' such that $A_{h_s}(G') = A_{h_s}(G) - \frac{\varepsilon-1}{n-1} < A_{h_s}(G)$ by replacing s_a to h_c, a contradiction. ■

From Lemma 2, we can compute the lower bound on the h-ASPL as follows:

Theorem 2 (Lower bound on the h-ASPL) *For any host-switch graph $G = (H, S, E)$ with fixed parameters n and r, the h-ASPL is not less than*

$$\begin{cases} D^- & \text{if } n = (r-1)^{D^- - 1} + 1, \\ D^- - \alpha/(n-1) & \text{otherwise,} \end{cases}$$

where $D^- = \lceil \log_{r-1}(n-1) \rceil + 1$ *is the lower bound on the diameter of G, and* $\alpha = (r-1)^{D^- - 2} - \lceil (n-1-(r-1)^{D^- - 2})/(r-2) \rceil$).

Proof: Let G be a host-switch graph with a source host h_s. These are two cases: $n = (r-1)^{D^- - 1} + 1$ or $n \neq (r-1)^{D^- - 1} + 1$.

Case 1: $n = (r-1)^{D^- - 1} + 1$

In this case $|H \setminus H_0| = (r-1)^{D^- - 1}$ holds. From Lemma 2, a host-switch graph with the minimum $A_{h_s}(G)$ satisfies $H \setminus H_0 = H_{D^-}$. Consequently $A_{h_s}(G)$ is equal to D^-.

Case 2: $n \neq (r-1)^{D^- - 1} + 1$

In this case $|H \setminus H_0| < (r-1)^{D^- - 1}$ holds. From Lemma 2, a host-switch graph with the minimum $A_{h_s}(G)$ satisfies $H \setminus H_0 = H_{D^-} \bigcup H_{D^- - 1}$. Consequently $A_{h_s}(G)$ is equal to $D^- - |H_{D^- - 1}|/(n-1)$.

To solve for the value of $|H_{D^- - 1}|$, consider a host-switch graph $G' = (H', S', E')$ such that $H' \setminus H'_0 = H'_{D^- - 1}$ and $|H'_{D^- - 1}| = (r-1)^{D^- - 2}$. Let us convert G' to G. To this end, we should convert $\lceil (n-1-(r-1)^{D^- - 2})/(r-2) \rceil$ hosts to switches, because we must connect additional $n - (r-1)^{D^- - 2}$ hosts and the number of connectable hosts increases by $(r-2)$ if we convert a hosts to a switch. Consequently $|H_{D^- - 1}|$ is equal to $(r-1)^{D^- - 2} - \lceil (n-1-(r-1)^{D^- - 2})/(r-2) \rceil$). This value is equal to α of the given statement. ■

5. Randomized Algorithm for Order/Radix Problem

In this section we present a randomized algorithm to find a host-switch graph with the low h-ASPL and then discuss the results.

5.1 Swap Operation: Local Search Restricted to Regular Host-Switch Graphs

We shall begin with a simple algorithm that can be applied only for a host-switch graph such that any switch in S has the fixed number k of neighbor switches and $p-k$ hosts, respectively. We call such a host-switch graph a *k-regular host-switch graph* (or simply *regular host-switch graph*). We can convert a k-regular host-switch graph $G = (H, S, E)$ to k-regular graph $G' = (V, E')$ by removing all the host-switch edges and letting $S \to V$. In general, let G' denote G converted as stated above, and then the diameter and the h-ASPL of a k-regular host-switch graph G are easily obtained from these of G'.

Let us consider a k-regular host-switch graph G. We can calculate $A(G)$ by using $A(G')$, as follows:

$$A(G) = \frac{(n/m)^2 \binom{m}{2} (A(G')+2) + 2\binom{n/m}{2} m}{\binom{n}{2}} = \frac{A(G')(mn-n)}{mn-m} + 2. \quad (1)$$

Similarly, the lower bound on the h-ASPL of the k-regular host-switch graph with fixed parameter n, m, and r can be found by using the Moore bound [1], the known lower bound on the ASPL of a K-regular graph with N vertices. Using the Moore bound, say $M(N,K)$, we can calculate $A(G)$ as follows.

$$A(G) \geq \frac{M(m, p-n/m)(mn-n)}{mn-m} + 2. \quad (2)$$

From the above, we can use a randomized algorithm similar to prior research that searches an undirected regular graph with a low ASPL [15]–[17]. As with the previous work, we can use a local search algorithm where a neighbor solution is given by the *swap operation* (Fig. 2), which converts $\{s_a, s_b\}, \{s_c, s_d\} \in E$ to $\{s_a, s_d\}, \{s_b, s_c\}$. We adopt simulated annealing (SA) to escape from a local solution.

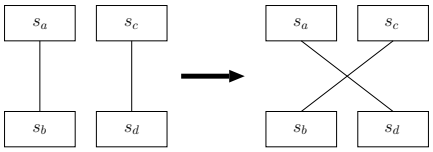


Fig. 2. Swap operation.

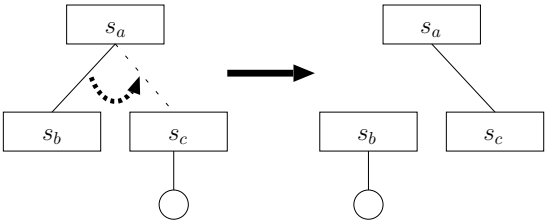


Fig. 3. Swing operation denoted as swing(s_a, s_b, s_c).

5.2 Swing Operation: Local Search for Any Host-Switch Graph

We extend the algorithm above so that it can change endpoints of host-switch edges as well as those of switch-switch edges. The extended algorithm is based on a new operation called a *swing operation* (Fig. 3). The swing operation converts $\{s_a, s_b\}, \{s_c, h_i\} \in E$ to $\{s_a, s_c\}, \{s_b, h_i\}$, and hence it can change host-switch edges. In other words, this operation increments k_b and decrements k_c. Let swing(s_a, s_b, s_c) denote the swing operation.

As stated above, the swap operation never changes host-switch edges, and contrariwise the swing operation always changes host-switch edges. Thus we should combine them to obtain good solutions. To this end, we introduce a *2-neighbor swing operation* (Fig. 4). For simplicity, hosts are omitted in the figure. This operation has the following four steps:

Step 1: Operate swing(s_a, s_b, s_c) and evaluate the solution, called the *1-neighbor solution*.

Step 2: If the 1-neighbor solution is accepted, then move to the 1-neighbor solution and the operation ends. Otherwise, go to the next step.

Step 3: Operate swing(s_d, s_c, s_b) and evaluate the solution, called the *2-neighbor solution.*

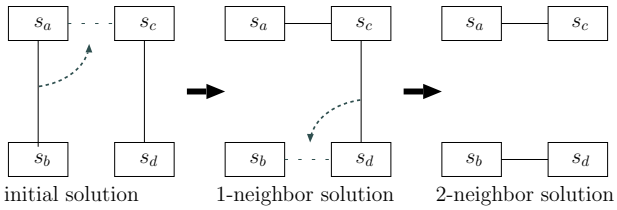


Fig. 4. 2-neighbor swing operation (hosts are omitted).

Step 4: If the 2-neighbor solution is accepted, then move to the 2-neighbor solution. Otherwise, the initial solution holds. Consequently, this operation contains both of the swap operation (if the 2-neighbor solution is accepted) and the swing operation (if the 1-neighbor solution is accepted).

5. 3 Discussion about Optimal Number of Switches

We carry out SAs with the swap operation and the 2-neighbor swing operation, and compare their results with the lower bound given by Theorem 2 and the Moore bound. We obtain the results for $n = 128, 256, 512, 1024$ and $r = 12, 24$, and show typical results among them in Fig. 5. Here, the Moore bound is calculated by Formula 2; however, n/m must be an integer, and thus the Moore bound of a host-switch graph has a value for specific pairs of n, m, and r. We hence extend the Moore bound so that the degree can be a rational number, not only an integer. We call it the *continuous Moore bound*. In Fig. 7 we show the difference between the Moore bound and the continuous Moore bound in the case of $n = 1024$ and $r = 24$.

Only in the case of $n = 128$ and $r = 24$ (Fig. 6a), the h-ASPL can be less than 3. This is because the switches can constitute a clique only in this case, as described in Section 3. 2. That is, $m \le n \le m(r-m+1)$ holds when $m = 8$. When $m \approx 8$, the optimized host-switch graph is similar to a host-switch graph such that the switches constitute a clique to obtain the low h-ASPL. In the cases of other pairs of n and r, $n \gg m(r-m+1)$ holds for any m, and consequently the h-ASPLs exceed 3.

In Fig. 5, a dotted line shows m such that the continuous Moore bound takes the minimum number. The important thing is that this value of m accords the value of m such that the h-ASPL of the optimized host-switch graph takes the minimum value in all the cases. Let m_{opt} and A_{opt} denote this value of m and the h-ASPL when $m = m_{\text{opt}}$. In Fig. 6 we show the distribution of the number of connected hosts of a switch, which we call the *host distribution*. Interestingly, the obtained graph includes switches that have different number of hosts. This corresponds to neither conventional direct nor indirect networks.

Other phenomenon of interest is that, when $m < m_{\text{opt}}$ or $m > m_{\text{opt}}$, the h-ASPL of a regular host-switch graph significantly exceeds A_{opt}, though that of a non-regular host-switch graph slightly exceeds A_{opt}. Let us discuss each case.

Case 1: $m > m_{\text{opt}}$

In this case, a non-regular host-switch graph can include unused switches that no shortest path between hosts includes. In the case of $(n, m, r) = (1024, 1024, 24)$ shown in Fig. 6c, the host distribution becomes as shown in Fig. 8. The figure illustrates over 70% switches are connected to no hosts. This indicates that the network includes otiose switches. A regular host-switch graph cannot contain such unused switches and all the switches must be connected with hosts.

Case 2: $m < m_{\text{opt}}$

In this case, only a host-switch graph with small number of switches can be constructed. Hence, when a host-switch graph is regular, the degree becomes too small and consequently the h-ASPL drastically increases. When a host-switch graph is not regular, however, a tree-like graph in which only a few switches exist can be constructed. That is why the h-ASPL can be less than the continuous Moore bound.

From the above, we have the essential observation about the preferred number of switches: for fixed n and r, we should construct a host-switch graph with m switches such that the continuous Moore bound takes the minimum value. Hence we should carry out the randomized algorithm only with this m. Accordingly, our proposed topology is generated as follows. First, for fixed n and r, we set m so that the continuous Moore bound takes the minimum value. Second, we carry out SA with the 2-neighbor swing operation. This method contracts with conventional optimization methods in which the number of switches is given in advance.

6. Comparisons with Conventional Topologies

6. 1 Review of Conventional Topologies

There are many conventional topologies of interest. Among them, we pick up typical topologies used in supercomputers listed in Top500 for November 2016: the torus used in Titan [27] and Sequoia [28], the fat-tree used in Tianhe-2 [26], and the dragonfly used in Cori [22] and Piz Daint [23]. We review them as a host-switch graph and compare them with our proposed host-switch graph. Note that the definitions described below are specialized for the comparisons, and there are other variations of each topology.

6. 1. 1 Torus: The *torus host-switch graph* with integer parameters dimension, say K, and base, say N, is a host-switch graph such that all the switches constitute a *K-ary N-torus*. In a K-ary N-torus, each node is identified by a K-bit base-N address, $a_{K-1}a_{K-2}...a_0$, and connected to node with addresses $a'_{K-1}a'_{K-2}...a'_0$ such that $a'_i \pm 1 \pmod N = a_i$ and $a'_j = a_j$ for $0 \le i \le K-1$ and $j \ne i$.

From the definition, the number of switches is

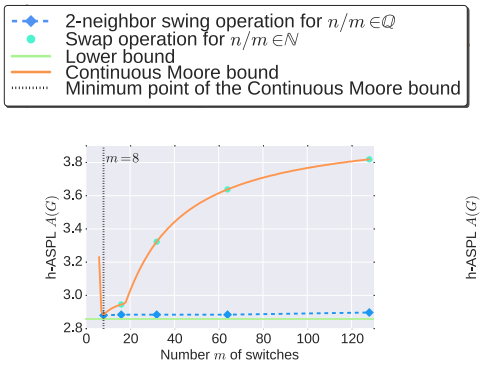
$$m = N^K. \tag{3a}$$

Each switch is connected with $2K$ other switches and can be connected with at most $r-2K$ hosts. Thus the number of hosts and the radix are

$$n \le (r-2K) \cdot N^K, \tag{3b}$$

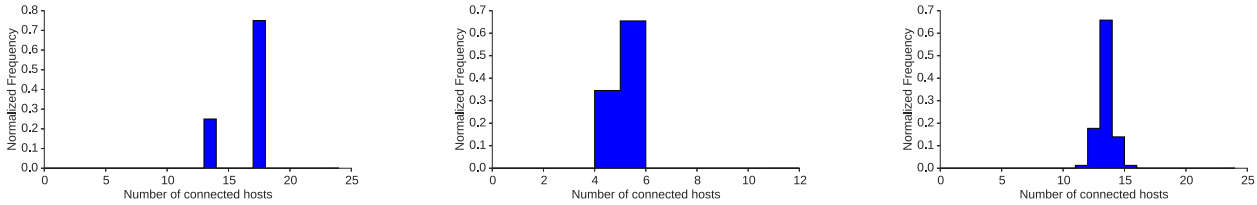
$$r > 2K. \tag{3c}$$

Note that the number of hosts connected with a switch is variable because the torus is a direct network.



(a) $n = 128, r = 24$ (b) $n = 256, r = 12$ (c) $n = 1024, r = 24$

Fig. 5. h-ASPL vs. number of switches



(a) $n = 128, r = 24$ (b) $n = 256, r = 12$ (c) $n = 1024, r = 24$

Fig. 6. Host distribution when $m = m_{\text{opt}}$

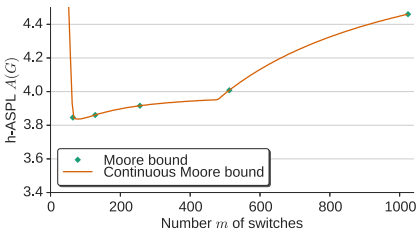


Fig. 7. Comparison between the Moore bound and the continuous Moore bound.

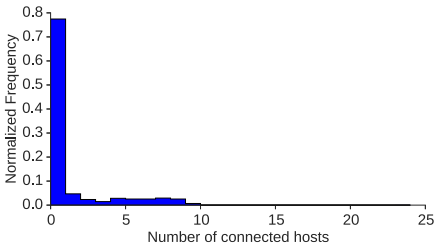


Fig. 8. Host distribution of a host-switch graph with unused switches $((n,m,r) = (1024, 1024, 24))$.

6.1.2 Dragonfly: The dragonfly, originally proposed in [21], is a hierarchical network. It can easily be modeled as a host-switch graph because it contains information of both switches and hosts. It has four parameters a, h, g, and p, which represent the numbers of switches in a group, edges of a switch connected with other groups, the total groups, and hosts connected with a switch, respectively. According to prior research [21], these parameters should satisfy $a = 2h = 2p$ to balance traffic loads, and hence we assume this equation holds. From the above, the radix is

$$r = (a-1) + h + p = 2a - 1. \tag{4a}$$

We assume $g = ah + 1$ holds, that is, there is exactly one edge between each pair of groups and the diameter and h-ASPL take the minimum values. Consequently the switches in a group constitute a a-vertex clique, and the groups also constitute a $a^2/2 + 1$-vertex clique, and the numbers of switches and hosts are

$$m = a\left(\frac{a^2}{2} + 1\right) = \frac{a^3}{2} + a, \tag{4b}$$

$$n \leq p \cdot m = \frac{a^4}{4} + \frac{a^2}{2}. \tag{4c}$$

6.1.3 Fat-Tree: There are many variations of fat-trees. In this paper, we adopt three-layer fat-tree such that the number of ports of a switch is uniform, which is a special instance of Clos network called a *K-ary fat-tree* [29]. The fat-tree can easily be modeled as a host-switch graph because it is an indirect network. The value of K corresponds the number of links per switch, and thus the radix is

$$r = K. \tag{5a}$$

A K-ary fat-tree consists of three layers: the core layer with $K^2/4$ switches, the aggregation layer with $K^2/2$ switches, and the edge layer with $K^2/2$ switches. Thus, the number of switches is

$$m = 5K^2/4. \tag{5b}$$

Each switch in the edge layer is connected with $K/2$ hosts, and thus the number of hosts is

$$n = K^3/4. \tag{5c}$$

6.2 Experimental Method

We compare the conventional topologies above with proposed topologies in terms of performance, bandwidth, power consumption, and cost breakdown. Since each conventional topology must take a specific combination of n, m, and r, we separately compare it with our proposed topology. The comparisons include three experiments below.

6.2.1 Performance evaluation: We simulate the execution of parallel applications that use Message Passing Interface (MPI) by using SIMGRID discrete event simulator (v3.15) [30]. The applications we use are NAS parallel benchmarks (version 3.3.1, MPI versions, Class A for IS and FT, and Class B for the others) [31]. To run the benchmarks, the number of processes must be the power of four, and thus we assume $n = 1024$ and the network size is set to suit it. Each host has 100 GFlops in all networks. We configure SIMGRID to utilize its built-in version of the MVAPICH2 implementation of MPI collective communications.

For each conventional topology, we construct the smallest host-switch graph such that the number of connectable hosts is 1024 or more, and we sequentially connect hosts to switches until n becomes 1024. For the proposed topology, we construct host-switch graph such that $n = 1024$ and r is the same as each conventional topology, as described in Section 5. Afterward, we sequentially connect hosts to switches in depth-first order by using backtracking.

6.2.2 Bandwidth evaluation: We evaluate bandwidth of networks by using METIS, a set of programs for partitioning graphs [32]. For each host-switch graph $G = (H, S, E)$, we partition the vertices in $V = H \cup S$ into 2-16 disjoint subsets equally so that the number c of edges such that two endpoints are in different subsets becomes minimum. Here c is defined as bandwidth. In particular, when we partition a graph into 2 subgraphs, we obtain bisection bandwidth. In general, interconnection networks with larger bisection bandwidth are better because minimum cut determines maximum possible flows through a network, according to the max-flow min-cut theorem [33].

6.2.3 Power and cost evaluation: To evaluate power consumption and cost, we design a physical floorplan which is sufficiently large to align all cabinets on a 2-D grid. We assume that each cabinet is 60 cm wide and 210 cm deep including space for the aisle, and calculate the number of cables and their lengths. If a cable length is over 100 cm, we use an optical cable. Otherwise, we use an electrical cable. We subsequently use power and cost models of Mellanox InfiniBand FDR10 switches and Mellanox InfiniBand FDR10 40Gb/s QSFP cables [2].

6.3 Results and Discussion

6.3.1 Comparison with Torus: We adopt the torus such that the dimension K is 5, which we call a 5-D torus, as with

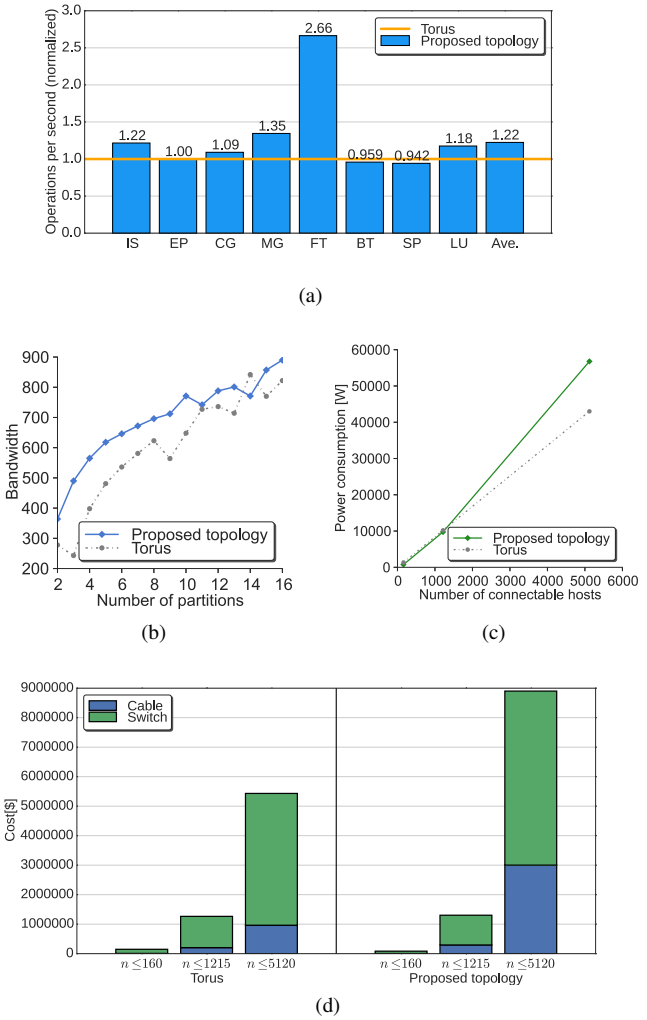


Fig. 9. Results of comparisons between 5-D torus and proposed topology: (a) Performance; (b) Bandwidth; (c) Power consumption; (d) Cost breakdown.

networks of Sequoia. From Formulae 3 we set N and r to 3 and 15, respectively, and consequently $n \leq 1215$, $m = 243$ and $r = 15$. Although we can also construct the 5-D torus such that $N = 4$ and $r = 11$, the number of host connected with a switch is only 1 in this case, and so it is impractical. The proposed topologies satisfy $n = 1024$, $m = 194$, and $r = 15$, and thus the number of switches decreases by 20%.

In Fig. 9a we show the results of the performance comparison. The proposed topology outperforms the 5-D torus by 22% on average. It achieves particularly high performance in the cases of IS (Integer Sort), FT (Fourier Transform), and MG (Multi-Grid), because they require random memory access, all-to-all communication, and long-distance communication, respectively, and thus the low h-ASPL effectively improves performance.

In Fig. 9b we show the results of the bandwidth comparison. Compared with the 5-D torus, the proposed topology increases bisection bandwidth by 31% and provides higher bandwidth regardless of the number P of partition, except for the case of $P = 14$. This indicates that the proposed topology, which

is optimized not for bandwidth, provides high bandwidth, as with random graphs [9].

In Fig. 9c we show the results of the power comparison. The proposed topology consumes less power when the number of connectable hosts is 1215 or less. Since we assume $n = 1024$ in the performance evaluation, we can say the proposed topology provides higher performance with lower power consumption compared with the 5-D torus. However, the proposed topology consumes more power when the number of connectable hosts is more than 1215. This is because we assume the torus with the fixed dimension and the fixed radix—5 and 15, respectively—and hence the cost of the torus increases only slightly as the number of connectable hosts increases. Thus, the performance of the torus would drastically degrade when the number of hosts is more than 1215. The proposed topology, on the other hand, uses more switches to reduce the h-ASPL.

In Fig. 9d we show the results of the cost comparison. They show a similar tendency as the results of power consumption. However, when the number of connectable hosts is 1215, the proposed topology requires more cost than the 5-D torus on account of cable complexity. In addition, the cable cost increases by 45%, though the switch cost decreases by 5%. Because the switch cost is dominant, the total cost increases only by 3%, which is small as compared with the performance improvement.

Overall, as compared with the 5-D torus, the proposed topology provides higher performance and bandwidth with comparable power consumption and cost when $n = 1024$. If the number of hosts becomes more than 1024, the 5-D torus requires small power consumption and cost, but it is not scalable in terms of performance.

6. 3. 2 Comparison with Dragonfly: From Formulae 4, we set a to 8, and consequently $n \leq 1056$, $m = 264$ and $r = 15$. The proposed topologies satisfy $n = 1024$, $m = 194$, and $r = 15$. Hence the number of switches decreases by 27%.

In Fig. 10a we show the results of the performance comparison. The proposed topology outperforms the dragonfly by 12% on average. These results, however, illustrate a trend different from the comparison with the 5-D torus. This is because the dragonfly provides low diameter, and the performance does not degrade even when the long-distance traffic occurs. From these results, we reconfirm that the dragonfly is an efficient topology with the specific pairs of n, m, and r, but nonetheless our proposed topology outperforms the dragonfly on average.

In Fig. 10b we show the results of the bandwidth comparison. The results of the proposed topology are the same as results in Fig. 9b because values of n, m, and r are the same. Compared with the dragonfly, the proposed topology increases bisection bandwidth by 24% and provides higher bandwidth regardless of the number of partition.

In Fig. 10c we show the results of the power comparison. The power consumption of both topologies are almost proportional to the number of connectable hosts. The radix increases as the number of connectable hosts increases in the case of the dragonfly, and hence the number of switches for proposed topology becomes lower, unlike the case of the 5-D torus.

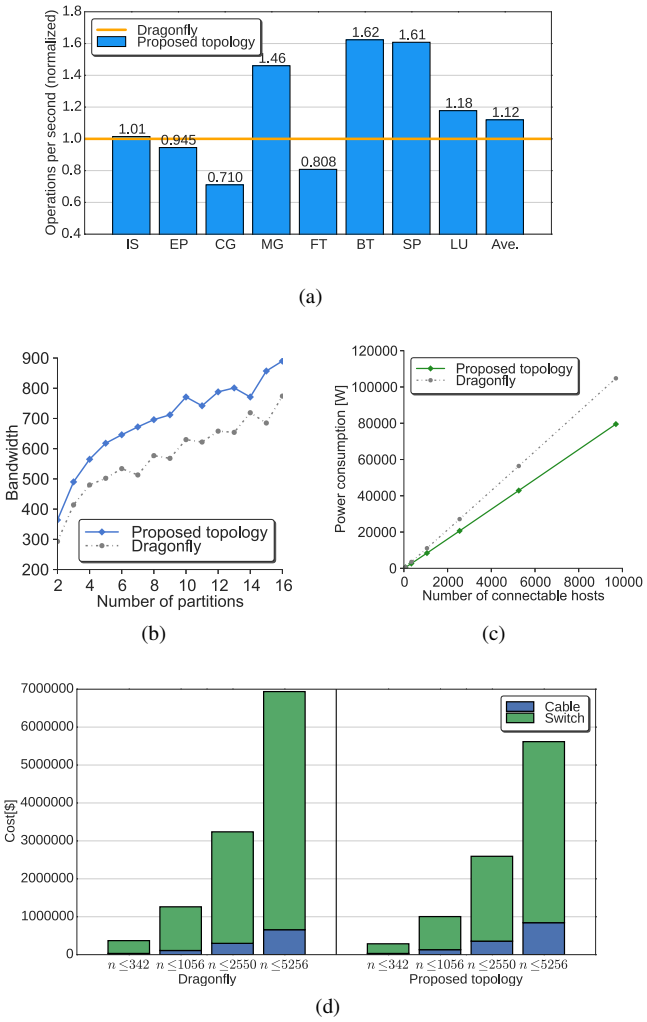


Fig. 10. Results of comparisons between dragonfly and proposed topology: (a) Performance; (b) Bandwidth; (c) Power consumption; (d) Cost breakdown.

Thus, compared with the dragonfly, the proposed topology can reduce the power consumption regardless of the number of connectable hosts.

In Fig. 10d we show the results of the cost comparison. We find the cable cost of the dragonfly is low, because the cable within a group in the dragonfly is short. However, as with the power consumption, the proposed topology can reduce cost as compared with the dragonfly regardless of the number of connectable hosts. Although the cable cost slightly increases on account of the cable complexity, its reduction rate is smaller than that of switch cost.

Overall, the proposed topology and the dragonfly have similar properties, but the proposed topology provides higher performance, higher bandwidth, lower power consumption, and lower cost. Furthermore, the proposed topology is more flexible in the sense that it can be designed for any possible combination of n, m, and r.

6. 3. 3 Comparison with Fat-Tree: From Formulae 5, we adopt a 16-ary fat-tree, and consequently $n \leq 1024$, $m = 320$ and $r = 16$. The proposed topologies satisfy $n = 1024$, $m =$

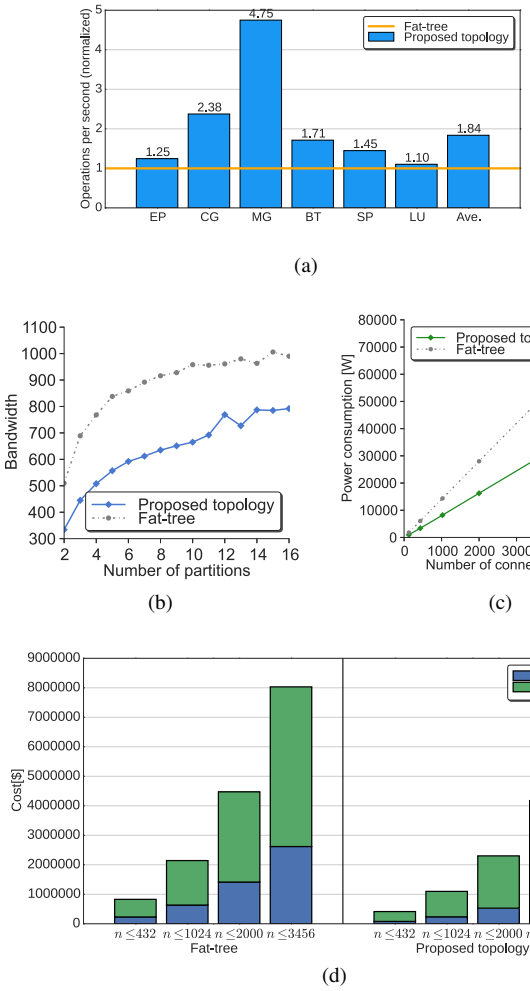


Fig. 11. Results of comparisons between fat-tree and proposed topology: (a) Performance; (b) Bandwidth; (c) Power consumption; (d) Cost breakdown.

183, and $r = 16$. Hence the number of switches decreases by 43%.

In Fig. 11a we show the results of the performance comparison (due to computational complexity, simulations for IS and FT are omitted). The proposed topology outperforms the fat-tree by 84% on average. As with the torus, the fat-tree degrades performance especially in MG, memory intensive application that requires long-distance communication. Furthermore, in the case of CG (Conjugate Gradient), the performance ratio is considerable, because CG requires irregular memory access and communication.

In Fig. 11b we show the results of the bandwidth comparison. Unlike the torus and the dragonfly, the fat-tree provides higher bandwidth as compared with the proposed topology. Bisection bandwidth is higher by 53%. This is because the fat-tree is designed with full bisection bandwidth as described in Section 6. 1. 3. Our results indicate that, even if the bisection bandwidth is high, the performance is not always high.

In Fig. 11c we show the results of the power comparison. As with the case of power comparison with the dragonfly, the power consumption of both topologies is almost proportional to the number of connectable hosts. From the figure we find the fat-tree consumes the largest power consumption among the three conventional topolgoies.

In Fig. 11d we show the results of the cost comparison. The fat-tree requires the largest cost among the three conventional topologies. Furthermore, unlike the torus and the dragonfly, the cable cost of the fat-tree is also higher than that of the proposed topology.

Overall, as compared with the fat-tree, the proposed topology drastically improves performance with lower power consumption and cost, although the bandwidth is lower. This results indicate that the h-ASPL is the important metrics for HPC as well as the bandwidth.

7. Conclusions

In this paper we have introduced a host-switch graph, a model of interconnection networks including hosts and switches. This model enables comprehensive study of interconnection networks. Our study focuses on reducing host-to-host average shortest path length (h-ASPL) and establishes a new optimization problem called the order/radix problem: given order and radix, find a host-switch graph with the minimum h-ASPL. For this problem, we show the lower bound on the h-ASPL and present a randomized algorithm. We show the optimal number of switches that provides the minimum h-ASPL, say m_{opt}, can be predicted by the Moore bound. We then proposed the topology with the m_{opt} switches.

We have compared the proposed topology with conventional topologies listed in Top500, the torus, the dragonfly, and the fat-tree, in terms of performance, bandwidth, power consumption, and cost breakdown. Our results demonstrate that, when the number of hosts is 1024, the proposed topology outperforms all of the three topologies in terms of operation per second for MPI applications by 12%–84% on average, while our topology can reduce the number of switches by 20%–43%. Thus we have successfully demonstrated that our method can directly be used for designing interconnection networks.

Acknowledgment

This work was partially supported by the JST/CREST program entitled “Research and Development on Unified Environment of Accelerated Computing and Interconnection for Post-Petascale Era” in the research area of “Development of System Software Technologies for post-Peta Scale High Performance Computing”.

References

[1] M. Miller and J. Širáň, “Moore graphs and beyond: A survey of the degree/diameter problem,” *Electronic Journal of Combinatorics*, vol. DS14, pp. 1–61, electronic only, 2005.

[2] M. Besta and T. Hoefler, “Slim fly: A cost effective low-diameter network topology,” in *Proc. of the Int’l Conf. for High Performance Computing, Networking, Storage and Analysis*, Nov. 2014, pp. 348–359.

[3] R. Mizuno and Y. Ishida, “Constructing large-scale low-latency network from small optimal networks,” in *Proc. of the Int’l Symp. on Networks-on-Chip*, Sept. 2016, pp. 1–6.

[4] “GraphGolf: the order/degree problem competition,” http://research.nii.ac.jp/graphgolf/, 2017.

[5] "Top500 supercomputer sites," https://www.top500.org/, 2017.
[6] B. Bollobás and F. R. K.Chung, "The diameter of a cycle plus a random matching," *SIAM Journal on Discrete Mathematics*, vol. 1, pp. 328–333, 1988.
[7] P. Erdős and A. Rényi, "On random graphs I," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.
[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.
[9] B. Bollobás, "The isoperimetric number of random regular graphs," *European Journal of Combinatorics*, vol. 9, pp. 241–244, 1988.
[10] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova, "A case for random shortcut topologies for HPC interconnects," in *Proc. of the Int'l Symp. on Computer Architecture*, June 2012, pp. 177–188.
[11] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Proc. of the 9th USENIX conference on Networked Systems Design and Implementation*, Apr. 2012, pp. 17:1–17:14.
[12] U. Y. Ogras and R. Marculescu, "It's a small world after all: NoC performance optimization via long-range link insertion," *IEEE Trans. on VLSI Systems*, vol. 14, pp. 693–706, July 2006.
[13] M. Koibuchi, I. Fujiwara, H. Matsutani, and H. Casanova, "Layout-conscious random topologies for HPC off-chip interconnects," in *Proc. of the Int'l Symp. on High Performance Computer Architecture*, Feb. 2013, pp. 484–495.
[14] J. Flich, T. Skeie, A. Mejía, O. Lysne, P. López, A. Robles, J. Duato, M. Koibuchi, T. Rokicki, and J. C. Sancho, "A survey and evaluation of topology-agnostic deterministic routing algorithms," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, pp. 405–425, 2012.
[15] K. Nakano, D. Takafuji, S. Fujita, H. Matsutani, I. Fujiwara, and M. Koibuchi, "Randomly optimized grid graph for low-latency interconnection networks," in *Proc. of the Int'l Conf. on Parallel Processing*, Aug. 2016, pp. 340–349.
[16] N. Shimizu and R. Mori, "Average shortest path length of graphs of diameter 3," in *Proc. of the Int'l Symp. on Networks-on-Chip*, Sept. 2016, pp. 1–6.
[17] T. Kitasuka and M. Iida, "A heuristic method of generating diameter 3 graphs for order/degree problem," in *Proc. of the Int'l Symp. on Networks-on-Chip*, Sept. 2016, pp. 1–6.
[18] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Physical Review Letters*, vol. 90, Feb. 2003.
[19] L. Gyarmati and T. A. Trinh, "Scafida: A scale-free network inspired data center architecture," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 5–12, Oct. 2010.
[20] C. L. Seitz, "The cosmic cube," *Communications of the ACM*, vol. 28, Jan. 1985.
[21] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. of the Int'l Symp. on Computer Architecture*, June 2008, pp. 77–88.
[22] K. Antypas, N. Wright, N. Cardo, A. Andrews, and M. Cordery, "Cori: a cray xc pre-exascale system for nersc," in *Proc. of cray user group conference*, May 2014.
[23] S. R. Alam, T. Athanassiadou, T. W. Robinson, G. Fourestey, A. Jocksch, L. Marsella, J.-G. Piccinali, J. Poznanovic, B. Cumming, and D. Ulmer, "First 12-cabinets cray xc30 system at cscs: Scaling and performance efficiencies of applications," in *Proc. of cray user group conference*, May 2013.
[24] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *IEEE Trans. on Computers*, vol. C-34, pp. 892–901, Oct. 1985.
[25] W. D. Hillis and L. W. Tucker, "The CM-5 connection machine: a scalable supercomputer," *Communications of the ACM*, vol. 36, Jan. 1993.
[26] X.-K. Liao, Z.-B. Pang, K.-F. Wang, Y.-T. Lu, M. Xie, J. Xia, D.-Z. Dong, and G. Suo, "High performance interconnect network for tianhe system," *Journal of Computer Science and Technology*, vol. 30, Mar. 2015.
[27] A. S. Bland, J. C. Wells, O. E. Messer, O. R. Hernandez, and J. H. Rogers, "Titan: Early experience with the cray xk6 at oak ridge national laboratory," in *Proc. of cray user group conference*, May 2012.
[28] M. Moudi and M. Othman, "The challenge of interconnect topologies to improve communication in supercomputers," in *Proc. of the Int'l Conf. on Recent Trends in Information Processing & Computing*, Dec. 2012, pp. 137–144.
[29] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. of the ACM SIGCOMM Confeence*, Aug. 2008, pp. 63–74.
[30] "SimGrid: Versatile Simulation of Distributed Systems," http://simgrid.gforge.inria.fr/, 2017.
[31] "The NASA Advanced Supercomputing (NAS) Parallel Benchmarks," http://www.nas.nasa.gov/Software/NPB/, 2017.
[32] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, pp. 359–392, Aug. 1998.
[33] P. Elias, A. Feinstein, and C. E. Shannon, "A note on maximum flow through a network," *IRE Transactions on Information Theory*, vol. 2, pp. 117–119, Dec 1956.

APPENDIX

Let a *clique host-switch graph* denote a host-switch graph such that all the switches constitute a clique. For any clique host-switch graph, each switch must be connected with exactly $m-1$ switches. Here, we prove that a host-switch graph with the lowest h-ASPL for given n and r is a clique host-switch graph. First, we have the following lemma.

Lemma 3 *A clique host-switch graph with the lowest h-ASPL has the minimum possible number of switches.*

Proof: Suppose that a clique host-switch graph with the lowest h-ASPL G has m switches while a clique host-switch graph with m' switches ($m' < m$) can be constructed. We can then reduce h-ASPL of G by removing a switch s_i and reconnect hosts connected with s_i to another switch, a contradiction. ■

Lemma 3 leads to:

Corollary 1 *Let k_i denote the number of hosts connected with s_i. A clique host-switch graph with the lowest h-ASPL satisfies $k_i \geq m$ for all i ($0 \leq i \leq m-1$).*

Here, we can derive the following theorem:

Theorem 3 *For fixed n and r, there exists a clique host-switch graph that has the lowest h-ASPL.*

Proof: Let G be a clique host-switch graph such that $k_i \geq m$ for all i ($0 \leq i \leq m-1$) and the number of switches is minimum. From Lemma 3, G is a clique host-switch graph with the lowest h-ASPL. Let G' be a host-switch graph with parameters n, m', and r, which is not a clique host-switch graph. Let us consider construct G' from G, and compare $A(G)$ and $A(G')$:

Case 1: $m' > m$

Trivially, $A(G') \geq A(G)$ holds since k_i cannot increase.

Case 2: $m' \leq m$

To increase k_i, we must cut an edge (s_i, s_j) ($i \neq j$) and reconnect a host to s_i. If we cut the edge, then the h-ASPL increases by at least $k_i \cdot k_j / \binom{n}{2}$. If we reconnect a host to s_i, then the h-ASPL decreases by at most $k_i / \binom{n}{2}$. Hence, the h-ASPL does not increase only if $k_j < 2$. From Corollary 1, G satisfies $k_j > m$, and G' satisfies $k_j < 2$ only after cutting at least $m-2$ edges connected with s_i. After cutting $m-2$ edges, however, s_i has only one edge, and hence we cannot cut an edge any more. Therefore, $A(G') \geq A(G)$ holds. ■