An Interconnection Network Exploiting Trade-off Between Routing Table Size and Path Length

Thanh-Chung Kieu, Khanh-Van Nguyen
Ha Noi University of Science and Technology
1 Dai Co Viet Road, Ha Noi, Viet Nam
kieuthanhchung@gmail.com, vannk@soict.hust.edu.vn

Nguyen T. Truong, Ikki Fujiwara, Michihiro Koibuchi
National Institute of Informatics / SOKENDAI
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
nguyen.88.tt@gmail.com, {ikki, koibuchi}@nii.ac.jp

***Abstract*—Various parallel applications require a low-latency interconnection network to achieve high performance and high scalability. Recently proposed random network topologies achieve low latency, but they require each switch to have a large number of routing table entries, e.g., larger than N for an N-node network, for implementing a minimal routing. In this study, we propose the use of a random network topology with our new routing scheme so that the required routing table size becomes small, e.g., 528 for 8192 nodes, at each switch. Our main finding is that our routing algorithm cannot always follow the minimal paths, but its average path length is still short when compared to that of existing network topologies.**

***Index Terms*—Network Topologies, small-world network, compact routing, average routing path length**

I. Introduction

Network topologies and their routing algorithms in interconnection networks have been widely studied for decades. A well-known combination is k-ary n-cube with dimension-order routing or Duato's protocol [1]. It is frequently used in supercomputers such as BlueGene/L Anton-2 [2] or Cray XT5 [3]. It requires $\frac{k \times n}{4}$ hops on average for the inter-node communication, and the diameter is as large as $\frac{k \times n}{2}$ when k is an even number. Some unique network structures, such as De-bruijn, Star, and Kautz, provide better average shortest path length (ASPL) when compared to the k-ary n-cubes.

In terms of ASPL and diameter, the random network topology is well-known to be better than the above network topologies, and its family appears in the graph catalog in the GraphGolf competition [4]. It assumes a topology-agnostic routing algorithm, e.g., up*/down* routing [5], and it requires routing tables that have $N \times C$ entries at each switch where N and C are the network size and the number of input ports of a switch, respectively.

As the number of nodes becomes large, e.g., 100,000, the required number of routing table entries will be larger than that implemented in recent switches, e.g., 48,000 in InfiniBand switch products [6] or 32,000 in Ethernet switch products.In this context, compact routing schemes aimed at finding the best trade-off between the number of routing table entries and the *stretch factor* of the routing path has been studied. The stretch factor is the worst-case ratio between the routing path length and the topological minimal distance for a source-and-destination pair. Two routing design directions/fashions have been proposed: (1) Hierarchical Routing and Addressing [6], [7] for regular topologies, and (2) the *universal compact routing* that can apply for arbitrary topologies [8]–[10] especially for Internet-like networks [11], [12].

In this study we propose to use a random network topology based on Kleinberg's small world model [13] with our hierarchical compact routing for better trade-off between the routing table size and the average path length. Our routing algorithm attempts to minimize the average path length for a given requirement on the number of routing table entries at every switch. To the best of our knowledge, this is the first work that considers this trade-off on interconnection networks for high-performance computing platforms.

The main contributions of this work are:

- We design and analyze a Hierarchical Routing and Addressing for random networks with small routing table and short average routing path length, e.g., 9.05 hops using 528 entries at each switch for a network of 8,192 switches.
- Implementation of our compact routing on the Small-World topology, called HR-SW, has a larger number of routing table entries compared to the hierarchical routing for torus topologies, but it reduces a lot on average routing path length, e.g., 43.4% shorter in a network of 8,192 switches. Comparing to the universal compact routing of Thorup and Zwick [9] for random networks, HR-SW achieves similar average path length, but it has a much smaller routing table, e.g., 30% smaller.

II. Background and Related Work

A *routing* is a mechanism that can transfer a message (packets of information) from any source nodes to any destination nodes of the network.

A. Distributed and Source Routing Implementations

There are two types of routing implementation in interconnection networks. A simple implementation is the use of a routing table at each switch (distributed routing). In this case, a packet header includes routing information, e.g., destination identifier, that is used as an index for the routing tables. This is commonly used in commercial interconnection networks, such as InfiniBand and Ethernet. Another Implementation is the source routing that packs all the path information into a packet header. It does not require routing table at every switch. In this study, we assume the distributed routing as the implementation on interconnection networks.

B. Routing Relations

Generally, routing algorithms are classified into three types of relations [1].

2379-1896/16 $31.00 © 2016 IEEE
DOI 10.1109/CANDAR.2016.41

1) The $N(\text{source}) \times N(\text{destination}) \mapsto P$ routing relation (all-at-once), where N is the node set and P is the path set.
2) The $C \times N \mapsto C$ routing relation, which considers the input port of the packet to find the output channel C.
3) The $N \times N \mapsto C$ routing relation, which only takes into account the current and destination nodes to determine the output channel.

All of the three types can be deployed using a routing table, but they differ in the number of entries. At a switch, the routing table stores the value of the relation, e.g., the identifier of the output port on the routing path, that is indexed by the destination identifier. To forward a packet, the switch has to compare the destination address of the message to all the indexes in order to choose an appropriate route. Hereafter we call this "the routing lookup function". Following the above taxonomy, the number of entries in each switch is at most $N \times N \times C$ in the first type, $N \times C$ in the second type, and N in the third type, respectively. The first routing relation can express an arbitrary routing algorithm but its routing table becomes large. By contrast the third routing relation requires the smallest number of routing table entries. In this study we focus on reducing the number of stored entries by constructing the routing schemes that require less than N entries (known as *compact routing* schemes approach). Next, we review the well-known compact routing schemes and their targeted topologies.

C. Network Topologies and Compact Routing Algorithms

Compact routing refers to the design of routing scheme that uses a small number of routing table entries (*RT*) at each node so that a small ratio between the path length of the route and the shortest path, i.e., the *stretch* of routes, is provided. For a given family of network topologies, finding the best trade-off between RT and stretch is the focus of this research area. The routing scheme that can apply on all the networks is called *universal compact routing*.

Focusing on small routing table size, Kleinrock and Kamoun proposed Hierarchical Routing and Addressing technique, which is the basis of today's Internet routing approach [7]. The main idea is that the nodes in a network are grouped into clusters, then the clusters may be grouped into clusters of clusters, and so on. Each node keeps complete information of the nodes that are close to it, e.g., in the same cluster, while stores less information of the further away nodes, e.g., stores the nodes in the other clusters by only one entry. The summary of related works in this area [11] show that the efficiency of the hierarchical approach in terms of RT-strech trade-off depends on (1) the abundance of remote nodes or (2) the regularity of the network topologies. Typically, the torus topology family has the former property, e.g., 3-D Torus with Dimension-Order Routing (DOR). Dragonfly [14], which is one of the current state-of-the-art high-radix topologies, and the b-ary tree structure are considered as the latter type of topologies. For example, Mariano et al. proposed to use the Hierarchical Routing and Addressing technique for Exascale HPC Systems that employ Flattened Butterfly, Folded-Clos, and Dragonfly topologies in [6].

By contrast, the hierarchical approach is not suitable for random network topologies because these network topologies have small average path length (sparse remote nodes) by exploiting randomness (irregularity). In this context, researchers paid attention on the universal compact routing that has a good RT-strech trade-off by finding the lower bound of the routing table size versus the stretch.

In a general view, a shortest path routing (stretch-1) requires $O(n)$ entries or $O(n \log(n))$ bits at each node, where n is the number of nodes. Below, we summarize the well-known theoretical bound of routing table size at each node with the "bits" metrics to keep the consistency with the previous work. The analysis in [8] showed that a compact routing scheme that uses only $O(n)$ bits leads to route a message with a stretch of factor at least 3. Thorup and Zwick proved that any compact routing scheme with stretch less than 5 can not archive the RT smaller than $\Omega(n^{1/2})$ [15].

Looking inside the concrete routing schemes, Cowen proposed the landmark-based algorithms [10] with stretch-3 and $\tilde{O}(n^{2/3})$ of RT[1]. Improving this landmark-based approach, Thorup and Zwick designed a new routing scheme that archives stretch-3 with only $\tilde{O}(n^{1/2})$ [9]. The main idea of this approach lays on finding a representative set of nodes as landmarks and finding the neighborhood for each remaining node (the cluster of a node). A non-landmark node keeps complete information of the nodes in its own cluster and all the landmarks. A non-landmark node also uses a landmark node to be its representative, i.e., the closest landmark in terms of graph distance. A landmark node do not have a cluster but it stores the information of all the nodes that it represents. Thus, the routing is simple: if the destination node is in the same cluster of the source node, it uses shortest path routing; otherwise the message is routed to the closest landmark of the destination node and then forwarded to the destination node.

Through the discussion in this section, we focus on the distributed routing implementation, in particular the case for $N \times N$ routing relation for small routing table size. Towards a good trade-off between RT and the routing path length in high-performance systems, we compare our proposed scheme with the stretch-1 Hierarchical Routing for Torus and Dragonfly networks [6], and the universal compact routing of Thorup and Zwick [9] for random networks.

III. Our Hierarchical Compact Routing

A. Basic approach

Following the survey in Section II, we can observe that if a random network has *regularity*, Hierarchical Routing and Addressing may archive a small routing table size and small average routing path length. To design a *regular random* network, we learn the idea of adding random links into a classical network such as a ring or a grid. The prior analysis [16] found that this method can significantly reduce the diameter of the network by taking advantage of the additional random links. The network also takes advantage of the regularity of the base network, which is helpful for deployment. For example, Shin et al. proposed degree-6 random networks for data centers [17] based on Kleinberg's small-world network model [13] starting from 2-D Torus (or 3-D Torus) and then adding random links. This design guarantees the ease of deployment in a server

[1] $\tilde{O}(f(n)) = f(n) \times \text{polylog}(n)$

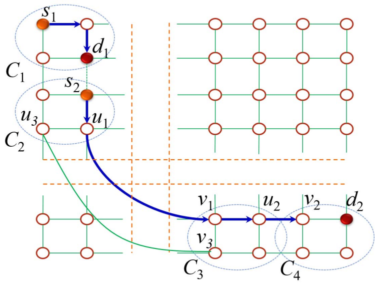


Figure 1. Examples of HR-SW Routing. A message is routed from s_1 to d_1 inside cluster C_1 via shortest path R(s_1,d_1). The route between different-cluster switches is combination of inter and intra-cluster routes, e.g., the route between s_2 to d_2 includes R(s_2,u_1), (u_1,v_1), R(v_1,u_2), (u_2,v_2), R(v_2,d_2). In this case, the intermediate link (u_1,v_1) that connect C_2 to C_3 is chosen instead of (u_3,v_3) because u_1 is nearer s_2 than u_3.

room using the regularity of the torus topology. Therefore, we consider a *regular random* network as a good choice to implement Hierarchical Routing and Addressing such as Kleinberg's small-world network model.

The Kleinberg small-world network model (*SW*-network) is constructed from a grid where each node has links to each other within r grid steps. Each node also has p random links generated with the probability inversely proportional to the link's distance. In detail, let $d(u, v)$ denote the grid distance between two nodes u and v. Each random outgoing link from u to v has a probability proportional to $d(u, v)^{-q}$ where q is a parameter called *clustering exponent*[2].

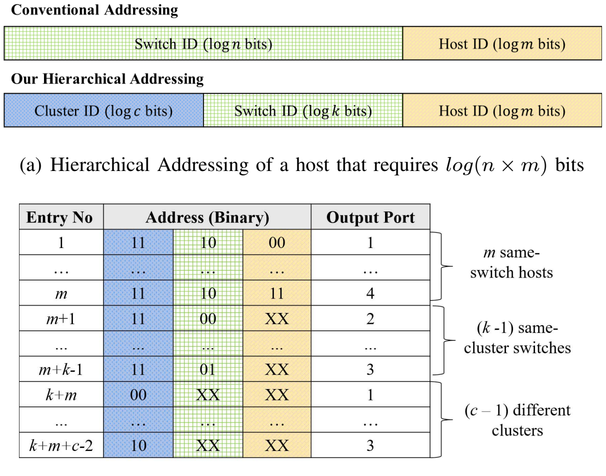
B. HR-SW routing

We now describe our Hierarchical Routing algorithm on the Kleinberg small-world network model (hereafter HR-SW routing). Remind that the main idea of Hierarchical Routing is based on the partitioning the network into clusters and the message forwarding mechanism at each switch.

We consider a grid-based *SW*-network G of n switches arranged into X rows $\times Y$ columns. We also assume that switches connect to the same number of end hosts denoted by m. In our scheme, the network G is equally partitioned into smaller grids of size $a \times b$ switches, i.e., each grid is considered as a cluster. The network now can be seen as an *SW*-network with the size of $c = \frac{X}{a} \times \frac{Y}{b}$ clusters where a cluster i also seen as a subgraph CL_i. Two clusters CL_1 and CL_2 are connected if there exists a link that connects a switch in CL_1 to a switch in CL_2, typically the random links.

Using the above partitioning, we generate the routing between two any given switches s and d. If the source switch s and the destination d are in the same cluster CL, the message is routed via a shortest path on the subgraph CL. We use R(s,d) to denote this intra-cluster routing path.

An inter-cluster routing between CL_s and CL_d includes two phases. The first phase is finding the shortest path between the clusters on the graph G'. Then each intermediate cluster of this inter-cluster routing path is replaced with the intra-cluster routing in the second phase. Without loss of generality, assume that $CL_s \to CL_i \to ... \to CL_d$, $i \in (1,k)$ present the k+1 hop-path between clusters. The route is the combination of R(s,u_1), (u_1,v_1), R(v_1,u_2) ... R(v_k,d) where (u_i,v_i) is the link connects clusters CL_{i-1} and CL_i together. Note that there exists many such pairs of u_i and v_i. In this case, we heuristically choose the switch u_i that nearest the current switch due to achieve short path length, e.g., s in the beginning or v_{i-1} at the cluster CL_{i-1}. Figure 1 illustrates an example of Hierarchical Routing with an intra-cluster path between s_1 and d_1, and an inter-cluster routing between s_2 and d_2.



(a) Hierarchical Addressing of a host that requires $log(n \times m)$ bits

Entry No	Address (Binary)			Output Port
1	11	10	00	1
…	…	…	…	…
m	11	10	11	4
m+1	11	00	XX	2
...	...	...	...	...
m+*k*-1	11	01	XX	3
k+*m*	00	XX	XX	1
...	…	…	…	…
k+*m*+*c*-2	10	XX	XX	3

(b) The routing table of a switch with $(c - 1) + (k - 1) + m$ entries

Figure 2. Hierarchical Addressing and Routing Table at a switch in a network includes n switches partitioned into c clusters (m hosts per switch, $k = \frac{n}{c}$ switches per cluster)

We have the following fact on the upper bound of the maximum routing path length.

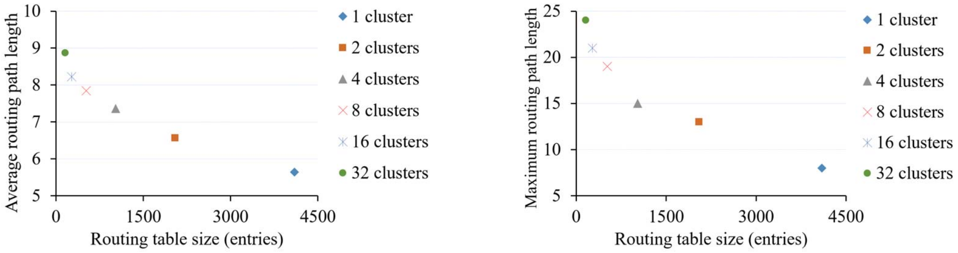
Fact 1: For a given random network G and a network of cluster G', let Δ denote the diameter of G' and δ denote the maximum diameter of the clusters. The maximum routing path length is at most $(\Delta + 1) \times \delta$.

C. Addressing

Let us describe our addressing mechanism that implements the HR-SW routing in detail. We consider a random network G of n switches. We also assume that switches connect to the same number of end hosts denoted by m. In our scheme, the network G is equally partitioned into c clusters of $k = \frac{n}{c}$ switches. Our Hierarchical Addressing is designed to support the longest-prefix-matching lookup technique implemented in current TCAM [18]. Each switch keeps the complete information of all the hosts directly connected to it and all the switches in the same cluster. Besides, all the switches of another cluster are grouped and stored by only one entry in the routing table, i.e., each cluster requires one routing table entry. Thus, the address of a host becomes a combination of host identifier (ID), the switch ID, and the cluster ID. This addressing mechanism requires the same size of memory as the conventional addressing as shown in Figure 2(a). Clearly, each switch requires $RT = (c-1)+(k-1)+m$ routing table entries as presented in Figure 2(b) and the equation below.

$$RT = c + \frac{n}{c} + m - 2 \tag{1}$$

[2] The analysis of Kleinberg [13] showed that the *SW*-network has good performance in terms of average path length when the clustering exponent is set in the range between 1.5 and 2.5



(a) Average routing path vs. routing table size

(b) Maximum routing path vs. routing table size

Figure 3. Trade-off between routing table size and routing path length of HR-SW Routing in a network of 4,096 switches.

The Equation 1 shows that the number of routing table entries depends on the number of clusters c (and also the cluster size $\frac{n}{c}$). Thus, we have to consider how to partition the network. We analyze the trade-off between routing table size, routing path length, and the number of clusters in Section IV.

IV. Experiment

A. Analysis of the HR-SW

We aim at measuring the impact of number of clusters to the routing table size (RT) and routing path length of HR-SW. In this experiment, we add only $p = 2$ random links at each switch with the clustering exponent $q = 1.6$. Since we assume that the switches connect to the same number of end hosts, we use the formula (1) for calculating the routing table size without including the number of host per switch, i.e., $RT = c + \frac{n}{c} - 1$.

For a given network size n, the RT now become a function of the number of clusters c and reach the minimum value when c approximately equals to $n^{1/2}$. Thus, we choose the number of clusters from 1 to 32 when the network size ranges from 1,024 to 8,192 switches. The case of 1 cluster means we use the shortest path routing where a switch knows the information of the whole network.

Figure 3 shows the average and maximum routing path length of HR-SW Routing in a network of 4,096 switches. Shorter routing path and smaller RT are considered better. We found that the larger number of clusters, the longer the routing path length. For example, when the number of clusters are 4 and 16, the average routing path length are 31% and 46% higher than the case of 1 cluster, respectively. Those are 88% and 163% for the maximum routing path length. The result implies that if we want to save more cost (typically the routing table size), we have to sacrifice performance (typically the routing path length).

In addition, we found that the increasing rate of the routing path length (also the decreasing rate of the routing table size) becomes slower when the number of clusters increases. Hence, we choose the case of 16 clusters when comparing HR-SW to the other compact routing algorithms in the next subsection.

B. HR-SW vs. Compact routing algorithms

We now compare our proposed HR-SW to other compact routing algorithms for interconnections of HPC systems such as stretch-1 Hierarchical Routing for Torus and Dragonfly [6] (represented as Shortest-3-D Torus, and Shortest-Dragonfly). We also choose the proposal on universal compact routing of Thorup and Zwick [9] for random network and small-world network as the main competitors (represented as TZ-Random, and TZ-SW).

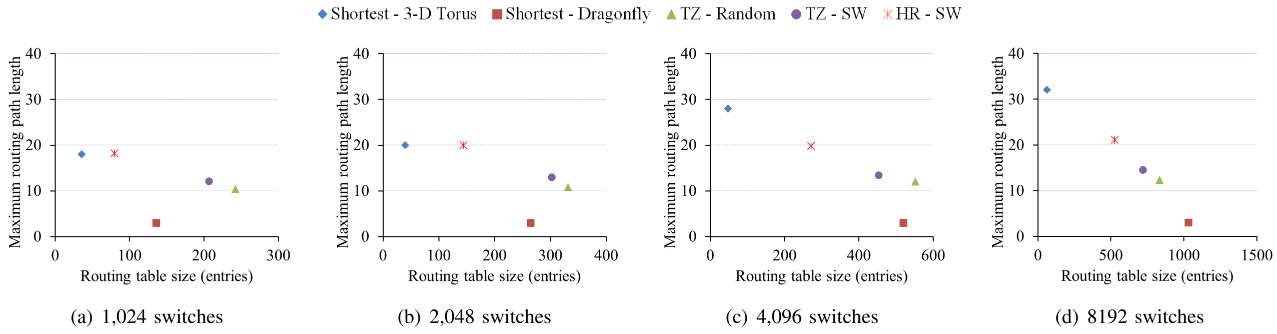
Regarding the evaluation of routing table size, all the evaluated hierarchical compact routing schemes including HR-SW require the same RT because these routing schemes divide the network into clusters equally. However, the compact routing TZ based on landmarks does not maintain this equal-size property, which is allowed in the Internet switches. To apply the TZ in random HPC networks, we use the maximum number of routing table size at a switch in this experiment.

We measure the maximum routing path length to compare the worst case in different network sizes from 1,024 to 8,192 switches in Figure 4. The comparison of the average routing path length for the normal case is shown in Figure 5. Shorter path length and smaller routing table size are considered better. In most network sizes, Dragonfly achieves the shortest routing path length whereas 3-D Torus leads to the longest. However implementation of compact routing on 3-D Torus has the smallest routing table size. Our proposed HR-SW, the TZ-Random, and the TZ-SW have lower values in terms of path length compared to 3-D Torus but higher number of routing table entries. For example, in 8,192-switch networks, the path length of HR-SW is lower than 3-D Torus by 34.4% at maximum and 43.4% on average. Comparing to TZ-Random, HR-SW has longer maximum routing path length, similar average path length but much smaller routing table size, e.g., at least 30% of stored entries are eliminated.

When the network size gets larger, although the 3-D Torus maintains the small routing table size, its routing path length significantly increases. By contrast, the Dragonfly and the TZ-Random are good in terms of routing path length but their routing table size increases quite quickly. Interestingly, our HR-SW has similar average routing path length compared to TZ-Random but has lower increasing rate of the routing table size. Thus, we say that our proposal achieves a good trade-off between the routing table size and the routing path length.

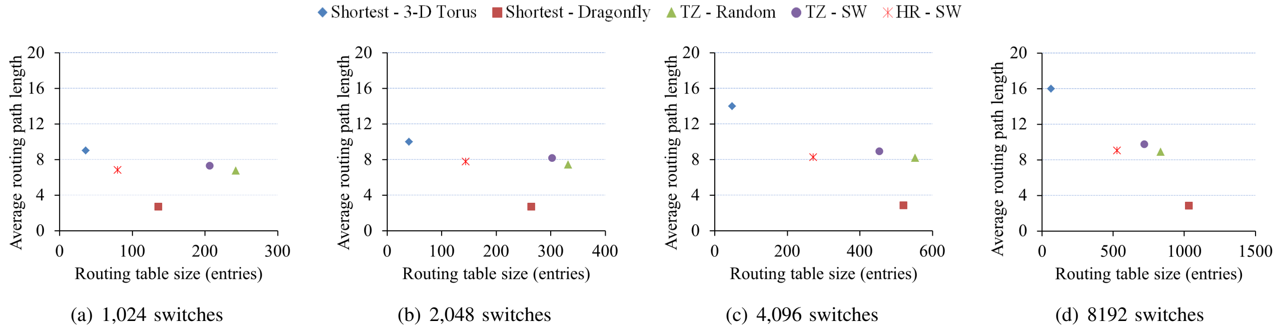
V. Conclusion

In this study, we have proposed the use of a random-based network topology with a hierarchical routing scheme for HPC interconnects in order to have a good trade-off between the routing table size and the average routing path length. The nodes are grouped into clusters so that each node stores the information of all the nodes of another cluster using only one table entry. Our routing cannot always follow the minimal



(a) 1,024 switches (b) 2,048 switches (c) 4,096 switches (d) 8192 switches

Figure 4. Maximum routing path length vs. maximum routing table size in network of 1,024, 2,048, 4,096 and 8192 switches.



(a) 1,024 switches (b) 2,048 switches (c) 4,096 switches (d) 8192 switches

Figure 5. Average routing path length vs. maximum routing table size in network of 1,024, 2,048, 4,096 and 8192 switches.

paths, whereas its average path length is fairly short when compared to that of various network topologies.

Our analysis results showed that the average routing path length decreases logarithmically as the maximum routing table size increases. When the number of clusters increases, the routing table size drastically reduces. For example, in a network of 8,192-switches, the routing table size is 4,098 and 528 for the cases of 2 and 16 clusters, respectively. The larger the number of clusters, the longer the routing path length. For example, with 4,096 nodes in 4-clusters and 16-clusters, the average routing path length increases 31% and 46% compared to the case of 1-cluster, respectively. These properties are quite different from those of existing topologies and their routing.

Acknowledgment

This work was partially supported by MIC/SCOPE(152103004) and SECOM Science and Technology Foundation.

References

[1] W. D. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[2] B. Towles, J. P. Grossman, B. Greskamp, and D. E. Shaw, "Unifying on-chip and inter-node switching within the Anton 2 network ," in *Proc. of the ACM/IEEE International Symposium on Computer Architecture, (ISCA)*, June. 2014, pp. 1–12.

[3] CrayXT5 Supercomputer, http://www.cray.com/.

[4] GraphGolf: The Order/degree Problem Competition., http://research.nii.ac.jp/graphgolf/.

[5] J. Flich, T. Skeie, A. Mejia, O. Lysne, P. Lopez, 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 Distributed Systems.*, vol. 23, no. 3, pp. 405–425, 2012.

[6] M. Benito, E. Vallejo, and R. Beivide, "On the use of commodity ethernet technology in exascale hpc systems," *2015 IEEE 22nd International Conference on High Performance Computing (HiPC)*, pp. 254–263, 2015.

[7] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks performance evaluation and optimization," *Computer Networks (1976)*, vol. 1, no. 3, pp. 155–174, 1977.

[8] C. Gavoille and M. Gengler, "Space-efficiency for routing schemes of stretch factor three," *Journal of Parallel and Distributed Computing*, vol. 61, no. 5, pp. 679–687, 2001.

[9] M. Thorup and U. Zwick, "Compact routing schemes," in *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 2001, pp. 1–10.

[10] L. J. Cowen, "Compact routing with minimum stretch," *Journal of Algorithms*, vol. 38, no. 1, pp. 170–183, 2001.

[11] D. Krioukov, k. c. claffy, K. Fall, and A. Brady, "On compact routing for the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 41–52, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1273445.1273450

[12] M. Enachescu, M. Wang, and A. Goel, "Reducing Maximum Stretch in Compact Routing," in *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*, 2008, pp. 336–340.

[13] J. Kleinberg, "The small-world phenomenon and decentralized search," *SIAM News*, vol. 37, no. 3, a short essay as part of Math Awareness Month 2004.

[14] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *Proc. of the International Symposium on Computer Architecture (ISCA)*, 2008, pp. 77–88.

[15] M. Thorup and U. Zwick, "Approximate Distance Oracles," in *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, ser. STOC '01. New York, NY, USA: ACM, 2001, pp. 183–192.

[16] 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 International Symposium on Computer Architecture (ISCA)*, 2012, pp. 177–188.

[17] J.-Y. Shin, B. Wong, and E. G. Sirer, "Small-world datacenters," in *Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC11)*. New York, NY, USA: ACM, 2011, pp. 2:1–2:13.

[18] H. Liu, "Routing table compaction in ternary CAM," *IEEE Micro*, vol. 22, no. 1, pp. 58–64, 2002.