PAPER

A Generalized Theory Based on the Turn Model for Deadlock-Free Irregular Networks

Ryuta KAWANO[†a)], ***Member***, **Ryota YASUDO**[†], ***Nonmember***, **Hiroki MATSUTANI**[†], ***Member***, **Michihiro KOIBUCHI**[††], ***Senior Member***, ***and*** **Hideharu AMANO**[†], ***Fellow***

SUMMARY Recently proposed irregular networks can reduce the latency for both on-chip and off-chip systems with a large number of computing nodes and thus can improve the performance of parallel applications. However, these networks usually suffer from deadlocks in routing packets when using a naive minimal path routing algorithm. To solve this problem, we focus attention on a lately proposed theory that generalizes the turn model to maintain the network performance with deadlock-freedom. The theorems remain a challenge of applying themselves to arbitrary topologies including fully irregular networks. In this paper, we advance the theorems to completely general ones. Moreover, we provide a feasible implementation of a deadlock-free routing method based on our advanced theorem. Experimental results show that the routing method based on our proposed theorem can improve the network throughput by up to 138 % compared to a conventional deterministic minimal routing method. Moreover, when utilized as the escape path in Duato's protocol, it can improve the throughput by up to 26.3 % compared with the conventional up*/down* routing.

key words: *interconnection networks, deadlock-free routing algorithm, high performance computing, irregular networks, virtual channels*

1. Introduction

To improve the performance of large parallel applications, low-latency and high-throughput interconnection networks are essential as well as processing performance of computational nodes. The performance on off-chip interconnection networks is usually dominated by the delay in switching fabrics (e.g., about a hundred nano-seconds in Infiniband QDR) rather than in a link and for injection. Therefore, researchers have recently focused attention on low-latency networks with high-radix switches, which can be modeled as small-diameter topologies with large degrees [1]–[3].

Recent approaches have shown that irregular topologies adopted in inter-switch networks can significantly reduce the end-to-end latency [4]–[7]. These networks can improve the performance of parallel applications not only for off-chip networks but for on-chip inter-core networks with low-radix routers [8], [9].

In this work, we focus on the practical use of the irregular networks. To improve the availability of these networks, routing algorithms have to guarantee deadlock-freedom in packet transfer. This is because conventional routing algorithms, such as the dimension order routing for k-ary n-cube topologies or the routing with node labeling for fat-tree topologies, cannot be naively utilized in the irregular networks. Topology-agnostic deadlock-free routing methods often face a trade-off among the buffer size required to implement Virtual Channels (VCs), the achieved latency, and the throughput.

The good trade-off mentioned above can be provided by routing algorithms based on the turn model [10]. This model prohibits a portion of local turns for packet transfer on a network. Since these turns are independent of the network structure and the network size, it can reduce the required buffers to support deadlock-freedom. Although this model cannot be applied naively to the irregular networks, various routing methods based on the turn model are proposed for these networks, including L-turn routing [11], Tree-turn routing [12], and a routing method for faulty networks [13]. However, these methods yet leave the challenge of a generalization of the turn model for irregular networks with arbitrary structures and network sizes.

In this work, we propose a novel and generalized theorem, called HiRy*, to design deadlock-free adaptive routing methods for arbitrary network topologies. The theorem is developed from a lately proposed routing theory called EbDa [14], a generalization of the turn model [10] for n-dimensional Mesh and k-ary n-cube topologies. We also provide a feasible implementation of a topology-agnostic routing algorithm based on our theorem HiRy.

The paper [15] from our earlier stage of research only treated limited cases with informally described algorithms. By contrast, this work formalizes description of the proposed theorems in detail and demonstrates the utility of the proposed method as an escape path in the fully-adaptive routing based on Duato's protocol [16].

The rest of the paper is organized as follows. Section 2 overviews the theorems of EbDa with an example of applying them to a conventional routing method based on the turn model. Section 3 describes our advanced theorem applicable to arbitrary topologies. In Sect. 4, we provide a new deadlock-free routing algorithm for irregular networks based on our theorem. Section 5 shows the theoretical analysis of our proposed method. Section 6 provides the

Manuscript received October 29, 2018.
Manuscript revised March 30, 2019.
Manuscript publicized October 8, 2019.

[†]The authors are with Dept. of ICS, Keio University, Yokohama-shi, 223–0061 Japan.
[††]The author is with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: blackbus@am.ics.keio.ac.jp

DOI: 10.1587/transinf.2018EDP7367

*The name HiRy is derived from the first two letters of the last author's first name and those of the first (or the second) author's.

evaluation of our provided routing method and comparison with conventional routing methodologies by a network simulator. Finally, we conclude this paper and mention future work in Sect. 7.

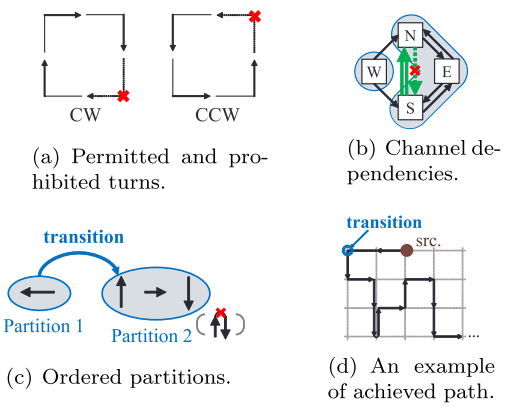
2. Related Work

Dally's theory [17] confirms that deadlock-freedom in a network is guaranteed if and only if the channel dependency graph (CDG) induced by the usage of channels with packets is acyclic. Routing algorithms based on this theory [16], [18]–[21] support deadlock-freedom with an exhaustive cycle dependency check for a given topology. This lacks their scalability for an arbitrary network, especially with a large network size.

The turn model [10] focuses on directions of channels in n-dimensional Mesh and k-ary n-cube topologies to design deadlock-free adaptive routing for these topologies. Figure 1 (a) shows an example of the turn model, called the West-First routing. This model shows that any loop is avoided by prohibiting a portion of turns. Various routing methods [22], [23] based on this model can be applied with high scalability because the prohibited turns are independent of the network structure and the network size.

A lately proposed theory, called EbDa [14], generalizes the turn model to design deadlock-free routing. It utilizes a partitioning strategy for the channel directions to form an acyclic channel dependency graph. EbDa can be applied to the West-First routing in a 2D Mesh topology shown in Fig. 1 (a). Let all of the links in the topology be classified according to their directions; i.e., they are grouped into N, S, E, and W links.

In EbDa, a set of directions that packets can use arbitrarily and repeatedly is arranged into a *partition*. In the West-First routing, the two groups {W} and {N, E, S} are generated, as shown in Fig. 1 (b). The solid arrows in the figure denote the permitted turns between the directions. Moreover, EbDa confirms that an additional turn from S to N can be permitted without causing any deadlock. The permitted and prohibited turns between N and S are represented as the doublet and dotted arrows, respectively.



(a) Permitted and prohibited turns.
(b) Channel dependencies.
(c) Ordered partitions.
(d) An example of achieved path.

Fig. 1 An example of turn model (West-First routing).

Figure 1 (c) represents the two partitions derived from the channel dependencies in Fig. 1 (b). A *transition* between the two partitions represents the permitted turns from W to any other direction. Figure 1 (d) shows an example of a path with the routing. Any loop in routing packets is removed with the following three limitations.

1. Packets injected to the source node use W direction before a turn to any other direction. The turn corresponds to the transition from the partition 1 to 2 in Fig. 1 (c). After the turn, the packets cannot use the W direction again because it means the wrong transition. This unidirectional transition avoids any loop between the partition 1 and 2.
2. After the transition, the packets have to move towards the eastern direction along the horizontal coordinate axis. It means that there are no loops for the horizontal movement within the partition 2.
3. The vertical movement of the packets cannot close any loop because of the prohibited 180-degree turn from N to S in the partition 2.

These three limitations correspond to the following three theorems introduced in EbDa, respectively.

(i) No cycles are formed with transitions in an ascending order among strictly ordered partitions that do not share any common channel with each other.

(ii) A partition is loop-free if the number of axes, whose positive and negative directions exist in the partition, is at most one.

(iii) A partition maintains its deadlock-freedom if the channels that can induce 180-degree turns are used in a strict order.

Note that for the West-First routing in Fig. 1, the vertical channels are ordered as (S, N) based on their directions to satisfy the condition in the third theorem.

The difference of this work from EbDa is that (1) unlike EbDa, our theorem HiRy can be applied to arbitrary topologies by introducing a concept of *regions* to define continuous directions of channels, and that (2) we implement a possible routing method based on HiRy and evaluate the performance, while EbDa does not introduce a new routing method.

3. Theory to Design Deadlock-Free Routing for Arbitrary Topologies

Unlike EbDa, channels on an arbitrary n-dimensional space can be arranged as diagonal ones; that is, they do not have to be parallel to any of the n coordinate axes. The other assumptions are the same as in EbDa. HiRy can be applied not only to wormhole switching networks but also to virtual cut switching and store-and-forward switching networks. Packets can be with arbitrary lengths. The number of Virtual Channels (VCs) in a physical channel can be an arbitrary

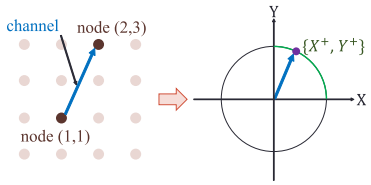


Fig. 2 Channel direction mapped on n-sphere ($n = 2$).

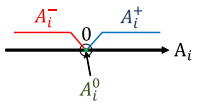
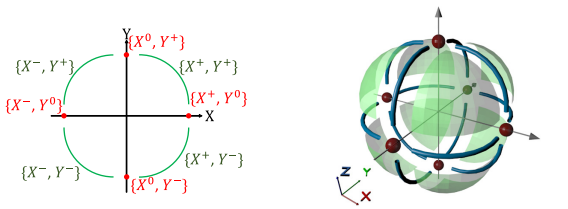


Fig. 3 Negative, zero, and positive coordinates on axis A_i.



(a) Regions on 2-sphere. (b) Regions on 3-sphere.

Fig. 4 Regions on 2D- and 3D-surfaces.

positive integer. The VCs are treated as disjoint channels even if they are on the same physical channel.

3.1 Definitions

We introduce an n-sphere[†] centered at the origin in the n-dimensional space and map a direction of a channel to a point on the n-sphere. Figure 2 shows an example of the mapping. In this example, the direction of the channel from the node $(1, 1)$ to $(2, 3)$ is mapped to the point on the 2-sphere (i.e., the circle) in the first quadrant. Note that the circular arc in the quadrant is labeled as $\{X^+, Y^+\}$. This labeling manner is defined hereinafter.

A coordinate space of an axis A_i is divided into negative, zero, and positive coordinates, which are denoted as A_i^-, A_i^0, and A_i^+, respectively (Fig. 3). By using this division recursively for all axes, the n-dimensional space can be split into 3^n parts. The n-sphere exists on all of the parts except for the origin. Therefore, the n-sphere can be divided into $3^n - 1$ regions with the division of each axis.

A *region* in the n-sphere is defined as a set of the coordinates for all axes, where each coordinate is either negative, zero, or positive. For example, the region in the first quadrant on the 2-sphere is described as $\{X^+, Y^+\}$ as shown in Fig. 4 (a). As shown in this figure, the 2-sphere can be divided into (1) 4 regions denoted as circular arcs, and (2) 4 regions denoted as vertices. Similarly, as shown in Fig. 4 (b), the 3-sphere can be divided into (1) 8 regions denoted as curved surfaces, (2) 12 regions denoted as circular arcs, and (3) 6 regions denoted as vertices.

[†]In this work, an n-sphere is defined as the generalization of a circle that overlies an n-dimensional space; e.g., a 2-sphere and a 3-sphere denote a circle and a sphere, respectively.

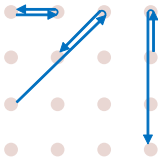


Fig. 5 Example of 180-degree turns.

The definition of a 180-degree turn is identical to the one given in EbDa:

Definition 1. *A 180-degree turn represents packet forwarding between two channels that have the exact opposite directions to each other (Fig. 5).*

In the same way as EbDa, a *partition* is introduced as a set of channels that packets can use arbitrarily and repeatedly except for 180-degree turns. While routing packets with the channels in the partition, all 180-degree turns are prohibited as default. In this work, we define the *partition* as a set of the *regions* that represent continuous directions of channels. With this definition, we can treat diagonal links in the n-dimensional space.

3.2 Theorem for Deadlock-Free Arbitrary Topologies

This section utilizes the definitions in Sect. 3.1 to generalize the theorems of EbDa. Lem. 1, Lem. 2, and Thm. 1 introduced in this section correspond to the three theorems of EbDa (ii), (iii), and (i) in Sect. 2, respectively.

Lemma 1. *A partition is deadlock-free if the number of axes, whose positive and negative coordinates exist in one of the regions in the partition, is at most one.*

Proof. The deadlock-freedom is supported if an acyclic channel dependency graph (CDG) is formed with packet transfer. Let A_c be the axis whose positive and negative directions can be taken in the partition. Along any of all axes except for A_c, packets always have to move in a unidirection of either the positive or negative direction. Therefore, any loop among the channels cannot be formed along these axes. Routing packets along A_c also cannot form any loop because all 180-degree turns are prohibited as default shown in Sect. 3.1. □

Figure 6 (a) illustrates an example of a loop-less path by prohibiting 180-degree turns along A_c. Figure 6 (b) shows another example of a partition for 2-dimensional networks. The partition includes five regions of $\{X^-, Y^0\}$, $\{X^-, Y^+\}$, $\{X^0, Y^+\}$, $\{X^+, Y^+\}$ and $\{X^+, Y^0\}$, prohibiting 180-degree turns along the X axis (i.e., those between $\{X^-, Y^0\}$ and $\{X^+, Y^0\}$). We can apply the proof of Lem. 1 with A_c = X. The axis A_c is hereinafter referred as a *complete axis*.

Lemma 2. *A partition that satisfies the condition of Lem. 1 maintains its deadlock-freedom if the 180-degree turns are*

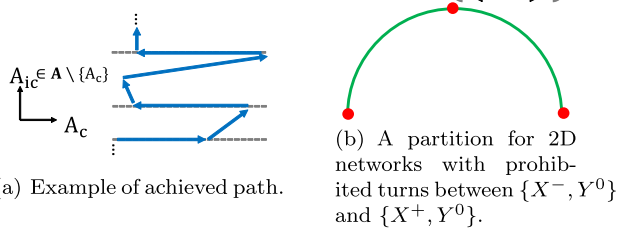
allowed along the complete axis A_c where the channels on each straight line parallel to the axis A_c are used in a strict order.

Proof. When the channels on one of the straight line parallel to A_c are used in a strict order, it cannot close any loop because of the prohibited 180-degree turn from either the positive or negative direction on each line parallel to the comple axis A_c. This relaxation for the condition in Lem. 1 does not cause any additional loop among all channels in the network. □

Figure 7 (a) illustrates an example of a loop-less path by prohibiting a 180-degree turn from A_c^- to A_c^+. In this example, the channels are used in a strict order (i.e., using A_c^+ before A_c^-) on each horizontal broken line parallel to A_c. Figure 7 (b) shows another example of a partition for 2-dimensional networks, whose configuration is the same as in Fig. 6 (b) except for permitting a turn from $\{X^+, Y^0\}$ to $\{X^-, Y^0\}$. Lem. 2 can be applied to this partition and therefore it keeps its deadlock-freedom.

Theorem 1. *Let us assume a set of partitions that satisfy the condition in Lem. 1 and 2 and that do not share any common region with each other. If these partitions are strictly ordered and the channels of the partitions are used in the order, the deadlock-freedom is guaranteed.*

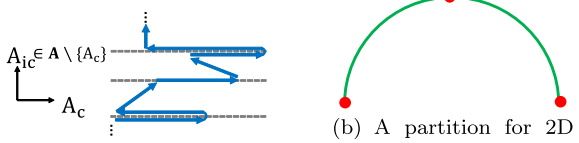
Proof. Dependencies among the partitions do not form any cycle when following the order. Each region in the network belongs to at most one of the partitions. This avoids any loop in the dependencies among the channels in the different partitions. Since the loop-freedom is supported within each partition with Lem. 1 and 2, the deadlock-freedom is supported as a whole. □



(a) Example of achieved path.

(b) A partition for 2D networks with prohibited turns between $\{X^-, Y^0\}$ and $\{X^+, Y^0\}$.

Fig. 6 Examples for proof of Lem. 1.



(a) Example of achieved path.

(b) A partition for 2D networks with a prohibited turn from $\{X^-, Y^0\}$ to $\{X^+, Y^0\}$.

Fig. 7 Examples for proof of Lem. 2.

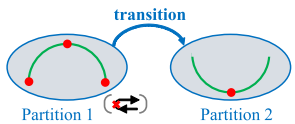


Fig. 8 Ordered partitions containing permitted regions.

Figure 8 shows an example of a transition between disjoint partitions. Since no regions are shared between the partition 1 and 2, any channel is also not shared between them. In the same way as EbDa, the transition among these disjoint partitions does not disturb the deadlock-freedom.

4. HiRy-Based Deadlock-Free Routing

We provide a feasible implementation of a deadlock-free routing method based on HiRy. The dimension of a network n and the number of VCs v for each physical channel are assumed as given inputs. A given network $G = (N, C)$ is arranged on the n-dimensional space, where N and C represent a set of nodes and a set of uni-directional channels, respectively.

Alg. 1 shows the main algorithm which consists of generating partitions (Sect. 4.1) and sorting them (Sect. 4.2). Packets are routed adaptively along the ordered partitions (Sect. 4.3).

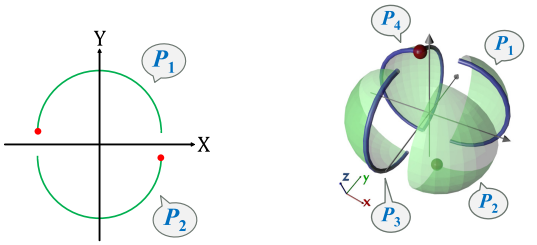
4.1 Generating Partitions for Each VC

Partitions that contain regions are generated based on HiRy. In order to put as many regions into each partition as possible, 2^{n-1} partitions are derived from orthants in the $(n-1)$-dimensional space constructed with all n axes except for a randomly chosen complete axis A_c.

The complete axis A_c is changed for each VC to achieve better optimization in the sorting part shown in Sect. 4.2. For each VC's index i, Alg. 1 repeatedly extracts the complete axis A_c randomly from $\mathbf{A}_c$ which stores the coordinate axes that have not been selected yet. If there are no coordinate axes that can be selected, all coordinate axes are

Algorithm 1 Generating partitions and their order

Input: Dimension n, # of Virtual Channels (VCs) v, Network $G = (N, C)$, satisfying $N \subset \mathbb{R}^n$ and $C \subseteq N^2$
Output: Ordered partitions $\mathcal{P} = (P_1, P_2, \cdots, P_{v \cdot 2^{n-1}})$
  Set of axes $\mathbf{A}$={A_1, A_2, $\cdots$, A_n}
  /* Partition for each VC */
  $\mathbf{A}_c \leftarrow$ {A_1, A_2, $\cdots$, A_n}
  for $1 \le i \le v$ **do**
    if $\mathbf{A}_c = \phi$ **then**
      $\mathbf{A}_c \leftarrow$ {A_1, A_2, $\cdots$, A_n}
    end if
    Randomly pick A_c out of $\mathbf{A}_c$
    Generate set of partitions $\mathbf{P}_i$ with axes $\mathbf{A}$ and complete axis A_c given (See Sect. 4.1)
  end for
  Merge sets of partitions $\{\mathbf{P}_i \mid 1 \le i \le v\}$ into set of partitions $\mathbf{P}$
  Sort $\mathbf{P}$ into $\mathcal{P}$ with network G given (See Sect. 4.2)



(a) An example of generated partitions for a 2D network (A_c = X).

(b) An example of generated partitions for a 3D network (A_c = Z).

Fig. 9 Generated partitions for a VC.

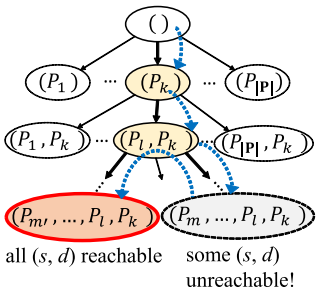


Fig. 10 Best-first search for appropriate order of partitions.

set in $\mathbf{A}_c$ again as the candidates. Then, A_c is continuously extracted in the same way.

The following three kinds of regions are added to the 2^{n-1} partitions.

1. Orthant regions: For each partition, regions located on the corresponding orthant are added. These regions do not contain a zero-coordinate for any of all axes except for A_c.
2. Boundary regions: These regions are adjacent to multiple orthants in the $(n-1)$-dimensional space. Each of them is added to one of the partitions corresponding to the adjacent orthants.
3. Regions of uni-directions in the complete axis A_c: These two regions are added to the different partitions from each other in order to avoid 180-degree turns along the axis A_c mentioned in Lem. 2.

Note that each region is always added to exactly one partition. Therefore, 2^{n-1} partitions in a VC do not have a common region. This condition is used to prove the deadlock-freedom in Sect. 4.4.

In an example for 2-dimensional networks in Fig. 9 (a), the following two partitions are generated for a VC with A_c = X.

- $P_1 = \{\{X^-, Y^0\}, \{X^-, Y^+\}, \{X^0, Y^+\}, \{X^+, Y^+\}\}$
- $P_2 = \{\{X^+, Y^0\}, \{X^+, Y^-\}, \{X^0, Y^-\}, \{X^-, Y^-\}\}$

In this figure, (1) sets of orthant regions are represented as circular arcs; e.g., three regions $\{X^-, Y^+\}$, $\{X^0, Y^+\}$, and $\{X^+, Y^+\}$ construct a set of orthant regions that are added to P_1. (2) there are no boundary regions. (3) two regions in the complete axis A_c = X are denoted as vertices; e.g., $\{X^-, Y^0\}$ and $\{X^+, Y^0\}$ are added to P_1 and P_2, respectively.

Similarly, in an example for 3-dimensional networks in Fig. 9 (b), four partitions P_1 to P_4 are generated for a VC with A_c = Z. In this figure, (1) sets of orthant regions are denoted as curved surfaces; e.g., three regions $\{X^+, Y^+, Z^-\}$, $\{X^+, Y^+, Z^0\}$, and $\{X^+, Y^+, Z^+\}$ construct a set of orthant regions that are added to P_1. (2) sets of boundary regions are denoted as circular arcs; e.g., three regions, $\{X^+, Y^0, Z^-\}$, $\{X^+, Y^0, Z^0\}$, and $\{X^+, Y^0, Z^+\}$ construct a set of boundary regions that are added to P_1. Note that they can instead be added to the neighboring partition P_2. (3) regions of the complete axis A_c = Z are denoted as vertices; e.g., $\{X^0, Y^0, Z^-\}$ is added to P_2.

Algorithm 2 Sorting Partitions

Input: Set of partitions $\mathbf{P} = \{P_1, P_2, \cdots, P_{|\mathbf{P}|(=v \cdot 2^{n-1})}\}$,
Network $G = (N, C)$, satisfying $N \subset \mathbb{R}^n$ and $C \subseteq N^2$
Output: Ordered partitions $\mathcal{P} = (P'_1, P'_2, \cdots, P'_{|\mathcal{P}|(=v \cdot 2^{n-1})})$

```
MAX_ITERATION ← 1,000
Set of temporary partition orders T = {()}
i_iter ← 0
while T ≠ φ and i_iter <MAX_ITERATION do
    i_iter ← i_iter + 1
    Pick T out of T that maximizes # of
    reachable (s, d)-pairs, breaking ties
    with the ASPL with T in ascending order
    if All (s, d)-pairs reachable by T and |T| = |P| then
        return T as P
    else
        for all {P | P ∈ P ∧ P ∉ Set(T)} do
            T' = copy of T
            Insert P to head of T'
            T.add(T')
        end for
    end if
end while
/* No valid partition order found with iterative search */
raise ERROR
```

4.2 Sorting Partitions

After generating the sets of partitions for all VCs, they are merged and sorted for the given network G. We adopt a heuristic best-first search (Fig. 10) to find an order of partitions that ensures all (s, d)-pairs reachable. Alg. 2 shows our implementation of sorting partitions.

In Fig. 10, each vertex in the tree represents temporary ordered partitions. For each iteration, a vertex T that maximizes the number of reachable (s, d)-pairs is selected among unvisited vertices. If there are some ties, they are broken with the average minimal path length in an increasing order. The search continues by adding children vertices of T for all of the rest partitions. If the visited vertex T includes all of the partitions in $\mathbf{P}$ and there exist some unreachable (s, d)-pairs, the vertex T is discarded and the search is continued by going back to another vertex of the tree. On the other hand, if T includes all of the partitions and ensures all

(s, d)-pairs reachable, T is returned as ordered partitions $\mathcal{P}$. In this work, the number of the iterations is limited to 1,000 to reduce the computational complexity.

Note that when the number of VCs v is insufficiently small, there is a possibility that Alg. 1 does not find a partition order $\mathcal{P}$ that ensures all (s, d)-pairs reachable. We evaluate the minimal number of VCs that achieves livelock-free routing in Sect. 5.1.

4.3 Routing Packets

Packets can adaptively use multiple minimal paths that are available with the partitions and their ordering between source and destination nodes. Although they can also use non-minimal paths, we do not recommend using them because they cause degradation in the throughput and the latency depending on the given networks and traffic patterns.

When the packets request allocation for multiple output VCs that induce the minimal paths, the priority is given to the VCs based on the order of the corresponding partitions to improve routing adaptivity.

4.4 Proof of Deadlock-Freedom

The deadlock-freedom of the provided HiRy-based routing is proved as follows.

Theorem 2. *The provided implementation of a routing method based on HiRy is deadlock-free.*

Proof. Since each region is always added to exactly one partition, 2^{n-1} partitions in a VC do not have a common region (Sect. 4.1). Moreover, the VCs are treated as disjoint channels even if they are on the same physical channel (Sect. 3). Therefore, the merged $v \cdot 2^{n-1}$ partitions also have no region in common. Thus, the deadlock-freedom is supported by applying Thm. 1. □

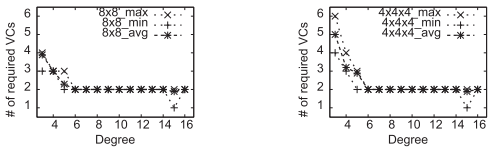
5. Theoretical Analysis

In this section, graph analysis for our HiRy-based routing method is provided. Routing methods are applied to 64- and 256-node random regular topologies. The degree of each node is denoted as $\deg(G)$; that is, the number of bidirectional links for each node is equal to $\deg(G)$.

For 64 nodes, the topologies are developed on the following coordinate spaces: 8×8 ($n = 2$) and $4\times4\times4$ ($n = 3$). Moreover, for 256 nodes, the topologies are developed on the following coordinate spaces: 16×16 ($n = 2$), $8 \times 8 \times 4$ ($n = 3$), and $4 \times 4 \times 4 \times 4$ ($n = 4$). Nodes are arranged on lattice positions in each space.

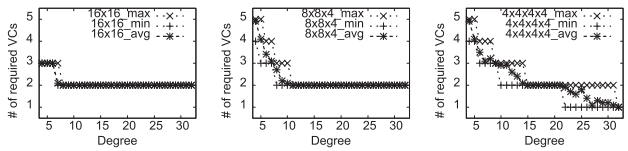
5.1 Minimal Number of Required VCs

The value $\deg(G)$ is varied from 3 to 16 for 64 nodes, and from 4 to 32 for 256 nodes. For each $(|N|, \deg(G))$-pair, 10 random topologies are generated to get the maximum, minimum, and average values, where $|N|$ represents the number of nodes. We calculate the minimal number of required VCs $v_{\min}$ by incrementing $v_{\min}$ repeatedly to achieve Alg. 1 until a partition order $\mathcal{P}$, ensuring all (s, d)-pairs reachable, is generated.



(a) 8×8 random topology. (b) $4\times4\times4$ random topology.

Fig. 11 Number of required virtual channels for 64 nodes.



(a) 16×16 random topology. (b) $8\times8\times4$ random topology. (c) $4\times4\times4\times4$ random topology.

Fig. 12 Number of required virtual channels for 256 nodes.

For 64 nodes, the maximum values with $\deg(G) = 3$ are 4 for 8×8 topologies and 6 for $4 \times 4 \times 4$ topologies, as shown in Fig. 11 (a) and 11 (b), respectively. We can see from these results that when a network with a small degree is developed in a space of a large dimension, it requires many VCs to make all (s, d)-pairs reachable. This is because the large dimension divides channels in a network into many partitions each of which consists of a few channels that are sparsely connected.

Similarly, for 256 nodes, the maximum number of required VCs for the 16×16 topology is 3, while those for the $8\times8\times4$ and $4\times4\times4\times4$ topologies are both 5, as shown in Fig. 12. Moreover, the minimal numbers of the degree that achieve the number of required VCs equal to 2 is 7, 8 and 10 for the 16×16, $8\times8\times4$, and $4\times4\times4\times4$ topologies, respectively. On the other hand, the required number of VCs can be equal to 1 in the case of $\deg(G) \geq 22$ for the $4 \times 4 \times 4 \times 4$ topology. In this case, there are relatively many partitions that include a large number of channels, which leads to high reachability of packets.

The proposed method based on HiRy can be applied to completely irregular networks with a moderate value of the dimension n. This is because the large number of n exponentially increases the number of partitions and decreases the average number of channels in each partition. In this case, the number of $\deg(G)$ has to be large in order to decrease the number of required VCs. Hence it is important to select the appropriate dimension n in order to keep a good balance between the number of partitions and the number of channels in each partition to reduce the number of required VCs. To develop a new methodology to get an optimal dimension n is left for our future work.

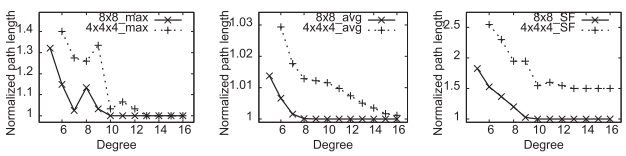
5.2 Path Lengths

In this evaluation, the maximum degrees are set to 16 for 64 nodes and 32 for 256 nodes. The number of VCs is fixed to 2. In a similar way to the previous evaluation, 10 random topologies are generated to evaluate the average values of the following three metrics: maximum path length, average path length, and Stretch Factor that denotes the maximum factor of the path lengths for all (s, d)-pairs. The maximum and average path lengths are divided by those of the shortest path length to obtain the normalized values.

For 64 nodes, Fig. 13 shows that the normalized values get close or equal to one when the degrees become large. Moreover, the networks with a small dimension can reduce the values. For the 2-dimensional networks, the shortest path routing can be achieved with 2 VCs in the case of $\deg(G) \geq 10$. On the other hand, for the 3-dimensional networks, the normalized value only of the maximum can be equal to one in the case of $\deg(G) \geq 13$. Although the increasing rate in the average is only 3 % for $\deg(G) = 6$, the value of the SF does not fall below the value of 1.5.

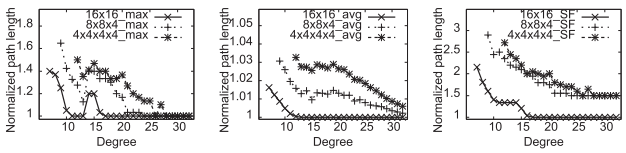
Similarly, for 256 nodes, the 2-dimensional networks can achieve the shortest path routing in the case of $\deg(G) \geq 17$, while the networks with larger dimensions cannot achieve, as shown in Fig. 14. Nonetheless, the 3- and 4-dimensional networks can suppress the increase rates in the average by 1.6 % and 3.3 %, respectively.

In summary, the configuration of the routing method can be varied depending on the given network or the performance to be achieved. A small-degree network with a small dimension can reduce the number of required VCs with the modestly small average path length. Moreover, a large-degree network with a small dimension has the possibility of achieving the shortest path routing, while that with a large dimension can be implemented with a smaller number of VCs.



(a) Maximum. (b) Average. (c) Stretch Factor.

Fig. 13 Normalized path lengths for 64 nodes with 2 VCs.



(a) Maximum. (b) Average. (c) Stretch Factor.

Fig. 14 Normalized path lengths for 256 nodes with 2 VCs.

6. Network Simulation

The HiRy-based routing method is compared with the conventional routing methods by a cycle-accurate network simulator Booksim [24].

6.1 Comparison with LASH-TOR

In this evaluation, the HiRy-based routing method is compared with LASH-TOR [21] that achieves deterministic shortest path routing using transitions among multiple VCs. This method splits a path for each (s, d)-pair into sub-paths that are assigned to multiple VCs.

Network parameters for the simulation are shown in Table 1. Evaluated topologies are the same as in Sect. 5 except for their degrees. The degrees of networks are set to 6 for 64 nodes and 13 for 256 nodes. These values are derived from the minimal numbers of degrees that LASH-TOR can achieve the shortest path routing with 2 VCs for the networks with. The simulation is performed under uniform, transpose, shuffle, and reverse traffics [25] for 64 and 256 nodes, as shown in Fig. 15 and 16, respectively.

For 64 nodes, the latency of HiRy with dimension $n = 3$ is increased by 0.3 % compared to that of LASH-TOR. This result stems from the prohibited turns, which induce both the partial adaptivity in routing packets and the non-minimal paths for some (s, d)-pairs. On the other hand, the saturation throughput of HiRy with $n = 2$ and $n = 3$ is both larger than that of LASH-TOR in most of the traffics. This is because LASH-TOR cannot use the alternative paths to reduce congestion, while HiRy can use them by choosing the multiple paths adaptively. In this evaluation, HiRy can increase the saturation throughput by up to 43.2 %.

For 256 nodes, HiRy can improve the performance in the synthetic traffic patterns. It can increase the saturation throughput by 138 % in the reverse traffic compared to LASH-TOR, as shown in Fig. 16 (d).

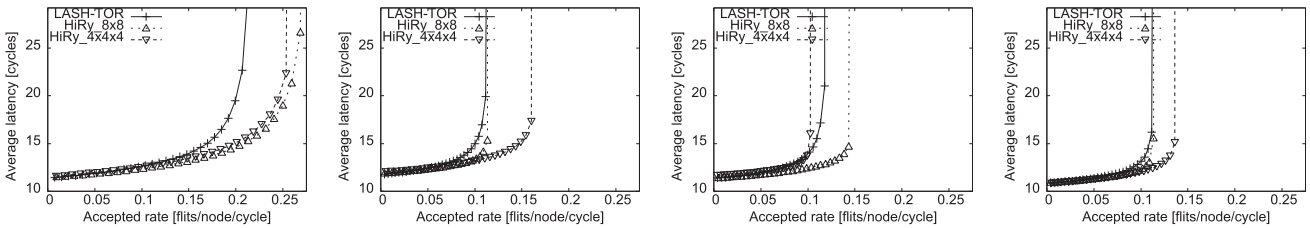
In summary, the implemented routing method based on HiRy can achieve the network performance better than the conventional shortest-path routing method for irregular networks. It can achieve the high saturation throughput by reducing the number of prohibited turns. Moreover, it can achieve the low latency by using the shortest paths in routing most of the packets in the traffics.

6.2 Applying to Duato's Protocol

The HiRy-based routing method with two VCs is compared with the conventional up*/down* routing [19] with the best

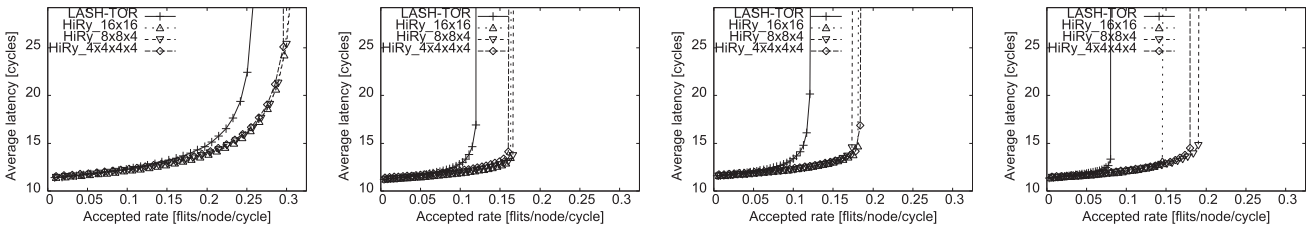
Table 1 Network parameters.

Simulation period	100,000 cycles
Packet size	1 flit
Number of VCs	2
Buffer size per VC	8 flits
Number of pipeline stages	4



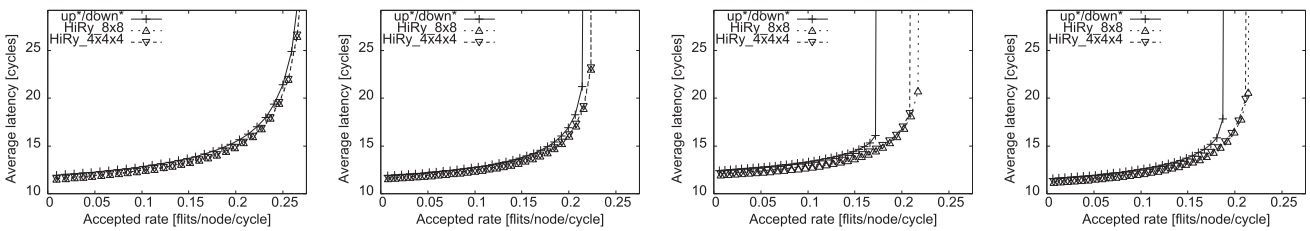
(a) Uniform traffic. (b) Transpose traffic. (c) Shuffle traffic. (d) Reverse traffic.

Fig. 15 Network performance for 64 nodes.



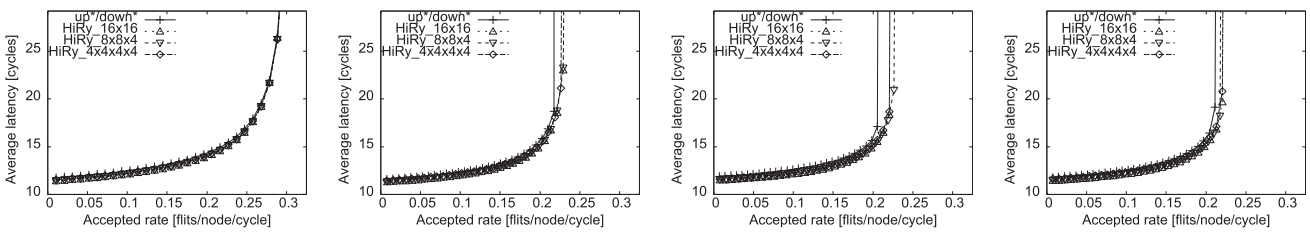
(a) Uniform traffic. (b) Transpose traffic. (c) Shuffle traffic. (d) Reverse traffic.

Fig. 16 Network performance for 256 nodes.



(a) Uniform traffic. (b) Transpose traffic. (c) Shuffle traffic. (d) Reverse traffic.

Fig. 17 Network performance for 64 nodes in the case of applying to Duato's protocol.



(a) Uniform traffic. (b) Transpose traffic. (c) Shuffle traffic. (d) Reverse traffic.

Fig. 18 Network performance for 256 nodes in the case of applying to Duato's protocol.

combination of two spanning trees [26]. Each of the two methods is utilized as the escape paths of Duato's protocol [16] that can achieve minimal adaptive routing with non-minimal deadlock-free escape paths. The network parameters and topologies are the same as in Sect. 6.1 except that three VCs are used in total, one for minimal adaptive routing and two for deadlock-free escape paths.

Figure 17 shows that for 64 nodes, HiRy-based routing method can decrease the network latency with the low injection rate by up to 4.1 % compared with the conventional up*/down* routing. This is because HiRy can achieve shortest path routing in most cases, as shown in the results of Sect. 6.1, while up*/down* routing cannot. Moreover, the saturation network throughput is improved by up to 26.3 %. This result arises from the adaptivity in the HiRy-based routing as described in Sect. 4.3.

Similarly to Sect. 6.1, HiRy can improve the network performance in the synthetic traffic patterns for 256 nodes, as shown in Fig. 18. It can increase the saturation throughput by 10.3 % in the shuffle traffic compared to up*/down* routing, as shown in Fig. 18 (c). Moreover, it can reduce the latency with low injection rate by up to 3.4 % in the shuffle traffic, as shown in Fig. 18 (c).

7. Conclusion and Future Work

In this work, we propose HiRy, the theorem for designing topology-agnostic deadlock-free routing, and provide a feasible implementation of an adaptive routing method for arbitrary networks based on HiRy. HiRy is developed from EbDa, the generalization of the turn model. We advance the theorems by introducing the concept of *regions* that define continuous directions of channels for arbitrary networks. Moreover, the implemented routing method based on HiRy can increase the number of permitted paths and thus can improve the network performance. To support all source-and-destination pairs reachable and to reduce the average path length, a heuristic approach is introduced.

Experimental results show that the routing method based on HiRy can be implemented with only one VC for each physical channel for a 256-node random topology with the degree of 22. Moreover, for a 256-node random topology with the degree of 17, the method can achieve the shortest path routing with two VCs for each physical channel. The results from the network simulation show that it can increase the throughput by up to 138 % compared to LASH-TOR that is one of the deterministic minimal routing methods. Furthermore, when utilized as the escape path in Duato's protocol, it can improve the throughput by up to 26.3 % compared with the conventional up*/down* routing.

As a future work, we will focus on developing a new methodology to find the appropriate dimension n for the given network and the number of VCs. Although the dimension n is given for the routing algorithm in this work, the choice of the dimension significantly influences the minimum number of required VCs or the resulted path lengths. We believe that there is still room for more effective utilization of the proposed theorem HiRy.

References

[1] J. Kim, W.J. Dally, and D. Abts, "Flattened Butterfly: a Cost-Efficient Topology for High-Radix Networks," Proc. International Symposium on Computer Architecture (ISCA), pp.126–137, June 2007.
[2] W. Bao, B. Fu, M. Chen, and L. Zhang, "A High-Performance and Cost-Efficient Interconnection Network for High-Density Servers," Proc. IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), pp.1246–1253, Nov. 2013.
[3] M. Besta and T. Hoefler, "Slim Fly: A Cost Effective Low-Diameter Network Topology," Proc. International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp.348–359, Nov. 2014.
[4] J.-Y. Shin, B. Wong, and E.G. Sirer, "Small-World Datacenters," Proc. Symposium on Cloud Computing (SoCC), pp.2:1–2:13, Oct. 2011.
[5] M. Koibuchi, H. Matsutani, H. Amano, D.F. Hsu, and H. Casanova, "A Case for Random Shortcut Topologies for HPC Interconnects," Proc. International Symposium on Computer Architecture (ISCA), pp.177–188, June 2012.
[6] A. Singla, C.Y. Hong, L. Popa, and P.B. Godfrey, "Jellyfish: Networking Data Centers Randomly," Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp.225–238, April 2012.
[7] "Graph golf: The order/degree problem competition." http://research.nii.ac.jp/graphgolf/.
[8] Ü.Y. Ogras and R. Marculescu, "'It's a Small World After All': NoC Performance Optimization Via Long-Range Link Insertion," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.14, no.7, pp.693–706, July 2006.
[9] H. Yang, J. Tripathi, N.E. Jerger, and D. Gibson, "Dodec: Random-Link, Low-Radix On-Chip Networks," Proc. IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.496–508, Dec. 2014.
[10] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," Proc. International Symposium on Computer Architecture (ISCA), pp.278–287, May 1992.
[11] M. Koibuchi, A. Funahashi, A. Jouraku, and H. Amano, "L-turn Routing: An Adaptive Routing in Irregular Networks," Proc. International Conference on Parallel Processing (ICPP), pp.383–392, Sept. 2001.
[12] J. Zhou and Y.-C. Chung, "Tree-turn routing: an efficient deadlock-free routing algorithm for irregular networks," The Journal of Supercomputing (J Supercomput), vol.59, no.2, pp.882–900, Feb. 2012.
[13] Z. Zhang, A. Greiner, and S. Taktak, "A Reconfigurable Routing Algorithm for a Fault-Tolerant 2D-Mesh Network-on-Chip," Proc. 45th ACM/IEEE Design Automation Conference (DAC), pp.441–446, June 2008.
[14] M. Ebrahimi and M. Daneshtalab, "EbDa: A New Theory on Design and Verification of Deadlock-free Interconnection Networks," Proc. International Symposium on Computer Architecture (ISCA), pp.703–715, June 2017.
[15] R. Kawano, R. Yasudo, H. Matsutani, M. Koibuchi, and H. Amano, "HiRy: An Advanced Theory on Design of Deadlock-Free Adaptive Routing for Arbitrary Topologies," Proc. 23rd International Conference on Parallel and Distributed Systems (ICPADS), pp.664–673, Dec. 2017.
[16] F. Silla and J. Duato, "High-Performance Routing in Networks of Workstations with Irregular Topology," IEEE Trans. Parallel Distrib. Syst., vol.11, no.7, pp.699–719, July 2000.
[17] W.J. Dally and C.L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," IEEE Trans. Comput., vol.C-36, no.5, pp.547–553, May 1987.
[18] W. Qiao and L.M. Ni, "Adaptive Routing in Irregular Networks Using Cut-Through Switches," Proc. International Conference on Parallel Processing (ICPP), pp.52–60, Aug. 1996.
[19] M.D. Schroeder, A.D. Birrell, M. Burrows, H. Murray, R.M. Needham, T.L. Rodeheffer, E.H. Satterthwaite, and C.P. Thacker, "Autonet: A High-speed, Self-configuring Local Area Network Using Point-to-point Links," IEEE J. Sel. Areas Commun., vol.9, no.8, pp.1318–1335, Oct. 1991.
[20] T. Skeie, O. Lysne, and I. Theiss, "Layered Shortest Path (LASH) Routing in Irregular System Area Networks," Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp.194–201, April 2002.
[21] T. Skeie, O. Lysne, J. Flich, P. Lopez, A. Robles, and J. Duato, "LASH-TOR: A Generic Transition-Oriented Routing Algorithm," Proc. International Conference on Parallel and Distributed Systems (ICPADS), pp.595–604, July 2004.
[22] G.M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Trans. Parallel Distrib. Syst., vol.11, no.7, pp.729–738, July 2000.
[23] A. Jouraku, M. Koibuchi, and H. Amano, "An Effective Design of Deadlock-Free Routing Algorithms Based on 2D Turn Model for Irregular Networks," IEEE Trans. Parallel Distrib. Syst., vol.18, no.3, pp.320–333, March 2007.
[24] N. Jiang, D.U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D.E. Shaw, J. Kim, and W.J. Dally, "A Detailed and Flexible Cy-

cle-Accurate Network-on-Chip Simulator," Proc. IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp.86–96, April 2013.

[25] W.J. Dally and B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2004.

[26] H. Matsutani, P. Bogdan, R. Marculescu, Y. Take, D. Sasaki, H. Zhang, M. Koibuchi, T. Kuroda, and H. Amano, "A Case for Wireless 3D NoCs for CMPs," Proc. Asia and South Pacific Design Automation Conference (ASP-DAC), pp.22–28, Jan. 2013.

Ryuta Kawano received the BE, ME and Ph.D. degrees from Keio University, Yokohama, Japan, in 2013, 2015 and 2018, respectively. He is currently an assistant professor in the Department of Information and Computer Science, Keio University. His research interests include the area of high-performance computing and interconnection networks. He is a member of the IEEE and a member of IEICE.

Ryota Yasudo received the BE, ME, and Ph.D. degrees from Keio University, Japan, in 2014, 2016, and 2019, respectively. He is currently an assistant professor at Hiroshima University. His current research interests include interconnection networks, parallel computing, and reconfigurable computing.

Hiroki Matsutani received the BA, ME, and Ph.D. degrees from Keio University in 2004, 2006, and 2008, respectively. He is currently an assistant professor in the Department of Information and Computer Science, Keio University. From 2009 to 2011, he was a research fellow in the Graduate School of Information Science and Technology, The University of Tokyo, and awarded a Research Fellowship of the Japan Society for the Promotion of Science.

Michihiro Koibuchi received the BE, ME, and Ph.D. degrees from Keio University, Yokohama, Japan, in 2000, 2002 and 2003, respectively. Currently, he is an associate professor in the Information Systems Architecture Research Division, National Institute of Informatics and the Graduate University of Advanced Studies, Tokyo, Japan. His research interests include the area of high-performance computing and interconnection networks. He is a member of the IEEE and a senior member of IEICE and IPSJ.

Hideharu Amano received Ph.D. degree from the Department of Electronic Engineering, Keio University, Japan in 1986. He is currently a professor in the Department of Information and Computer Science, Keio University. His research interests include the area of parallel architectures and reconfigurable systems.