High Performance Hierarchical Torus Network Under Adverse Traffic Patterns

M.M. Hafizur Rahman*, Yukinori Sato†, and Yasushi Inoguchi†

*Dept. of Computer Science, KICT, IIUM, Jalan Gombak-53100, Malaysia
E-mail: rahmanjaist@gmail.com, hafizur@iium.edu.my
†Center for Information Science, JAIST, Nomi-Shi, Ishikawa 923-1292, Japan
E-mail: {yukinori & inoguchi}@jaist.ac.jp

***Abstract*— A Hierarchical Torus Network (HTN) is a 2D-torus network of multiple basic modules, in which the basic modules are 3D-torus networks that are hierarchically interconnected for higher level networks. The dynamic communication performance of the HTN using the dimension-order routing under common traffic patterns have been evaluated, and have been shown to be good. However, dynamic communication performance of HTN under adverse traffic patterns has not been evaluated yet. In this paper, we evaluate the dynamic communication performance of HTN using a deadlock-free dimension order routing with 3 virtual channels under adverse traffic patterns, and compare it with H3D-mesh, mesh, and torus networks. It is shown that even under adverse traffic patterns, the HTN yields high throughput and low average transfer time, which provide better dynamic communication performance than H3D-mesh, mesh, and torus networks.**

***Index Terms*— HTN, deadlock-free routing, adverse traffic patterns, dynamic communication performance.**

I. Introduction

High-performance computing is necessary in solving the grand challenge problems in many areas such as development of new materials and sources of energy, development of new medicines and improved health care, strategies for disaster prevention and mitigation, weather forecasting, and for scientific research including the origins of matter and the universe. This makes the current supercomputer changes into massively parallel computer (MPC) systems with thousands of node (Jaguar, Cray XT5-HE), that satisfy the insatiable demand of computing power. In near future, we will need computer systems capable of computing at the petaflops or exaflops level. To achieve this level of performance, we need MPC with tens of thousands or millions of nodes. Interconnection networks are the crucial elements for building MPCs [1]. For future MPC with millions of nodes, the large diameter of conventional topologies is intolerable. Hence, hierarchical interconnection network (HIN) [2], [3] is an efficient way to interconnect the future MPC. A variety of hypercube based HINs found in the literature [4], however, its huge number of physical links make it difficult to implement. To alleviate this problem, k-ary n-cube based HIN [5]–[8] is a plausible alternative way.

It has already been shown that a torus network has better dynamic communication performance than a mesh network. This is the key motivation that led us to consider a hierarchical interconnection network, in which both the basic module (BM) and the interconnection of higher levels have toroidal interconnections. A Hierarchical Torus Network (HTN) [9]–[11] has been proposed as a new hierarchical interconnection network for MPC systems. The HTN consists of a basic module (BM) which is a 3D-torus ($m \times m \times m$). The BMs are hierarchically interconnected by 2D-torus ($n \times n$). To reduce the number of vertical links between silicon planes, we consider higher-level networks as 2D-toroidal connections instead of 3D-toroidal connections, despite the fact that a 3D-torus has better performance than a 2D-torus network. The HTN is attractive since its hierarchical architecture permits the systematic expansion of millions of nodes. We have shown that the HTN possesses several attractive features including constant node degree, small diameter, small average distance, better bisection width, small number of wires, a particularly small number of vertical links, and economic layout area [9].

Wormhole routing (WH) [12] is still the dominant switching technique in MPC systems. Because it has low buffering requirements, and more importantly, it makes latency independent of the message distance. WH typically divide each message into packets, which are then divided into flits. The header flit contains the routing information, and advances along the specified route according to the routing algorithm, the remaining data flits follow the header flit through the network in a pipelined fashion. Since wormhole routing relies on a blocking mechanism for flow control, deadlock can occur because of cyclic dependencies over network resources during message routing. Virtual channels (VCs) [13], [14] were originally introduced to make the routing algorithm deadlock-free in wormhole-routed networks. It is also shown that VCs can also be used to improve network performance and latency by relieving contention.

A group of nodes or all nodes of a MPC work in concert to solve large problems. The nodes communicate or coordinates among themselves by transferring messages though a router, using an efficient routing algorithm. Efficient routing is crucial to the performance of a MPC systems. In a practical router design, the routing decision process should be fast to reduce network latency, while

doi:10.4304/jnw.7.3.456-467

keeping easy implementation in hardware. Dimension-order routing is still very popular because of its low cost and simple router design. This is why most existing MPC, such as IBM Blue Gene (as escape paths), Touchstone, Ametek 2010, and Cosmic cube, use dimension-order routing. We evaluated the dynamic communication performance of the HTN with a dimension-order routing algorithm under various traffic patterns using 3 virtual channels in our previous studies, and it is proved to be better than that of conventional and other hierarchical networks [15]. However, the dynamic communication performance of the HTN under any adverse traffic patterns has not yet been evaluated. The main objective of this paper is to investigate the impact of adversity of traffic on the HTN.

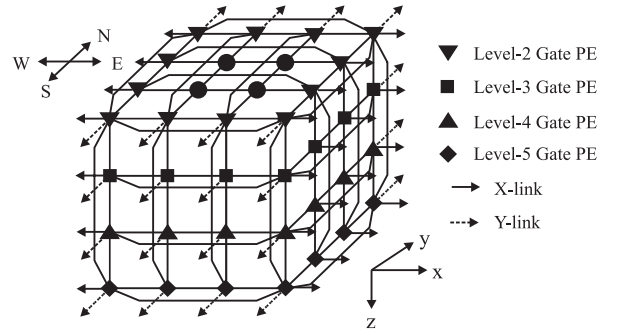
In high traffic, several flits compete for the same resources, either physical links or VCs, but as only one flit can use them, the remainder flits stay buffered in the network, thus blocking other flits and so forth. In this situation, the network is saturated and performance degradation appears. Usually, the traffic generated by real applications is bursty and traffic peaks may saturate the network. Throughput falls down and message latency considerably increases in any networks under adverse traffic. We create this adverse traffic by injecting more packets and synthetic traffic patterns where most of the packets crosses the bisection of the network. These synthetic traffic are called adverse traffic patterns.

The remainder of the paper is organized as follows. In Section II, we briefly describe the basic structure of the HTN. The routing of message in the HTN and its freedom from deadlock is also proved in Section III. The dynamic communication performance of the HTN under adverse traffic pattern is discussed in Section IV. Finally, Section V presents the conclusion of this paper. .

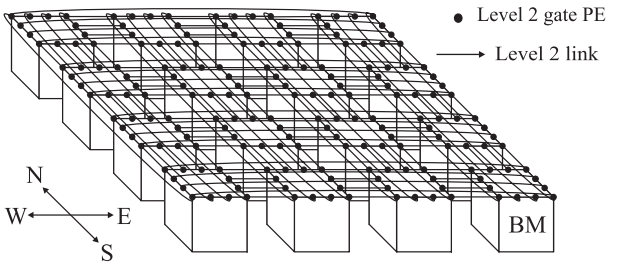
II. INTERCONNECTION OF THE HTN

The *Hierarchical Torus Network (HTN)* [9] is a hierarchical interconnection network consisting of BM that are hierarchically interconnected for higher level networks. The BM of the HTN is a 3D-torus network of size $(m \times m \times m)$, where m is a positive integer. m can be any value, however the preferable one is $m = 2^p$, where p is a positive integer. The BM of a $(4 \times 4 \times 4)$ torus, as depicted in Fig. 1(a), has some free ports at the contours of the xy-plane. A $(m \times m \times m)$ BM has $4 \times m^2$ free ports for higher level interconnection. All free ports, typically one or two, of the exterior Processing Elements (PEs) are used for inter-BM connections to form higher level networks. All ports of the interior PEs are used for intra-BM connections.

Successively higher level networks are built by recursively interconnecting lower level subnetworks in a 2D-torus network of size $(n \times n)$, where n is also a positive integer. A Level-2 HTN can be formed by interconnecting n^2 BMs as a $(n \times n)$ 2D-torus. Similarly, a Level-3 network can be formed by interconnecting n^2 Level-2 subnetworks, and so on. Thus, Level-L is interconnected as a 2D-torus network, in which Level-$(L-1)$ is used as subnet modules. BMs with the same co-ordinate position in each Level-$(L-1)$ subnetwork are interconnected by a 2D-torus network in a Level-L interconnection. As portrayed in Fig. 1(b), a Level-2 HTN, can be formed by interconnecting 16 BMs as a (4×4) 2D-torus network. Each BM is connected to its logically adjacent BMs.



(a) Basic module $(4 \times 4 \times 4)$



(b) Level-2 HTN (4×4)

Figure 1. Interconnection of a HTN

For each higher level interconnection, a BM must use $4m(2^q)$ of its free links: $2m(2^q)$ free links for y-direction and $2m(2^q)$ free links for x-direction interconnections. Here, $q \in \{0, 1, \ \ldots \ , p\}$, is the inter-level connectivity, where $p = \lfloor log_2^m \rfloor$. $q = 0$ leads to minimal inter-level connectivity, while $q = p$ leads to maximum inter-level connectivity. As depicted in Figure 1(a), for example, the $(4 \times 4 \times 4)$ BM has $4 \times 4^2 = 64$ free ports. With $q = 0$, $(4 \times 4 \times 2^0 =)$ 16 free links are used for each higher level interconnection, 8 for y-direction and 8 for x-direction interconnections. The highest level network which can be built from a $(m \times m \times m)$ BM is $L_{max} = 2^{p-q} + 1$. With $q = 0$, Level-5 is the highest possible level to which a $(4 \times 4 \times 4)$ BM can be interconnected. The total number of nodes in a network having $(m \times m \times m)$ BMs and $(n \times n)$ higher level is $N = \left[m^3 \times n^{2(L_{max}-1)}\right]$. Thus, the maximum number of nodes which can be interconnected by the HTN is $N = \left[m^3 \times n^{2(2^{p-q})}\right]$. If $m = 4$, $n = 4$, and $q = 0$, then $N = 4^3 \times 4^8 = 4194304$, i.e, about 4.2 million. A BM with $m = 4$ and the higher levels with $n = 4$ is the most interesting network size because it has better granularity than the larger sizes.

PEs in the BM are addressed by three base-m numbers, the first representing the x-axis, the second representing the y-axis, and the last representing the z-axis. PEs at

Level-L are addressed by two base-n numbers, the first representing the x-axis and the second representing the y-axis. The address of a PE at Level-L HTN is represented as shown in Eq. 1.

$$A^L = \begin{cases} (a_z)(a_y)(a_x) & \text{if } L = 1, \text{ i.e., BM} \\ (a_y^L)(a_x^L) & \text{if } L \geq 2 \end{cases} \quad (1)$$

More generally, in a Level-L HTN, the node address is represented by:

$$\begin{aligned} A &= A^L A^{L-1} A^{L-2} \ ... \ ... \ ... \ A^2 A^1 \\ &= a_\alpha \ a_{\alpha-1} \ a_{\alpha-2} \ a_{\alpha-3} \ ... \ ... \ ... \ a_3 \ a_2 \ a_1 \ a_0 \\ &= a_{2L} \ a_{2L-1} \ a_{2L-2} \ a_{2L-3} \ ... \ ... \ ... \ a_3 \ a_2 \ a_1 \ a_0 \\ &= (a_{2L} \ a_{2L-1}) \ (a_{2L-2} \ a_{2L-3}) \ ... \ ... \ ... \\ &\qquad ... \ ... \ ... \ (a_4 \ a_3) \, (a_2 \ a_1 \ a_0) \end{aligned} \quad (2)$$

Here, the total number of digits is $\alpha = 2L+1$, where L is the level number. The first group contains three digits and the rest of the groups contain two digits. Groups of digits run from group number 1 for Level-1, i.e., the BM, to group number L for the L-th level. In particular, i-th group $(a_{2i} \ a_{2i-1})$ indicates the location of a Level-$(i-1)$ subnetwork within the i-th group to which the node belongs; $2 \leq i \leq L$. In a two-level network, for example, the address becomes $A = (a_4 \ a_3) \ (a_2 \ a_1 \ a_0)$. The last group of digits $(a_4 \ a_3)$ identifies the BM to which the node belongs, and the first group of digits $(a_2 \ a_1 \ a_0)$ identifies the node within that basic module. The detailed architecture of the HTN was presented in [9].

III. Routing Algorithm for HTN

In this section, we review the routing algorithm proposed in our previous study [15] for the convenience of readers. Routing of messages in the HTN is performed from top to bottom. That is, it is first done at the highest level network; then, after the packet reaches its highest level sub-destination, routing continues within the subnetwork to the next lower level sub-destination. This process is repeated until the packet arrives at its final destination. When a packet is generated at a source node, the node checks its destination. If the packet's destination is the current BM, the routing is performed within the BM only. If the packet is addressed to another BM, the source node sends the packet to the outlet node which connects the BM to the level at which the routing is performed.

A. Dynamic Routing Algorithm

For routing messages using dimension-order routing in HTN, first find the nonzero offset in the most significant position by subtracting the current address from the destination. Then make a step towards nullifying the offset by sending the packet in descending order. When the offset along a dimension is zero, then the routing message is switched over to the next dimension. Routing at the higher level is performed first in the y-direction and then in the x-direction. In a BM, the routing order is z-direction, y-direction, and x-direction, respectively.

Routing in the HTN is strictly defined by the source node address and the destination node address. Let a source node address be $s_\alpha, s_{\alpha-1}, s_{\alpha-2}, ..., s_1, s_0$, a destination node address be $d_\alpha, d_{\alpha-1}, d_{\alpha-2}, ..., d_1, d_0$, and a routing tag be $t_\alpha, t_{\alpha-1}, t_{\alpha-2}, ..., t_1, t_0$, where $t_i = d_i - s_i$. The source node address of HTN is expressed as $s = (s_{2L}, s_{2L-1}), \ (s_{2L-2}, s_{2L-3}), ..., (s_2, s_1, s_0)$. Similarly, the destination node address is expressed as $d = (d_{2L}, d_{2L-1}), (d_{2L-2}, d_{2L-3}), ..., \ (d_2, d_1, d_0)$. Figure 2 shows the routing algorithm for the HTN.

B. Deadlock-Free Routing

A useful MPC systems must be both efficient and reliable. A key component of reliability is its routing algorithm should be deadlock-free. Virtual channels [13] are widely used to solve the problem of deadlock in wormhole-routed networks. Since the hardware cost increases as the number of virtual channels increases, the unconstrained use of virtual channels is prohibited for cost-effective parallel computers. A deadlock-free routing algorithm with a minimum number of virtual channels is preferred. However, there is a trade-off between performance and the number of virtual channels [16]–[18]. One design alternative that can be considered is to implement dimension-order routing with extra virtual channels instead of adaptive routing, since virtual channels substantially improve performance [13] and are relatively inexpensive compared to the logic involved in implementing adaptive routing. In [13], Dally showed that the performance of a dimension-order routing under uniform traffic pattern improves significantly as virtual channels are initially added. The benefits then diminish as more channels are added.

We presented a deadlock-free routing with the minimum number of VCs in [10]; and the minimum number is 2. In this paper, we have presented a proof for deadlock-free of that routing using 1 extra VC over the minimum number, because 3 VCs result the best cost-performance trade-off [11]. We have applied the dimension-order routing in each level of the HTN like hierarchical routing algorithms (HiRA) [19]. To prove the proposed routing algorithm for the HTN is deadlock free, we divide the routing path into three phases, as follows:

- *Phase 1:* Intra-BM transfer path from source PE to the face of the BM.
- *Phase 2:* Higher level transfer path.
  - **sub-phase** $2.i.1$: Intra-BM transfer to the outlet PE of Level $(L-i)$ through the y-link.
  - **sub-phase** $2.i.2$: Inter-BM transfer of Level $(L-i)$ through the y-link.
  - **sub-phase** $2.i.3$: Intra-BM transfer to the outlet PE of Level $(L-i)$ through the x-link.
  - **sub-phase** $2.i.4$: Inter-BM transfer of Level $(L-i)$ through the x-link.
- *Phase 3:* Intra-BM transfer path from the outlet of the inter-BM transfer path to the destination PE.

The proposed routing algorithm enforces some routing restrictions to avoid deadlocks [14], [21]. Since

```
Routing HTN(s,d);
source node address:s_α, s_α−1, s_α−2, ..., s_1, s_0
destination node address: d_α, d_α−1, d_α−2, ..., d_1, d_0
tag: t_α, t_α−1, t_α−2, ..., t_1, t_0
  for i = α : 3
    if (i/2 = 0 and (t_i > 0 or t_i = −(n − 1))), routedir = North; endif;
    if (i/2 = 0 and (t_i < 0 or t_i =  (n − 1))), routedir = South; endif;
    if (i%2 = 1 and (t_i > 0 or t_i = −(n − 1))), routedir = East; endif;
    if (i%2 = 1 and (t_i < 0 or t_i =  (n − 1))), routedir = West; endif;
      while (t_i ≠ 0) do
        N_z = outlet_z(s, d, L, routedir)
        N_y = outlet_y(s, d, L, routedir)
        N_x = outlet_x(s, d, L, routedir)
        BM_Routing(N_z, N_y, N_x)
        if (routedir = North or East), move packet to next BM; endif;
        if (routedir = South or West), move packet to previous BM; endif;
        if (t_i > 0), t_i = t_i − 1; endif;
        if (t_i < 0), t_i = t_i + 1; endif;
     endwhile;
  endfor;
    BM_Routing(t_z, t_y, t_x)
end
BM_Routing (t_2, t_1, t_0);
BM_tag t_2, t_1, t_0 = receiving node address (r_2, r_1, r_0) − destination (d_2, d_1, d_0)
  for i = 2 : 0
    if (t_i > 0 and t_i ≤ m/2) or (t_i < 0 and t_i = −(m − 1)), movedir = positive; endif;
    if (t_i > 0 and t_i = (m − 1)) or (t_i < 0 and t_i ≥ −m/2), movedir = negetive; endif;
    if (movedir = positive and t_i > 0), distance = t_i; endif;
    if (movedir = positive and t_i < 0), distance = m + t_i; endif;
    if (movedir = negative and t_i < 0), distance = t_i; endif;
    if (movedir = negative and t_i > 0), distance = −m + t_i; endif;
 endfor
  while(t_2 ≠ 0 or distance_2 ≠ 0) do
    if (movedir = positive), move packet to +z node; distance_2 = distance_2 − 1; endif;
    if (movedir = negetive), move packet to −z node; distance_2 = distance_2 + 1; endif;
  endwhile;
  while(t_1 ≠ 0 or distance_1 ≠ 0) do
    if (movedir = positive), move packet to +y node; distance_1 = distance_1 − 1; endif;
    if (movedir = negetive), move packet to −y node; distance_1 = distance_1 + 1; endif;
  endwhile;
  while(t_0 ≠ 0 or distance_0 ≠ 0) do
    if (movedir = positive), move packet to +x node; distance_0 = distance_0 − 1; endif;
    if (movedir = negetive), move packet to −x node; distance_0 = distance_0 + 1; endif;
  endwhile;
end
```

Figure 2. Dimension-Order Routing Algorithm of the HTN

dimension-order routing is used in HTN, routing at the higher level is performed first in the y-direction and then in the x-direction. In a BM, the routing order is initially in the z-direction, then in the y-direction, and finally in the x-direction. A lemma and a corollary are stated below without proof, which was presented in [10]. By using the following lemma, corollary, and theorem, we will prove that the proposed routing algorithm for the HTN is deadlock-free using 3 virtual channels.

Lemma 1: If a message is routed in the order $z \rightarrow y \rightarrow x$ in a 3D-torus network, then the network is deadlock free with 2 virtual channels [10].

Corollary 1: If the message is routed in the $y \rightarrow x$ direction in a 2D-torus network, then the network is deadlock free with 2 virtual channels [10].

Theorem 1: A Hierarchical Torus Network (HTN) with 3 virtual channels is deadlock-free.

Proof: Both the BM and the higher levels of the HTN

have a toroidal interconnection. In phase-1 and phase-3 routing, packets are routed in the source-BM and destination-BM, respectively. The BM of the HTN is a 3D-torus network. According to Lemma 1, the number of necessary virtual channels for phase-1 and phase-3 is 2. Intra-BM links between inter-BM links on the xy-plane of the BM are used in sub-phases $2.i.1$ and $2.i.3$. These sub-phases utilize channels over intra-BM links, sharing either the channels of phase-1 or phase-3. PEs at the contours of the xy-plane are assigned to each high level as gate nodes. The exterior links of the BM are used in sub-phase $2.i.2$ and sub-phase $2.i.4$, and these links form a 2D-torus network, which is the higher-level interconnection of the HTN. According to Corollary 1, the number of necessary virtual channels for this 2D-torus network is also 2. The mesh connection of the higher level 2D-torus network shares the virtual channel of either sub-phase $2.i.1$ or sub-phase $2.i.3$. The wrap-around connection of the higher level 2D-torus networks requires 1 more virtual channel.

Therefore, the total number of necessary virtual channels for the whole network is 3.

IV. Dynamic Communication Performance

The overall performance of a multicomputer system is affected by the performance of the interconnection network as well as by the performance of the node. Low performance of the communication network will severely limit the speed of the entire multicomputer system. Therefore, the success of massively parallel computers is highly dependent on the efficiency of their underlying interconnection networks.

In our previous study published in [22], we have considered only short message and three adverse traffic patterns, named hot-spot, center-reflection, and tornado traffic patterns to investigate the impact of adverse traffic pattern on HTN performance. In this current study, we have considered two bit permutation and combination traffic patterns such as bit-flip and perfect-shuffle traffic patterns along with these three traffic patterns. We have created adverse traffic situation by injecting more packets in the network by increasing the message length to compete for the network resources. This different message length provides versatile study of dynamic communication performance under the adverse traffic patterns. We have also evaluated the dynamic communication performance of the HTN using more hot spot traffic percentage than that of previous study [22].injecting more packets in the network by increasing the message length to compete for the network resources. This different message length provides versatile study of dynamic communication performance under the adverse traffic patterns. We have also evaluated the dynamic communication performance of the HTN using more hot spot traffic percentage than that of previous study [22].

A. Performance Metrics

The dynamic communication performance of a multicomputer is characterized by message latency and network throughput. Message latency refers to the time elapsed from the instant when the header flit is injected to the network from the source to the instant when the last data flit of the message is received at the destination. Network throughput refers to the maximum amount of information delivered per unit of time through the network. For the network to have good performance, low latency and high throughput must be achieved. In computer simulation, latency is measured in simulator clock cycles and throughput is measured in flits per node and per clock cycle.

B. Simulation Environment

We have developed a wormhole routing simulator using C language to evaluate the dynamic communication performance. In our simulation, we use a dimension-order routing, which is exceedingly simple, provides the only route for the source-destination pair. We have evaluated the dynamic communication performance of HTN, H3D-mesh [23], mesh, and torus networks. Extensive simulations have been carried out for some adverse traffic patterns: hot-spot [24], tornado [25], center-reflection [26], bit-flip [27], and perfect shuffle [27] traffic patterns. In the evaluation of dynamic communication performance, flocks of messages are sent through the network to compete for the output channels. Packets are transmitted by the request-probability r during T clock cycles and the number of flits which reached at destination node and its transfer time is recorded. Then the average transfer time and throughput are calculated and plotted as average transfer time in the horizontal axis and throughput in the vertical axis. The process of performance evaluation is carried out with changing the request-probability r. For each simulation, we have considered that the message generation rate is constant and the same for all nodes.

Flits are transmitted at $20,000$ cycles i.e., $T = 20000$. In each clock cycle, one flit is transferred from the input buffer to the output buffer, or vice versa if the corresponding buffer in the next node is empty. Thus, transferring data between two nodes takes 2 clock cycles. For all of the simulations we have considered short (16 flits), medium (64 flits), and long (256 flits); and the buffer length of each channel is 2 flits. For fair comparison of dynamic communication performance, three VCs per physical link are simulated, and the VCs are arbitrated by a round robin algorithm.

C. Traffic Patterns

In an interconnection network, sources and destinations for messages form the traffic pattern. Traffic characteristics such as message length, message arrival time at the sources, and destination distribution have significant performance implications. Message destination distributions vary a great deal depending on the network topology and the application's mapping onto different nodes. Depending on the characteristics of the application, some nodes may communicate with each other more frequently

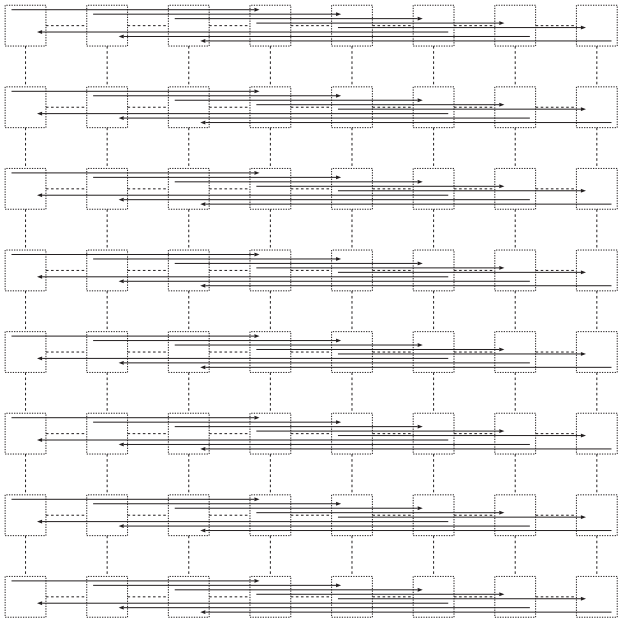


Figure 3. Tornado traffic patterns on a 8×8 mesh network

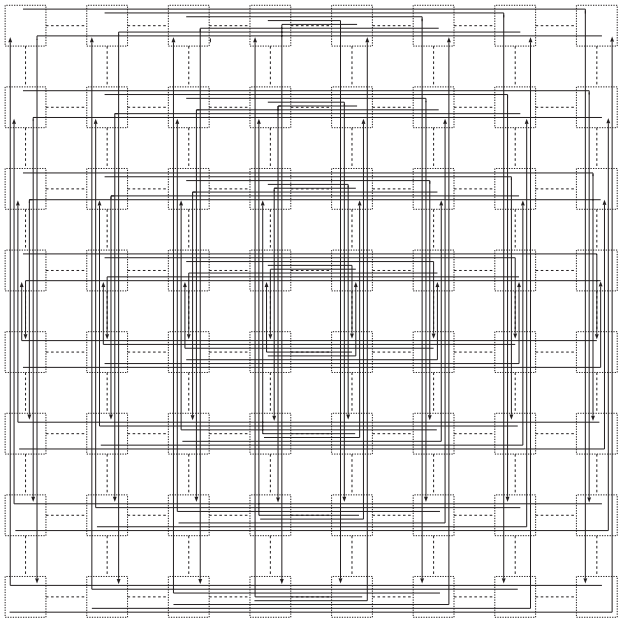


Figure 4. Center-reflection traffic patterns on a 8×8 mesh network

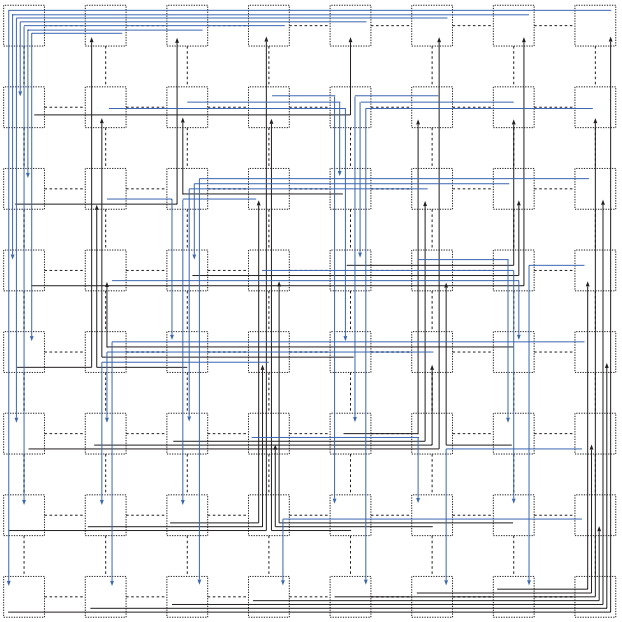


Figure 5. Bit-flip traffic patterns on a 8×8 mesh network

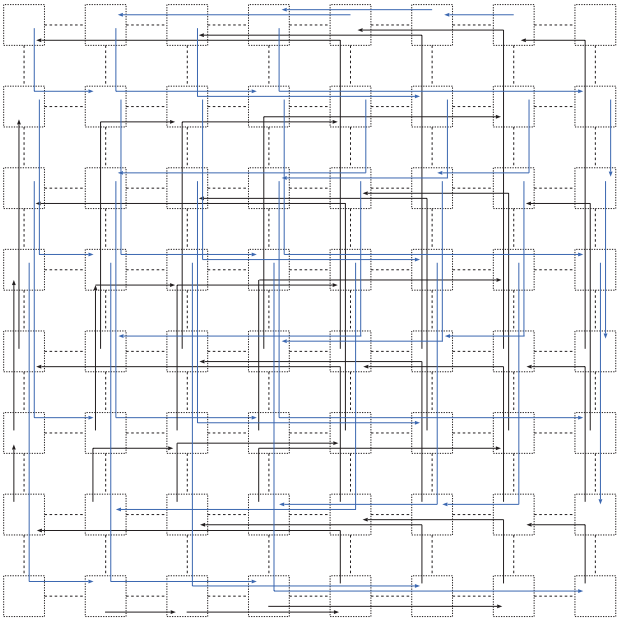


Figure 6. Perfect-shuffle traffic patterns on a 8×8 mesh network

than others. Consequently, adverse traffic situation of the congested node cause uneven usage of traffic resources, significantly degrading the dynamic communication performance of the network. In order to evaluate the dynamic communication performance we use the following five adverse traffic patterns.

- **Hot-Spot** – For generating hot spot traffic we used Pfister and Norton model [24]. According to this model, each node first generates a random number. If that number is less than a predefined threshold, the message will be sent to the hot-spot node. Otherwise, the message will be sent to other nodes, with a uniform distribution.
- **Tornado** – The source s_x sends packets to destination $d_x = s_x + (\lceil k/2 \rceil - 1) \mod k$, i.e., $(k-1)/2$ hops to the right in the lowest dimension [25]. For a 3D network, the node (x, y, z) sends packets to node $\{(x + [k/2] - 1) \mod k, y, z\}$. Where k is the radix of the network.
- **Center Reflection** – With center reflection traffic, the source-destination pair is determined by conceptually reflecting the network about the center in all dimensions. A source at (x, y, z) sends a message to a destination at $(k-x-1, k-y-1, k-z-1)$ [26].

- **Bit-flip** – The node with binary coordinates $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0 communicates with the node $(\overline{b_0}, \overline{b_1},$ ... ... $\overline{b_{\beta-2}}, \overline{b_{\beta-1}})$ [27].
- **Perfect Shuffle** – The node with binary coordinates $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0 communicates with the node $(b_{\beta-2}, ..., b_0, b_{\beta-1})$, i.e., rotate left 1 bit [27].

The evaluation of dynamic communication performance under the worst-case traffic than that of adversarial traffic patters is more accurate to show the suitability of any arbitrary network under a routing algorithm. However, the worst-case pattern is often subtle. The adverse traffic patterns are those patterns which cause load imbalance. In hot-spot traffic, a particular communication link experiences a much greater number of requests than the rest of the links – more than it can service. In a remarkably short period of time, the entire network may become congested. Hot spots are particularly insidious because they may result from the cumulative effects of very small traffic imbalances. Therefore, hot-spot is the most imbalanced traffic pattern. Hot spots often occur because of the burst nature of program communication and data requirements.

To show the adversity and congestion, we have plotted the traffic distribution using tornado, center-reflection, bit-flip, and perfect shuffle traffic pattern on a 8×8 mesh network in Figures 3, 4, 5, and 6, respectively. It is seen that in tornado traffic pattern, the lower dimension of the the network is congested. The tornado traffic pattern is designed as an adversary for torus network. It is seen in Fig. 4 that in center-reflection traffic pattern all the packets crosses the bisection of the network. Thus, the middle of the network is congested. It means that center-reflection traffic pattern is the most congested traffic pattern. In Fig. 5, it is seen that the center nodes, top-left node, and bottom-right node are congested. In Fig. 6, it is seen that the packets are distributed almost all the nodes. The congestion is amortized in the all nodes of the network. Now, let us see that how these adversity of traffic patterns affect the dynamic communication performance of various networks in the following subsection.

D. Dynamic Communication Performance Evaluation

We have evaluated the dynamic communication performance of several networks using dimension-order routing under five different traffic patterns: hot-spot, tornado, center reflection, bit-flip, and perfect shuffle.

1) Hot-Spot Traffic: For generating hot spot traffic we used a model proposed by Pfister and Norton [24]. Each node first generates a random number. If that number is less than a predefined threshold, the message will be sent to the hot-spot node. Otherwise, the message will be sent to other nodes, with a uniform distribution. In uniform distribution, message destinations are chosen randomly with equal probability among the nodes in the network.

Figure 7 depicts the message latency versus network throughput curves for various hot-spot traffic percentage. Figure 7(a), (b) and (c) represent the result of simulations using 5% 10%, and 15% hot spot traffic, respectively. It is shown that the average transfer time of the HTN is far lower than that of the mesh and torus networks, and a significantly lower than that of H3D-mesh network. One interesting point to be observed in Figure 7 is that the relative difference in maximum throughput between torus and HTN increases with the increase of hot-spot traffic. Therefore, with the most imbalanced hot-spot traffic pattern, HTN results better dynamic communication performance than that of the other hierarchical and conventional networks.

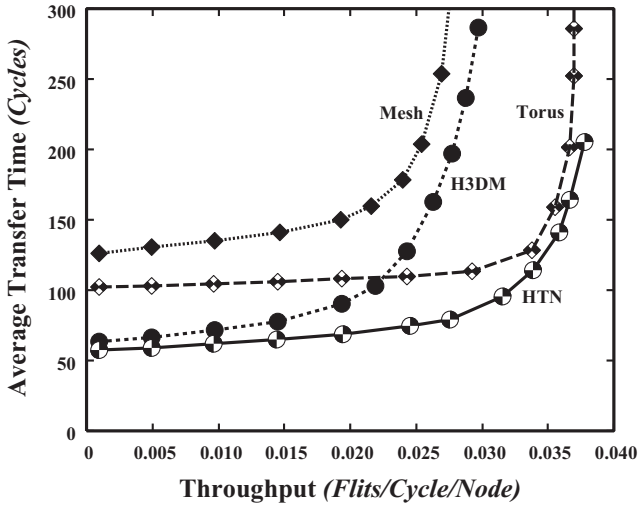
2) Tornado Traffic: HTN is a hierarchical interconnection network of torus-torus combination, and tornado traffic is applied in every level of the HTN. In the BM, Node(x, y, z) only sends packets to Node$\{(x + [k/2] - 1) \bmod k, y, z\}$. In Level-2 network, BM$(x, y)$ sends packet to BM$\{(x + [k/2] - 1) \bmod k, y\}$.

The dynamic communication performance of various networks under the tornado traffic pattern is portrayed in Fig. 8 for short, medium, and long message in (a), (b), and (c), respectively. The figure shows the average transfer time as a function of network throughput. Each curve stands for a particular network. From Fig. 8, it is seen that the average transfer time of the HTN is far lower than that of the mesh and torus networks, and noticeably lower than that of H3D-mesh network. The maximum throughput of the HTN is far higher than that of conventional mesh & torus and hierarchical H3D-mesh networks. HTN achieves better dynamic communication performance than the H3D-mesh, mesh, and torus networks. It is also shown that the throughput of the HTN is increasing with the increase of message length and the relative difference between maximum throughput of HTN and other network is increasing with the increase of message length.

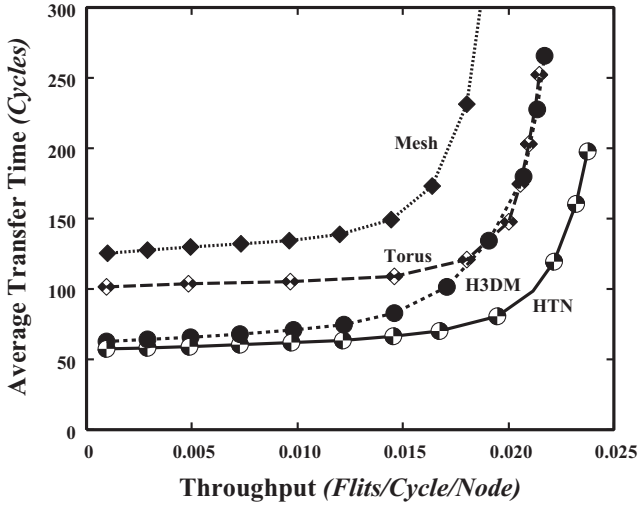
3) Center Reflection Traffic: In center-reflection traffic pattern, a source at (x, y, z) sends a message to a destination at $(k-x-1, k-y-1, k-z-1)$, where k is the number of node in one direction. The source $BM(x, y)$ sends a message to a destination at $BM(k - x - 1, k - y - 1)$, where k is the number of BMs in one direction of a Level-2 HTN.

Figure 9 depicts the results of simulations under center-reflection traffic pattern of the various networks. From Fig. 9 (a), (b), & (c), for both short, medium, and long message, respectively. It is seen that the average transfer time of the HTN is far lower than that of the mesh and significantly lower than H3D-mesh network. Usually the average transfer time at zero load called zero load latency of the HTN is less than that of the H3D-mesh networks; but the difference is not impressive for other traffic patterns. However, in center-reflection traffic pattern, the zero load latency of the HTN is remarkably lower than that of the H3D-mesh network.

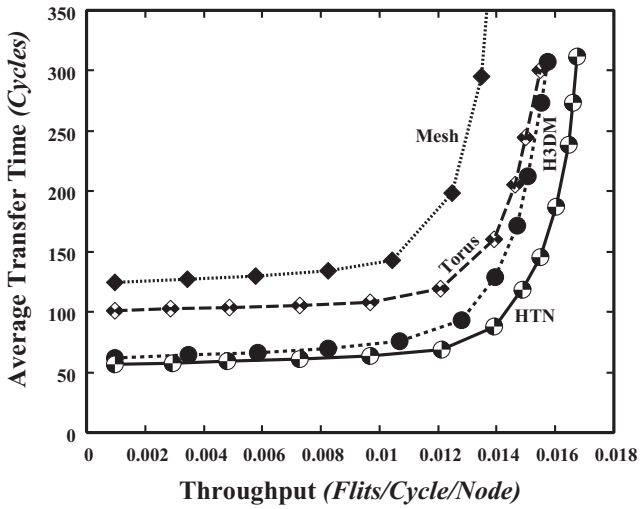
The maximum throughput of the HTN is far higher than that of mesh and H3D-mesh networks. HTN achieves better dynamic communication performance than the other conventional mesh and hierarchical H3D-mesh networks under the center-reflection traffic pattern. In center-reflection traffic patterns, 100% packets cross the bisection of the network, and thus the middle of the network



(a) 5% Hot-Spot Traffic

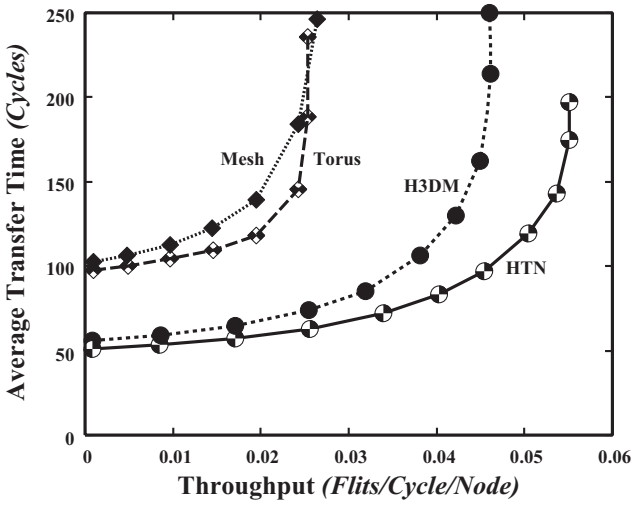


(b) 10% Hot-Spot Traffic

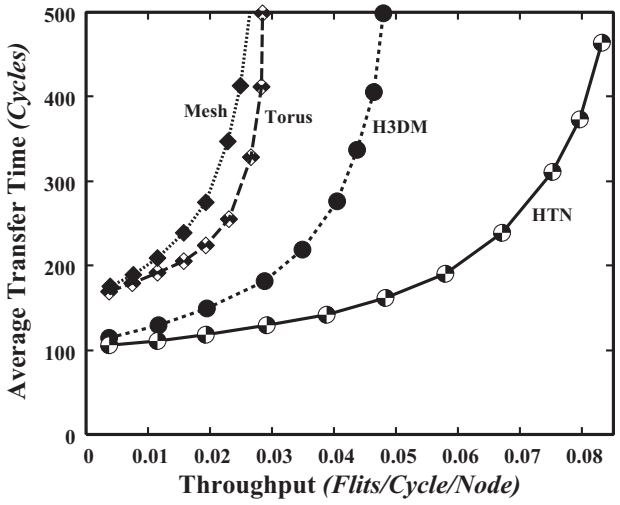


(c) 15% Hot-Spot Traffic

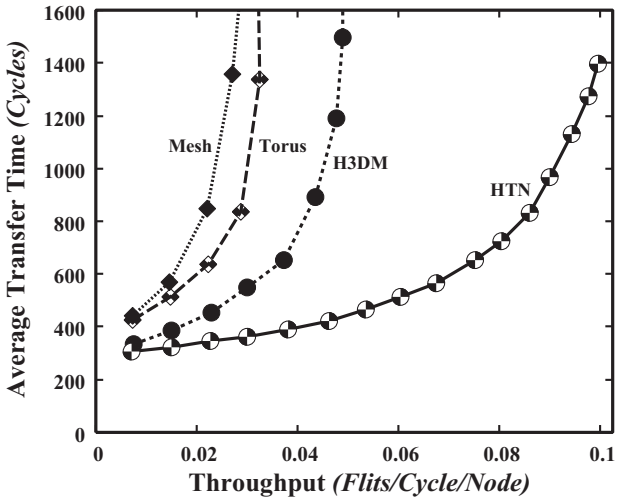
Figure 7. Dynamic communication performance of various networks using dimension-order routing with hot-spot traffic pattern: 1024 nodes, 3 VCs, 16 flits, and 2 buffers



(a) Short Message

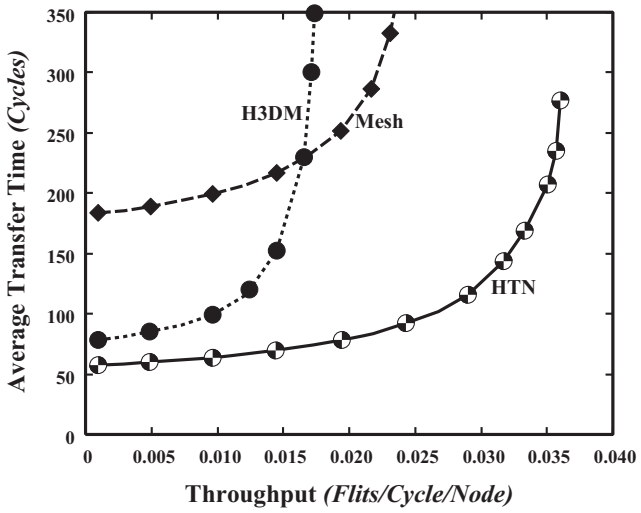


(b) Medium Message

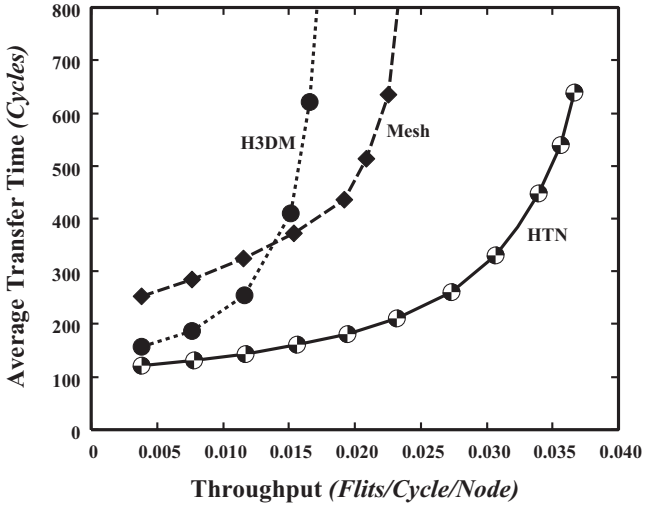


(c) Long Message

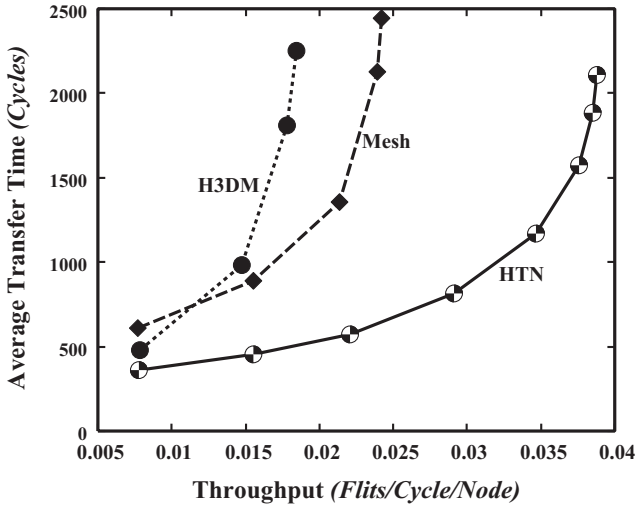
Figure 8. Dynamic communication performance of various networks using dimension-order routing with tornado traffic pattern: 1024 nodes, 3 VCs, 16 flits, and 2 buffers



(a) Short Message



(b) Medium Message



(c) Long Message

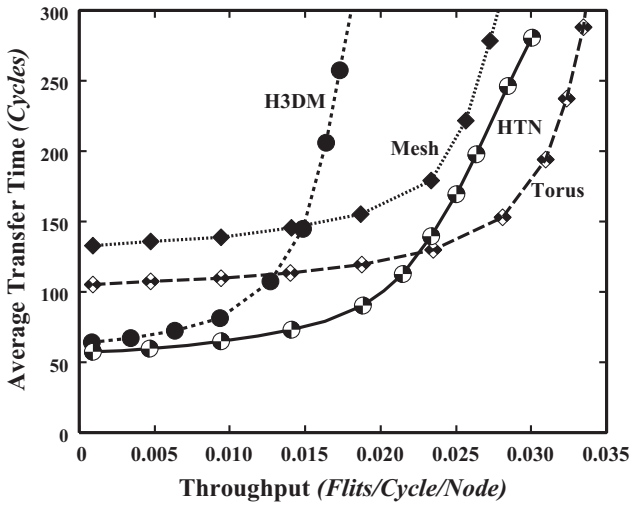
Figure 9. Dynamic communication performance of various networks using dimension-order routing with center reflection traffic pattern: 1024 nodes, 3 VCs, 16 flits, and 2 buffers

is heavily congested when the traffic is increased. Therefore, even with the most congested traffic pattern (center reflection) HTN yields better dynamic communication performance than that of other networks.

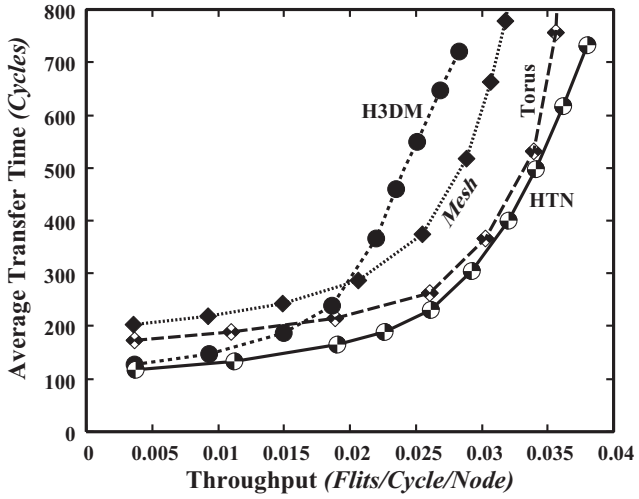
4) Bit-Flip Traffic: In a bit flip traffic pattern, a node with address $\mathrm{Node}\,(b_{\beta-1}, b_{\beta-2}\ \ldots\ \ldots\ b_1, b_0)$ sends messages to $\mathrm{node}\left(\overline{b_0}, \overline{b_1},\ \ldots\ \ldots\ \overline{b_{\beta-2}}, \overline{b_{\beta-1}}\right)$. Figure 10 portrays the result of simulations under bit flip traffic pattern for the various networks for short, medium, and long message in Fig. 10 (a), (b), and (c), respectively. It is seen that the average transfer time of the HTN is far lower than that of the mesh & torus networks and significantly lower than H3D-mesh network. The maximum throughput of the HTN is higher than that of the mesh and H3D-mesh networks; however, it is lower than that of the torus networks for short message as shown in Fig. 10(a). With the increase of message length, from short to medium or long, the maximum throughput of the HTN is higher than that of torus network. Also the difference is increasing with the increase of message length from medium to long message as shown in Fig. 10(c).

5) Perfect Shuffle Traffic: In perfect-shuffle traffic, the node with binary coordinates $b_{\beta-1}, b_{\beta-2}\ \ldots\ \ldots\ b_1, b_0$ communicates with the $\mathrm{node}\,(b_{\beta-2}, b_{\beta-3},\ \ldots\ \ldots\ b_1, b_0, a_{\beta-1})$. Figure 11 portrays the results of simulations under perfect-shuffle traffic pattern of the various networks. It is seen that the average transfer time of the HTN is lower than that of conventional mesh & torus networks and hierarchical H3D-mesh networks. The maximum throughput of the HTN is higher than that of torus and H3D-mesh. However, it is lower than that of mesh network for short message and almost equal to that of mesh network for medium message. If we increase the message length to long message, then it is seen that the maximum throughput of the HTN is higher than that of the mesh network. Therefore, HTN yields high dynamic communication performance than mesh, torus, and H3D-mesh networks under the perfect-shuffle traffic.

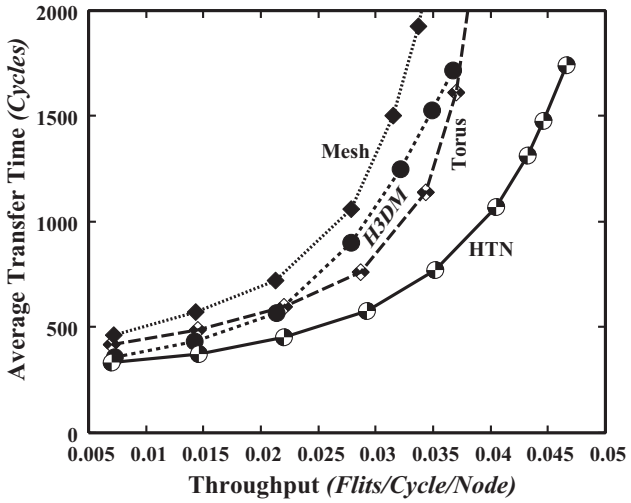
We have created the adverse traffic by some synthetic traffic patterns, where most of the packets crossing the bisection of the network as shown in Figures 3, 4, 5, and 6 and most imbalanced traffic hot-spot traffic pattern. From the dynamic communication performance it is seen that the maximum throughput of the HTN is higher than other networks under most imbalanced traffic patterns shown in Fig. 7. It is higher than that of the other networks under most congested traffic patterns shown in Fig. 9. In the tornado traffic patterns, the traffic is congested only in the lower dimension and in the bit-flip traffic patterns, the traffic is congested almost in the down diagonal line as shown in Fig. 3 and 5, respectively. In these congested traffic patterns too, HTN yield high throughput as shown in Figures 8 and 10, respectively. In the perfect-shuffle traffic patterns, the traffic is distributed almost all part of the network, i.e., the congestion is amortized in all part of the network. In this pattern, HTN yields trivially high maximum throughput than that of other networks



(a) Short Message

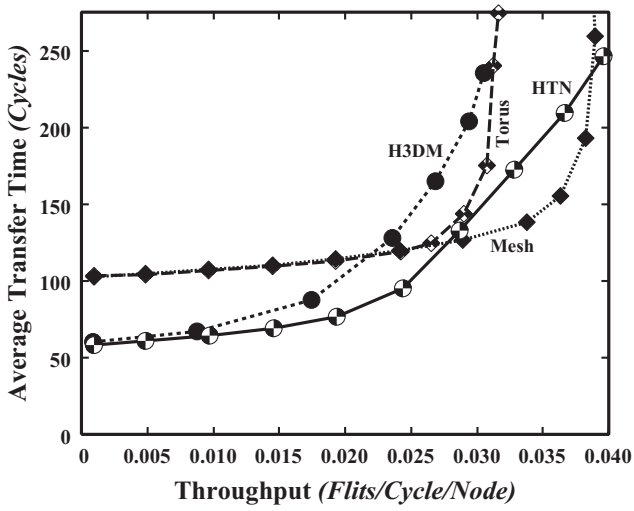


(b) Medium Message

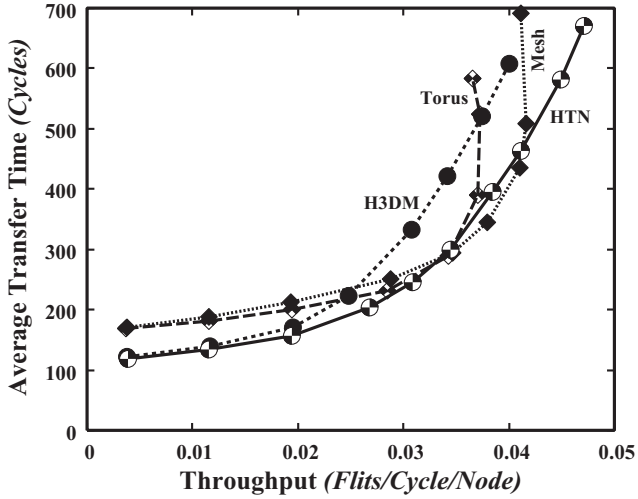


(c) Long Message

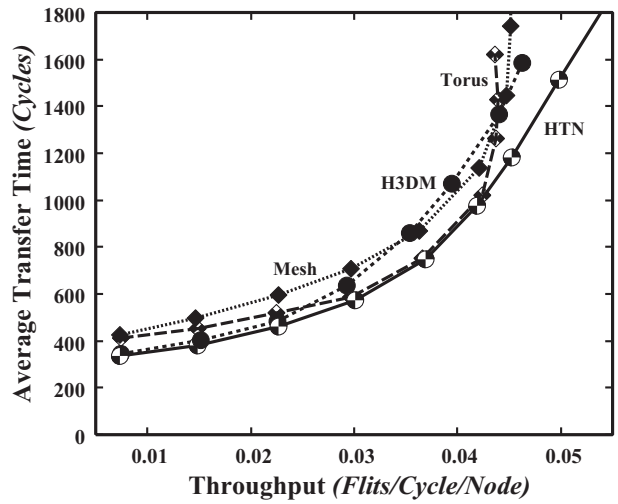
Figure 10. Dynamic communication performance of various networks using dimension-order routing with bit-flip traffic pattern: 1024 nodes, 3 VCs, 16 flits, and 2 buffers



(a) Short Message



(b) Medium Message



(c) Long Message

Figure 11. Dynamic communication performance of various networks using dimension-order routing with perfect-shuffle traffic pattern: 1024 nodes, 3 VCs, 16 flits, and 2 buffers

shown in Fig. 11. Also with the increase of message length, HTN yields high throughput. Therefore, HTN yields better dynamic communication performance under adverse traffic situation.

In our another study for TESH network yet to be published, we have used the clock time selected by minimum cycle time through hardware implementation using VHDL to evaluate the dynamic communication performance. In this paper, the dynamic communication is evaluated using simulation clock cycle to show the superiority of the HTN over other networks under adverse traffic patterns.

V. Conclusion

By using the routing algorithm described in this paper and some adverse traffic patterns, we have evaluated the dynamic communication performance of the HTN as well as that of several other commonly used networks and hierarchical interconnection networks. We have investigated the impact of adverse traffic pattern on the dynamic communication performance of the HTN. The average transfer time of HTN is lower than that of the H3D-mesh, mesh, and torus networks under adverse traffic patterns. Maximum throughput of the HTN is also far higher than that of those networks under the same adverse situation. A comparison of dynamic communication performance reveals that the HTN outperforms the H3D-mesh, mesh, and torus networks under adverse traffic patterns because it yields low latency and high throughput, which are indispensable for high performance massively parallel computer systems. Also, the more adverse the traffic situation is, the better dynamic communication performance does the HTN yield.

This paper focuses on the dynamic communication performance of the HTN under some adverse traffic patterns. Issues for future work include the following: (1) investigation of embedding of other frequently used topologies onto the HTN and (2) replacement of the long length electronic links by optical links, i.e., to study the architecture and performance of opto-electronic (hybrid)-HTN [29].

Acknowledgment

The author was a Post Doctoral researcher at JAIST supported by JSPS. This work is supported in part by JSPS fellowship program and Grand-in-Aid for Scientific Research (B), 21-09058, JSPS, Japan. The authors are grateful to the anonymous reviewers for their constructive comments which helped to greatly improve the clarity of this paper. The preliminary version of this paper has been published in the proceedings of the 13^{th} ICCIT, pp. 210 - 215, Dhaka, Bangladesh, 2010 [22].

References

[1] W.J. Dally, Performance analysis of k-ary n-cube interconnection networks, *IEEE Trans. Computers*, Vol. 39, No. 6, pp. 775–785, 1990.

[2] Y.R. Potlapalli, Trends in interconnection network topologies: Hierarchical networks, *Int'l. Conf. on Parallel Processing*, pp. 24–29, 1995.

[3] M. Abd-El-Barr and T.F. Al-Somani, Topological Properties of Hierarchical Interconnection Networks: A Review and Comparison, *Journal of Electrical and Computer Engineering*, Hindawi Publishing Corporation, Vol. 2011, 12 pages.

[4] L.N. Bhuyan and D.P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, *IEEE Trans Computers*, Vol. 33, No. 4, pp 323–333, 1984.

[5] P.L. Lai, H.C. Hsu, C.H. Tsai, I.A. Stewart, A class of hierarchical graphs as topologies for interconnection networks, *Theoretical Computer Science, Elsevier*, Vo. 411, pp. 2912–2924, 2010.

[6] Youyao Liu, Cuijin Li, and Jungang Han, RTTM: A New Hierarchical Interconnection Network for Massively Parallel Computing, *Proc. of the HPCA*, LNCS 5938, pp. 264–271, 2010.

[7] J.M. Camara, M. Moreto, E. Vallejo, R. Beivide, J. M. Alonso, C. Martinez, and J. Navaridas, Mixed-radix Twisted Torus Interconnection Networks, *Proc. of the IEEE International Parallel and Distributed Processing Symposium*, pp.80, 2007.

[8] J.G. Lpez, M. Imine, R.C. Rumn, J.M. Pedersen, O.B. Madsen, Multilevel Network Characterization using Regular Topologies, *Computer Networks, Elsevier*, vol. 52, pp. 23442359, 2008.

[9] M.M. Hafizur Rahman and S. Horiguchi, HTN: A New Hierarchical Interconnection Network for Massively Parallel Computers, *IEICE Trans. on Inf. & Syst.*, vol.E86-D, no.9, pp. 1479-1486, 2003.

[10] M.M. Hafizur Rahman and S. Horiguchi, A deadlock-free routing algorithm using minimum number of virtual channels and application mappings for Hierarchical Torus Network, *IJHPCN*, vol. 4, no. 3/4, pp. 174-187, 2006.

[11] M.M. Hafizur Rahman and S. Horiguchi, High Performance Hierarchical Torus Network under Matrix Transpose Traffic Patterns, *Proc. of the 7^{th} ISPAN*, pp. 111–116, Hong Kong, PRC, 2004.

[12] L.M. Ni and P.K. McKinley, A Survey of Wormhole Routing Techniques in Direct Networks, *IEEE Computer*, vol.26, no.2, pp. 62-76, 1993.

[13] W.J. Dally, "Virtual-Channel Flow Control," IEEE Trans. Parallel and Distrib. Syst., vol.3, no.2, pp.194-205, 1992.

[14] W.J. Dally and C.L. Seitz, "Deadlock Free Message Routing in Multiprocessor Interconnection Networks," IEEE Trans. on Computers, C36, No. 5, pp. 547-553, 1987.

[15] M.M. Hafizur Rahman and S. Horiguchi, Dynamic Communication Performance of a Hierarchical Torus Network under Non-uniform Traffic Patterns, *IEICE Trans. on Inf. & Syst.*, vol.E87-D, no.7, pp.1887-1896, 2004.

[16] W. Feng and K.G Shin, The effect of virtual channels on the performance of wormhole algorithms in multicomputer networks, *UM directed Study Report*, May 1994.

[17] H.Sarbazi-Azad, L.M. Mackenzie, and M.O. Khaoua, The Effect of the Number of Virtual Channels on the Performance of Wormhole-Routed Mesh Interconnection Networks, *Proc. of the 16th UKPEW*, Glasgow, 2000.

[18] Xiaoding Zhang, System Effects of Interprocessor Communication Latency in Multicomputers, *IEEE Micro*, vol.11, no.2, pp. 12-55, 1991.

[19] R. Holsmark, S. Kumar, M. Palesi, and A. Mekia, HiRA: A Methodology for Deadlock Free Routing in Hierarchical Networks on Chip, *Proc. of the 3^{rd} ACM/IEEE NOCS*, pp. 2-11, 2009.

[20] Xin Yu and T.S. Li, On shortest path routing algorithm in crossed cube-connected ring networks, *Proc. of the CyberC*, pp. 348 - 354, 2009.

[21] M. Koibuchi, K. Anjo, Y. Yamada, A. Jouraku, and H. Amano, "A Simple Data Transfer Technique using Local Address for Networks-on-Chips," IEEE Transactions on

Parallel and Distributed Systems, Vol. 17, No. 12, pp. 1425-1437, 2006.
[22] M M Hafizur Rahman, Y. Sato, Y. Inoguchi, High Performance Hierarchical Torus Network Under Adverse Traffic Patterns, *Proc. of the* 13^{th} *ICCIT*, pp. 210-215, Dhaka, Bangladesh, 2010.
[23] S. Horiguchi, "New Interconnection for Massively Parallel and Distributed System," Research Report, Grant-in-Aid Scientific Research, pro. no.09044150, JAIST, pp. 1-72,1999.
[24] G.F. Pfister and V.A. Norton, "Hot Spot Contention and Combining in Multistage Interconnection Networks," IEEE Trans. Comput., vol. 34, no. 10, pp. 943-948, 1985.
[25] W.J. Dally and B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann Publishers, 2004
[26] L. Schwiebert and R. Bell, Performance Tuning of Adaptive Wormhole Routing through Selection Function Choice, Journal of Parallel and Distributed Computing, vol. 62, no. 7, pp. 1121-1141, 2002.
[27] H.H. Najaf-abadi and H. Sarbazi Azad, The Effects of Adaptivity on the Performance of the OTIS-Hypercube Under Different Traffic Patterns, *Proc. of IFIP Int'l. Conf. NPC2004*, LNCS, Springer, pp. 390–398, 2004.
[28] Jose Duato, Sudhakar Yalamanchili, and Lionel Ni, *Interconnection Network: an Engineering Approach*, IEEE CS Press, Los Alamitos, California, USA, 1997.
[29] L. Xiao and K. Wang, Reliable Opto-Electronic Hybrid Interconnection Network, *Proc. of the* 9^{th} *I-SPAN*, pp. 239-244, 2008.

M.M. Hafizur Rahman received his B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh, in 1996. He received his M.Sc. and Ph.D. degree in Information Science from the Japan Advanced Institute of Science and Technology (JAIST) in 2003 and 2006, respectively. Dr. Rahman is now an assistant professor in the Dept. of Computer Science, Kulliyyah of Information and Communication Technology (KICT), International Islamic University, Malaysia (IIUM), Malaysia. Prior to join in the IIUM, he was an associate professor in the Dept. of CSE, KUET, Khulna, Bangladesh. He was also a visiting researcher in the School of Information Science at JAIST and a JSPS postdoctoral research fellow at Graduate School of Information Science (GSIS), Tohoku University, Japan & Center for Information Science, JAIST , Japan in 2008 and 2009 & 2010-2011, respectively. His current research include parallel and distributed computer architecture, hierarchical interconnection networks and optical switching networks. Dr. Rahman is member of the IEB of Bangladesh.

Yukinori Sato received the BS, MS, and Ph.D. degree in Information Science from Tohoku University in 2001, 2003, 2006, respectively. From 2006, he engaged in embedded processor system design in Sendai Software Development center of FineArch Inc. and also became a joint research member at Tohoku University. From 2007, he has been working at Japan Advanced Institute of Science and Technology (JAIST) as an assistant professor. His research interests include high-speed and low-power computer architectures and reconfigurable computing. Dr. Sato is a member of the IEEE, ACM, IEICE and IPS of Japan.

Yasushi Inoguchi received his B.E. degree from Department of Mechanical Engineering, Tohoku University in 1991, and received MS degree and Ph.D. from Japan Advanced Institute of Science and Technology (JAIST) in 1994 and 1997, respectively. He is currently an Associate Professor of Center for Information Science at JAIST. He was a research fellow of the Japan Society for the promotion of Science from 1994 to 1997. He is also a researcher of PRESTO program of Japan Science and Technology Agency from 2002 to 2006. His research interest has been mainly concerned with parallel computer architecture, interconnection networks, GRID architecture, and high performance computing on parallel machines. Dr. Inoguchi is a member of IEEE and IPS of Japan.