IEICE TRANS. INF. & SYST., VOL.E92–D, NO.5 MAY 2009

PAPER

TTN: A High Performance Hierarchical Interconnection Network for Massively Parallel Computers

M.M. Hafizur RAHMAN[†a)], ***Nonmember***, **Yasushi INOGUCHI**[††], **Yukinori SATO**[††], ***and*** **Susumu HORIGUCHI**[†††], ***Members***

SUMMARY Interconnection networks play a crucial role in the performance of massively parallel computers. Hierarchical interconnection networks provide high performance at low cost by exploring the locality that exists in the communication patterns of massively parallel computers. A **T**ori connected **T**orus **N**etwork (**TTN**) is a 2D-torus network of multiple basic modules, in which the basic modules are 2D-torus networks that are hierarchically interconnected for higher-level networks. This paper addresses the architectural details of the TTN and explores aspects such as node degree, network diameter, cost, average distance, arc connectivity, bisection width, and wiring complexity. We also present a deadlock-free routing algorithm for the TTN using four virtual channels and evaluate the network's dynamic communication performance using the proposed routing algorithm under uniform and various non-uniform traffic patterns. We evaluate the dynamic communication performance of TTN, TESH, MH3DT, mesh, and torus networks by computer simulation. It is shown that the TTN possesses several attractive features, including constant node degree, small diameter, low cost, small average distance, moderate (neither too low, nor too high) bisection width, and high throughput and very low zero load latency, which provide better dynamic communication performance than that of other conventional and hierarchical networks.

key words: *interconnection network, TTN, static network performance, wormhole routing, deadlock-free routing, traffic patterns, dynamic communication performance*

1. Introduction

Interconnection networks are the key elements for building massively parallel computers [1]. In such computers, with millions of nodes, the large diameter of conventional topologies is completely infeasible. Hierarchical interconnection networks [2] are a cost-effective way to interconnect a large number of nodes. A variety of hypercube-based hierarchical interconnection networks have been proposed [3]–[7], but for massively parallel computer systems, the number of physical links becomes prohibitively large. To alleviate this problem, several *k*-ary *n*-cube based hierarchical interconnection networks, such as H3D-Mesh [8], [9], H3D-Torus [10], [11], MH3DT [15], and Cube Connected Cycles (CCC) [16] have been proposed. However, the dynamic communication performance of these networks is still very low, especially in terms of network throughput.

Manuscript received August 12, 2008.
Manuscript revised January 2, 2009.

[†]The author is with the Dept. of CSE, Khulna University of Engineering and Technology (KUET), Khulna – 9203, Bangladesh.
[††]The authors are with the Center for Information Science, JAIST, Nomi-shi, 923–1292 Japan.
[†††]The author is with the Graduate School of Information Science, Tohoku University, Sendai-shi, 980–8579 Japan.
a) E-mail: hafiz90305@gmail.com

DOI: 10.1587/transinf.E92.D.1062

A Tori connected mESH (TESH) network [12] is an interconnection network aiming for large-scale 3D massively parallel computers, consisting of multiple basic modules (BMs) which are 2D-mesh networks. The BMs are hierarchically interconnected by a 2D-torus to build higher level networks. The restricted use of physical links between BMs in the higher level networks and within the BMs reduces the dynamic communication performance of this network [14]. With the increase of inter-level connectivity, the dynamic communication performance of the TESH network is shown to be better than that of a mesh network. However, it is still not as good as that of torus and hierarchical H3D-Torus, and MH3DT networks [15] with 2 virtual channels.

Our main objective is to find a network which is suitable for interconnecting a large number of nodes while maintaining good dynamic communication performance. It has already been shown that a torus network has better dynamic communication performance than a mesh network [1]. To fulfill our objective, with this key motivation, we have replaced the 2D-mesh of a TESH network by a 2D-torus network. That is, we made a hierarchical network as torus-torus combination. The modified TESH network consists of BMs which are themselves 2D-tori, hierarchically interconnected by 2D-torus networks. Analogous to the TESH network, we called it **T**ori-connected **T**orus **N**etwork (**TTN**). TTN is a hierarchical interconnection network, thus allowing exploitation of computation locality, as well as providing sacalability up to a million of nodes.

In massively parallel computers, an ensemble of nodes works in concert to solve large application problems. The nodes communicate data and coordinate their efforts by sending and receiving messages in the massively parallel computer through a router, using a routing algorithm. The routing algorithm specifies how a message selects its network path to move from source to destination. Efficient routing is critical to the performance of interconnection networks. In a practical router design, the routing decision process must be as fast as possible to reduce network latency.

Wormhole routing [17], [18] has become the dominant switching technique used in contemporary massively parallel computer systems. This is because it has low buffering requirements, and more importantly, it makes latency independent of the message distance. 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 [19], [20]

were originally introduced to solve the problem of deadlock in wormhole-routed networks.

Deterministic, dimension-order routing has been popular in massively parallel computers because it has minimal hardware requirements and allows the design of simple and fast routers [19]. Although there are numerous paths between any source and destination, dimension-order routing defines a single path from source to destination. If that selected path is congested, the traffic between that source and the destination is delayed, despite the presence of uncongested alternative paths. However, it is still very popular because of its low cost and simple router design. This is why most existing parallel computers, such as J-machine, Touchstone, Ametek 2010, and Cosmic cube, use deterministic routing.

In this paper, we address the architectural details of the TTN and evaluate its static network performance and dynamic communication performance. The static network performance is evaluated in terms of node degree, network diameter, cost, average distance, arc connectivity, bisection width, and wiring complexity. The dynamic communication performance is drastically reduced if a deadlock occurs in the routing of messages in a massively parallel computer system. Therefore, we propose a deadlock-free routing algorithm for the TTN, evaluate the dynamic communication performance using the proposed routing algorithm with various traffic patterns, and show the superiority of the TTN over several other networks.

The remainder of the paper is organized as follows. In Sect. 2, we briefly describe the network structure and addressing of nodes of the TTN. The dynamic routing algorithm is proposed in Sect. 3 and its freedom from deadlock is also proved. Static network performance and the dynamic communication performance are discussed in Sect. 4 and Sect. 5, respectively. Finally, in Sect. 6, we conclude this paper.

2. Interconnection of the TTN

2.1 Architecture of TTN

The *Tori-connected Torus Network (TTN)* is a hierarchical interconnection network consisting of Basic Modules (BM) that are hierarchically interconnected to form a higher level network. The BM of the TTN is a 2D-torus network of size $(2^m \times 2^m)$. In this paper, unless specified otherwise, BM refers to a Level-1 network. Successively higher level networks are built by recursively interconnecting immediately lower level subnetworks in a 2D-torus network. A higher-level network is built using a 2D-toroidal connection among (2^{2m}) immediate lower level subnetworks. The architecture of the TTN is described using the following two definitions.

Definition 1: A $2^m \times 2^m$ BM consists of a 2D-torus network of 2^{2m} processing elements (PE) having 2^m rows and 2^m columns, where m is a positive integer.

Definition 2: A TTN(m, L, q), which by definition is constructed using $2^m \times 2^m$ basic modules, has L levels of hierarchy. The last parameter q is the inter-level connectivity.

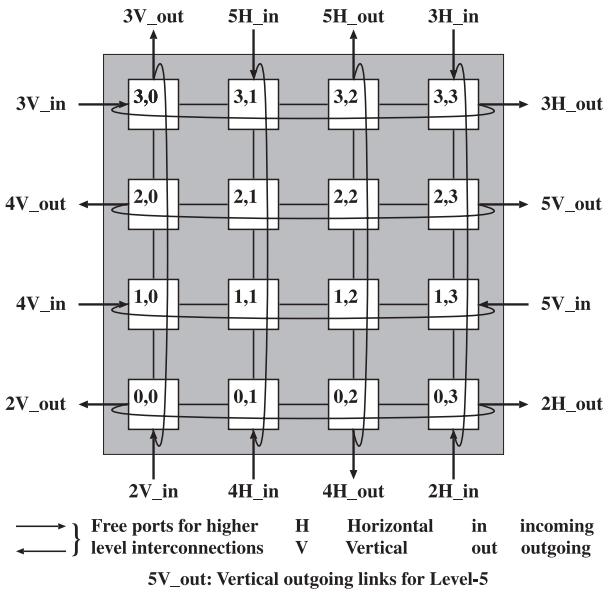


Fig. 1 Basic module of the TTN.

If $m = 2$, the size of the BM is (4×4), and in this paper, we focus attention on $m = 2$ i.e., (4×4) BMs. A BM of (4×4) is shown in Fig. 1. As seen in the figure, the BM has some free ports in the periphery for higher level interconnection. All ports of the interior Processing Elements (PEs) are used for intra-BM connections. All free ports of the exterior PEs, either one or two, are used for inter-BM connections to form higher level networks.

In principle, m could be any positive integer value. However, if $m = 1$, then the network degenerates to a binary hypercube network. Hypercube is not a suitable network, because its node degree increases along with the increase of network size. If $m = 2$, then it is considered the most interesting case, because it has better granularity than the large BMs. If $m = 3$, then the size of the BM becomes (8×8) with 64 nodes. Correspondingly, the Level-2 network would have 64 BMs. In this case, the total number of nodes in a Level-2 network is $N = 2^{2\times3\times2} = 4096$ nodes, and Level-3 network would have 262144 nodes. Clearly, the granularity of the family of networks is rather coarse. In addition, the matters of redundancy and reconfiguration become more difficult. Redundancy and reconfiguration are beyond the scope of this paper. In the rest of this paper we consider $m = 2$, therefore, we focus on a class of TTN$(2, L, q)$ networks. Several lemmas are stated below without proof. The proofs are straightforward and are omitted for brevity.

Lemma 1: A $2^m \times 2^m$ BM has 2^{m+2} free ports at the contours for higher level interconnection.

It is useful to note that for each higher level interconnection, a BM uses $4\times(2^q) = 2^{q+2}$ of its free links, $2(2^q)$ free links for vertical interconnections and $2(2^q)$ free links for horizontal interconnections. Here, $q \in \{0, 1, ..., m\}$, is the

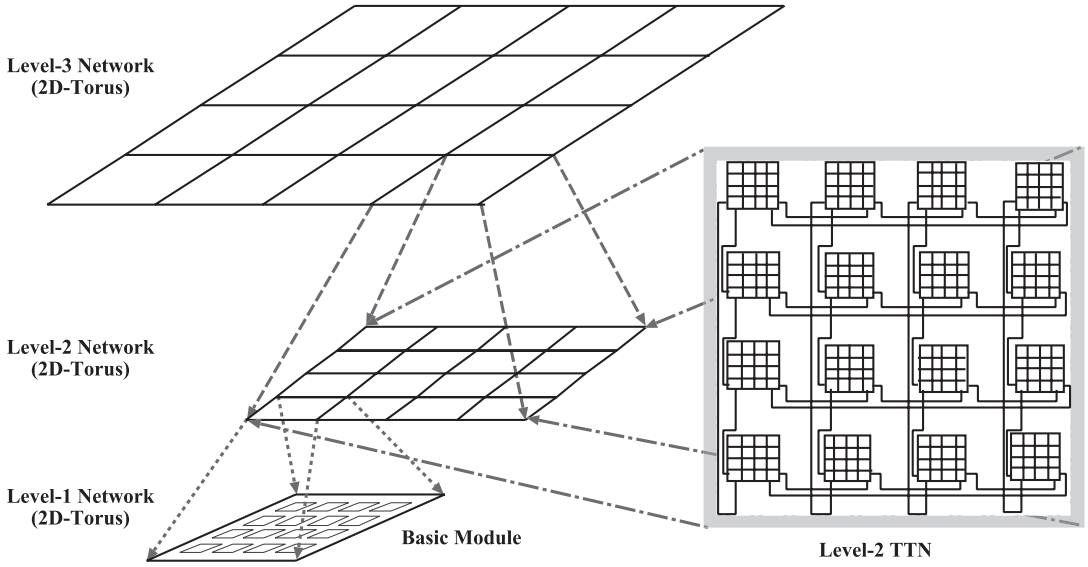


Fig. 2 Interconnection of a TTN.

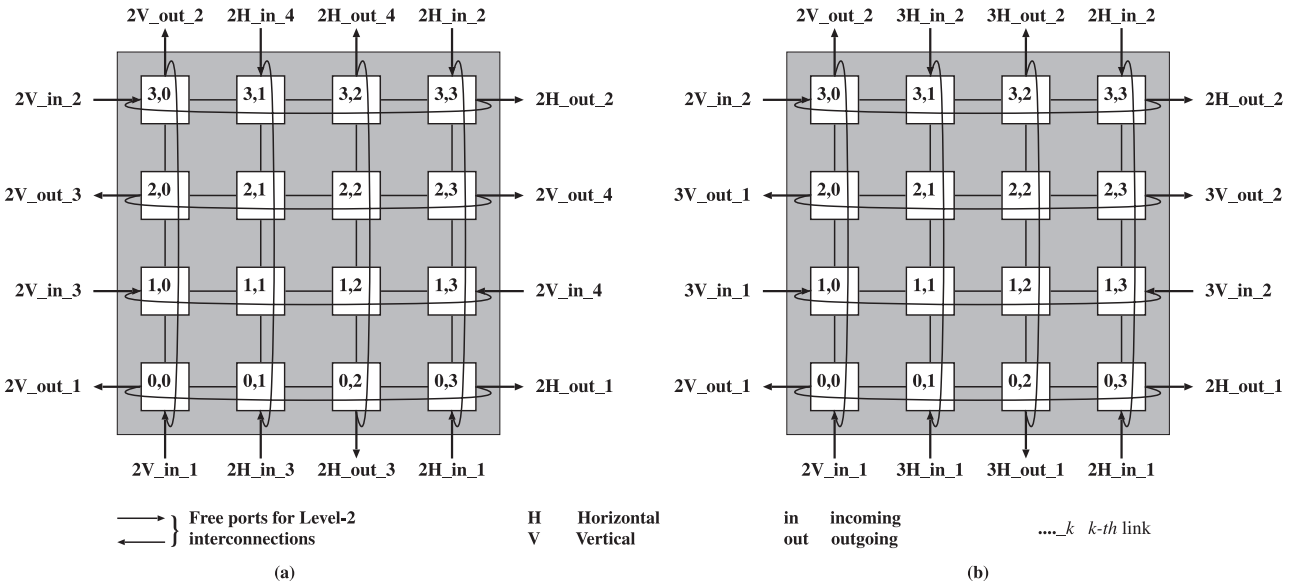


Fig. 3 A 4×4 basic module customized for: (a) a TTN(2, 2, 2) and (b) a TTN(2, 3, 1).

inter-level connectivity. $q = 0$ leads to minimal inter-level connectivity, while $q = m$ leads to maximum inter-level connectivity. As shown in Fig. 1, for example, the (4×4) BM has $2^{2+2} = 16$ free ports. If we chose $q = 0$, then $4(2^0) = 4$ of the free ports and their associated links are used for each higher level interconnection, 2 for horizontal and 2 for vertical interconnection. Among these 2 links, one is used for incoming link and another one for used for outgoing link, i.e., a single links is used for vertical_in, vertical_out, horizontal_in, and horizontal_out.

However, if the value of L is only 2 or 3, then we have the option of using more than one link for the vertical_in, vertical_out, horizontal_in, and horizontal_out connection, this means that the inter-level connectivity (q) is increasing. If $L = 2$, then 4 links can be used for each of the vertical_in, vertical_out, horizontal_in, and horizontal_out connections, so 16 links are used in Level-2 interconnection, as shown in Fig. 3 (a). Here the inter-level connectivity, $4(2^q) = 16$, $\Rightarrow q = 2$. Figure 3 (a) shows a TTN(2,2,2). Note that the last digit in the labels denotes the link number, for example, 2V_out_k indicates link number k for Level-2 vertical connections.

Similarly, if $L = 3$, then 2 links can be used for each vertical_in, vertical_out, horizontal_in, and horizontal_out connections, i.e., q becomes 1. Figure 3 (b) shows the TTN(2,3,1). With the increase of inter-level connectivity, the network diameter decreases, while the bisection width increases.

Lemma 2: The highest level network which can be built

from $(2^m \times 2^m)$ BM is $L_{max} = 2^{m-q} + 1$.

With $q = 0$, for example, $L_{max} = \left(2^{2-0} + 1\right) = 5$. Level-5 is the highest possible level for (4×4) BM interconnection. The limitation of having a maximum level of hieraqrchy is not a serious constraint. For the case of (4×4) BM with $q = 0$, a network built with the highest level, Level-5, consists of 1 million PEs. Successive higher level networks are built recursively by interconnecting the immediately lower level sub-networks. A higher-level network having $(2^m \times 2^m)$ BM is built using a (2^{2m}) subnetworks using a 2D-toroidal connection. For example, considering $(m = 2)$ a Level-2 subnetwork, can be formed by interconnecting $2^{2\times2} = 16$ BMs. Similarly, a Level-3 network can be formed by interconnecting 16 Level-2 subnetworks, and so on. This phenomena is illustrated in Fig. 2, a Level-2 TTN can be formed by interconnecting 16 BMs as a (4×4) 2D-torus network. Each BM is connected to its logically adjacent BMs. To avoid clutter, the wraparound links of the BMs are not shown.

Note that the choice of (2^{2m}) subnetworks to built the higher level networks is natural. This choice maintains the regularity of the network structure and thus makes the addressing of the nodes more convenient.

Lemma 3: The total number of nodes in a TTN having $(2^m \times 2^m)$ BMs is $N = 2^{2mL}$. Using maximum level of hierarchy, $L_{max} = (2^{m-q} + 1)$, the maximum number of nodes which can be interconnected by a TTN(m, L, q) is $N = 2^{2m(2^{m-q}+1)}$.

The question may arise, whether we need massively parallel computers with thousands of nodes or millions of nodes. The answer is 'yes'. Solving the most challenging problems in many areas of science and engineering, such as defense (maintaining national security), aerospace (space exploration and shuttle operation), disaster management (recovering from natural disaster), and weather forecasting (predicting and tracking severe weather), requires teraflop performance for more than a thousand hours at a time. This is why, in the near future, we will need computer systems capable of computing at the tens of petaflops level or even exaflops level. To achieve this level of performance, we need massively parallel computers with thousands or millions of nodes.

2.2 Addressing of Nodes

Base-4 numbers are used for convenience of address representation. As seen in Fig. 1, nodes in the BM are addressed by two digits, the first representing the row index and the next representing the column index. More generally, in a Level-L TTN, the node address is represented by:

$$\begin{aligned} A &= A^L A^{L-1} A^{L-2} \ldots\ldots\ldots A^2 A^1 \\ &= a_{n-1}a_{n-2}a_{n-3} \ldots\ldots\ldots a_2 a_1 a_0 \\ &= a_{2L-1}a_{2L-2}\, a_{2L-3}a_{2L-4} \ldots\ldots\ldots a_3 a_2\, a_1 a_0 \\ &= (a_{2L-1}\, a_{2L-2})\ (a_{2L-3}\, a_{2L-4})\ \ldots\ldots \\ &\quad \ldots\ldots\ldots (a_3\, a_2)\, (a_1\, a_0) \end{aligned} \tag{1}$$

Here, the total number of digits is $n = 2L$, where L is the level number. A^L is the address of level L in row-major scheme, and $(a_{2L-1}a_{2L-2})$ is the co-ordinate position of Level-$(L-1)$ for Level-L network. Pairs of digits run from group number 1 for Level-1, i.e., the BM, to group number L for the L-th level. Specifically, l-th group $(a_{2l-1}\ a_{2l-2})$ indicates the location of a Level-$(l-1)$ subnetwork within the l-th group to which the node belongs; $1 \le l \le L$. In a two-level network the address becomes $A = (a_4\ a_3)\ (a_1\ a_0)$. The first pair of digits $(a_4\ a_3)$ identifies the BM to which the node belongs, and the last pair of digits $(a_1\ a_0)$ identifies the node within that BM.

The assignment of inter-level ports for the higher level networks has been done quite carefully so as to minimize the higher level traffic through the BM. The address of a node n^1 encompasses in BM_1 is represented as $n^1 = \left(a^1_{2L-1}a^1_{2L-2} \ldots\ldots\ldots a^1_3 a^1_2\ a^1_1 a^1_0\right)$. The address of a node n^2 encompasses in BM_2 is represented as $n^2 = \left(a^2_{2L-1}a^2_{2L-2} \ldots\ldots\ldots a^2_3 a^2_2\ a^2_1 a^2_0\right)$. The node n^1 in BM_1 and n^2 in BM_2 are conected by a link if the following condition is satisfied.

$$\begin{aligned} &\exists i\{a^1_i = (a^2_i \pm 1) \bmod 2^m \\ &\wedge \forall j(j \neq i \ \rightarrow\ a^1_j = a^2_j)\} \end{aligned} \tag{2}$$

where $i, j \ge 2$

The inter-level links could be unidirctional or bidirectional. However, in our another study yet to be published, to implement the TTN in 3D integration, we have considered that the physical link between layer is very limited to be an unidirectional. In this paper, we have also considered the inter-level links are unidirectional.

3. Routing Algorithm for TTN

3.1 Dynamic Routing Algorithm

Routing of messages in the TTN is performed from top to bottom as in TESH network [22]. 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.

As mentioned earlier that dimension-order routing has been popular in massively parallel computer system because it has minimal hardware requirements and allows the design of simple and fast router. This is why, we have considered the dimension order routing algorithm for the TTN. We use the following strategy: at each level, vertical routing is performed first. Once the packet reaches the correct

Routing TTN(s,d);
source node address:$s_{2L-1}, s_{2L-2}, s_{2L-3}, ..., s_1, s_0$
destination node address: $d_{2L-1}, d_{2L-2}, d_{2L-3}, ..., d_1, d_0$
tag: $t_{2L-1}, t_{2L-2}, t_{2L-3}, ..., t_1, t_0$
  for $i = 2L - 1 : 2$
    if $\{(d_i - s_i + 2^m) \bmod 2^m\} \leq \frac{2^m}{2}$ then routedir = positive;
      $t_i = \{(d_i - s_i + 2^m) \bmod 2^m\}$;
    else routedir = negative;
      $t_i = \{2^m - (d_i - s_i + 2^m) \bmod 2^m\}$; endif;
    g = get_group_number$(s, d,$ routedir);
    while $(t_i \neq 0)$ do
      if $(i \bmod 2) = 0$, then
        outlet_node$_x$ = outlet_x$(g, \lfloor \frac{i}{2} + 1 \rfloor, H,$ routedir);
        outlet_node$_y$ = outlet_y$(g, \lfloor \frac{i}{2} + 1 \rfloor, H,$ routedir); endif;
      if $(i \bmod 2) = 1$, then
        outlet_node$_x$ = outlet_x$(g, \lfloor \frac{i}{2} + 1 \rfloor, V,$ routedir);
        outlet_node$_y$ = outlet_y$(g, \lfloor \frac{i}{2} + 1 \rfloor, V,$ routedir); endif;
      BM_Routing(outlet_node$_x$, outlet_node$_y$)
      if (routedir = positive), move packet to next BM; endif;
      if (routedir = negative), 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_y, t_x)
end
BM_Routing (t_1, t_0);
BM_tag t_1, t_0 = receiving node address (r_1, r_0) − destination (d_1, d_0)
  for $i = 1 : 0$
    if ($t_i > 0$ and $t_i \leq 2^{m-1}$) or ($t_i < 0$ and $t_i = -(2^m - 1)$), movedir = positive; endif;
    if ($t_i > 0$ and $t_i = (2^m - 1)$) or ($t_i < 0$ and $t_i \geq -2^{m-1}$), movedir = negative; endif;
    if (movedir = positive and $t_i > 0$), distance = t_i; endif;
    if (movedir = positive and $t_i < 0$), distance = $2^m + t_i$; endif;
    if (movedir = negative and $t_i < 0$), distance = t_i; endif;
    if (movedir = negative and $t_i > 0$), distance = $-2^m + t_i$; endif;
  endfor
  while($t_1 \neq 0$ or distance$_1 \neq 0$) do
    if (movedir = positive), move packet to $+y$ node; distance$_1$ = distance$_1 - 1$; endif;
    if (movedir = negative), move packet to $-y$ node; distance$_1$ = distance$_1 + 1$; endif;
  endwhile;
  while($t_0 \neq 0$ or distance$_0 \neq 0$) do
    if (movedir = positive), move packet to $+x$ node; distance$_0$ = distance$_0 - 1$; endif;
    if (movedir = negative), move packet to $-x$ node; distance$_0$ = distance$_0 + 1$; endif;
  endwhile;
end

Fig. 4 Routing algorithm of the TTN.

row, then horizontal routing is performed. Routing in the TTN is strictly defined by the source node address and the destination node address. Let a source node address be $s = (s_{2L-1}, s_{2L-2}), (s_{2L-3}, s_{2L-4}), ..., (s_3, s_2), (s_1, s_0)$, a destination node address be $d = (d_{2L-1}, d_{2L-2}), (d_{2L-3}, d_{2L-4}), ..., (d_3, d_2), (d_1, d_0)$, and a routing tag be $t = (t_{2L-1}, t_{2L-2}), (t_{2L-3}, t_{2L-4}), ..., (t_1, t_0)$, where $t_i = d_i - s_i$. Figure 4 shows the routing algorithm for the TTN. The function *get_group_number* gets a group number. Arguments of this function are s, d, and routing direction. Each free-link is labeled as (g, l, d, δ), where $2 \leq l \leq L$ is the level, $d \in \{V, H\}$ is the dimension, and $\delta \in \{+, -\}$ is the direction. The functions *outlet_x* and *outlet_y* results the outlet node of the BM for higher level.

Let us consider an example in which a packet is to be routed from source node 000000 to destination node 231112. In this case, routing is to be done at Level-3, therefore the source node sends the packet to the outlet node of Level-3, 00 00 30, whereupon the packet is routed at Level-3, as shown in Fig. 5. Here again, the wraparound links of Level-1, Level-2, and Level-3 networks are not shown to avoid clutter. After the packet reaches the Level-2$_(2, 3)$ network, then routing within that network continues until the packet reaches the BM$(1, 1)$. Finally, as shown in the

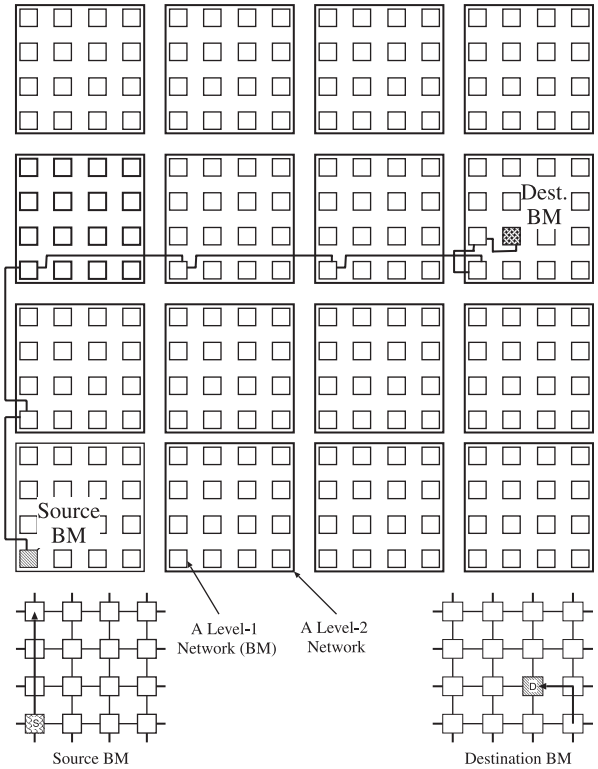


Fig. 5 Routing algorithm of the TTN.

bottom-right corner of Fig. 5, the packet is routed to its destination Node$(1, 2)$ within the destination BM. Here the strategy that the direction of communication of the higher level networks for vertical links is from bottom to top, and for horizontal links it is from left to right. In dimension-order routing, the routing path is determined by the source and destination node addresses. That is, the source and destination nodes are sufficient to determine the path traced by a packet. At each level, vertical routing (first in the y-axis) is performed first. Once the packet reaches the correct row, then the horizontal routing (routing in the x-axis) is performed.

3.2 Proof of Freedom from Deadlock

Routing algorithms for interconnection networks aim to minimize message blocking by efficiently utilizing network physical links and virtual channels while ensuring freedom from deadlock. Deadlock is the situation in which some packets can never advance because of blocking by other packets. If a deadlock occurs, packet delivery is delayed indefinitely. In addition to this, packet delivery rate is also reduced. In short, once a deadlock has occurred, the dynamic communication performance is drastically reduced, which is undesirable. A good routing algorithm for a wormhole-routed network must reduce message latency and increase network throughput as much as possible, with freedom from deadlock.

A deadlock-free routing can be constructed for arbi-

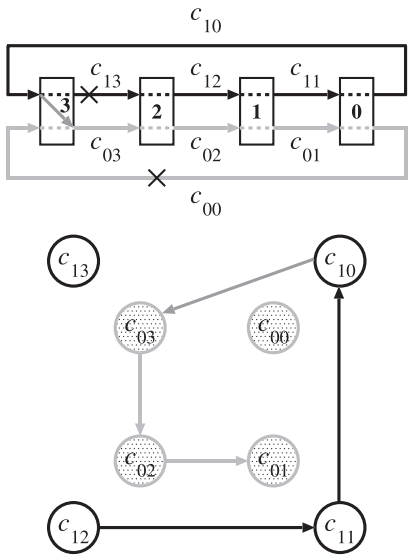


Fig. 6 Deadlock-free routing in a 4-node ring network.

trary interconnection networks by introducing virtual channels. In this section, we investigate the number of virtual channels required to make the routing algorithm for the TTN deadlock-free. Using the investigated number of virtual channels, we present a proof that the TTN is deadlock-free. To prove the proposed routing algorithm for the TTN is deadlock free, we divide the routing path into three phases, as follows:

- *Phase 1:* Intra-BM transfer path from source node to the outlet node 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 [19]. Since dimension-order routing is used in the TTN, routing of messages is first performed in the vertical direction, and then in the horizontal direction. The interconnection of the BM and the higher level network of TTN is a toroidal connection. Thus, to prove that the routing algorithm of the TTN deadlock-free, we will describe the deadlock-free routing of a torus network. Dimension order routing is deadlock-free in a network if and only if the channel dependency graph is acyclic [19]. As an example, we show a simple 4-node ring network in Fig. 6. Using 2 virtual channels is fairly easy here. Split each channel into upper virtual channels (C_{10}, ..., C_{13}), and lower virtual channels (C_{00}, ..., C_{03}). Whenever a packet is destined for a node which is in the descending order, the lower virtual channels are used. If the destination is in the upper position with respect to the source, then the packet starts moving left using upper channels until it reaches an end-to-end (wrap-around links connected) node, when it switches to lower channels and keeps using them until it reaches the destination.

An $n-$dimensional torus is the Cartesian product of a ring and $(n-1)-$dimensional sub-torus. Therefore, the idea of a ring network can be applied to a higher dimensional torus network. Informally, a deadlock-free routing algorithm for a torus network can be obtained if the messages are routed in order of descending node address, and the following two conditions are satisfied:

- when the nonzero offset in the most significant position is found by subtracting the current address from the destination.
- make a step toward nullifying the offset by sending the packet along that dimension in descending order, so that the upper virtual channel is used if the corresponding offset is greater than zero, and the lower virtual channel is used if the corresponding offset is less than zero.

A lemma and a corollary are stated below with proof. By using the following lemma and corollary, we will prove that the proposed routing algorithm for the TTN is deadlock-free.

Lemma 4: If a message is routed in the order $y \rightarrow x$ in a 2D-torus network, then the network is deadlock free with 2 virtual channels.

Proof: If the channels are allocated according to Eq. (3) for a 2D-torus network, and the messages are routed according to the above-mentioned phenomena, then cyclic dependency will not occur. Therefore, freedom from deadlock is proved. Initially, messages are routed over virtual channel 0 (lower). Then, messages are routed over virtual channel 1 (higher) if the message is going to use a wrap-around channel.

$$C = \begin{cases} (l, vc, a_1), & y+ \text{ channel}, \\ (l, vc, 2^m - a_1), & y- \text{ channel}, \\ (l, vc, a_0), & x+ \text{ channel}, \\ (l, vc, 2^m - a_0), & x- \text{ channel} \end{cases} \tag{3}$$

Here, $l = \{l_0, l_1, l_2, l_3\}$ are the links used in the BM, and $l = \{l_0, l_1\}$ and $l = \{l_2, l_3\}$ are the links used in the y–direction and x–direction interconnections, respectively. $vc = \{VC_0, VC_1\}$ are the virtual channels, 2^m is the size of the BM, and a_0 and a_1 are the node addresses in the BM.

Corollary 1: A higher level TTN is also a 2D-torus network, and is deadlock-free with 2 virtual channels.

Proof: If the channels are allocated as shown in Eq. (4) for the higher level 2D-torus network, then freedom from deadlock is proved.

$$C = \begin{cases} (l, vc, a_{2L-1}), & y+ \text{ channel}, \\ (l, vc, 2^m - a_{2L-1}), & y- \text{ channel}, \\ (l, vc, a_{2L-2}), & x+ \text{ channel}, \\ (l, vc, 2^m - a_{2L-2}), & x- \text{ channel} \end{cases} \quad (4)$$

Here, $l = \{l_4, l_5\}$ are the links used for higher-level interconnection, l_4 is used for vertical interconnection, and l_5 is used for horizontal interconnection. $vc = \{VC_0, VC_1\}$ are the virtual channels, m is the size of the BM, and a_{2L-1} and a_{2L-2} are the node addresses in the higher level, where L is the level number.

Theorem 1: A TTN with 4 virtual channels is deadlock free.

Proof: The BM and the higher levels of the TTN are toroidal interconnections. In phase-1 and phase-3 routing, packets are routed in the source-BM and destination-BM, respectively. The BM of the TTN is a 2D-torus network. According to Lemma 4, the number of necessary virtual channels for phase-1 and phase-3 is 2. The routings of the message in source-BM and destination-BM are carried out separately. The virtual channels required in *phase*-1 and *phase*-3 can share each other. The higher level network consists of inter-BM links and intra-BM links between inter-BM links. Intra-BM links between inter-BM links are used in sub-phases 2.i.1 and 2.i.3. Thus, sub-phases 2.i.1 and 2.i.3 utilize channels over intra-BM links, sharing the channels of either phase-1 or phase-3. The free links of the BM are used in inter-BM routing, i.e., sub-phases 2.i.2 and 2.i.4, and these links form a 2D-torus for the higher level network. According to Corollary 1, the number of virtual channels necessary for this 2D-torus network is also 2.

Therefore, the total number of virtual channels required to make the whole network deadlock free is 4.

4. Static Network Performance

The topology of an interconnection network determines many architectural features that affect several performance metrics. Although the actual performance of a network depends on many technological and implementation issues, several topological properties and performance metrics can be used to evaluate and compare different network topologies in a technology-independent manner. Most of these properties are derived from the graph model of the network topology. In this section, we discuss some of the properties and performance metrics that characterize the cost and performance of an interconnection network.

Comparing the performance of different hierarchical interconnection networks such as TTN, TESH [12], H3D-Torus [11], MH3DT [15], and CCC [16] networks is not an easy task, because each network has different architecture, which makes it difficult to match the total number of nodes. The total number of nodes in a TTN is $N = 2^{2mL}$. If $m = 2$ and $L = 3$ then the total number of nodes of the TTN is 4096. Level-3 TESH network has 4096 nodes. Level-2 MH3DT and H3D-Torus networks (when $m = 4$ and $n = 4$), 64×64 mesh network and 64×64 torus network all have 4096 nodes. The d-dimensional CCC network has $d \times 2^d$ nodes. If $d = 9$, then the total number of nodes of the CCC network is $9 \times 2^9 = 4608$. Due to the structure of the CCC network [16], it is not possible to construct a 4096-node CCC network. We have compared the static network performance and dynamic communication performance of various 4096-node networks. It has already been shown that the static network performance and dynamic communication performance of the MH3DT network is better than that of H3D-Torus network [15]. This is why, the performance of H3D-Torus network is not considered here. The static network performance of various networks with 4096 nodes, along with that of a CCC network with 4608 nodes, is tabulated in Table 1. The static network performance of Level-2 TTN and TESH network along with 16×16 mesh and 16×16 torus networks, all have 256 nodes, are also tabulated in Table 1.

The static network performance of the 4608-node CCC network can not be compared with the other 4096-node networks. However, its performance is included in Table 1 to show its topological properties.

4.1 Node Degree

The *node degree* is defined as the number of physical links emanating from a node. Since each node of the TTN has six links, its degree is 6. For the TTN, the node degree is independent of network size. Constant degree networks are easy to expand, and the cost of the network interface of a node remains unchanged with increasing size of the network. The I/O interface cost of a particular node is proportional to its degree. The degree of the TTN is higher than that of its counterpart TESH network but lower than that of MH3DT network.

4.2 Diameter

The *diameter* of a network is the maximum inter-node distance, i.e., the maximum number of links that must be traversed to send a message to any node along the shortest path. As a definition, the distance between adjacent nodes is unity. Diameter is the maximum distance among all distinct pairs of nodes along the shortest path. The diameter is commonly used to describe and compare the static network performance of the network's topology. Networks with small diameters are preferable. The smaller the diameter of a network, the shorter the time to send a message from one node to the node farthest away from it. In fact, the diameter sometimes (but not always) determines the lower bound for the running time of an algorithm performed on the network. Table 1 shows a comparison of the TTN diameter with the diameter of several other networks. Clearly, the TTN has a much smaller diameter than its rival TESH network, the conventional mesh and torus networks. Although MH3DT has a diameter comparable to that of the TTN, it requires more links.

The diameter of the TTN with $q = 0$ is calculated using

Table 1 Comparison of static network performance of various networks.

	Node Degree	Diameter	Cost	Average Distance	Arc Connectivity	Bisection Width	Wiring Complexity
256 Node							
2D-Mesh	4	30	120	10.67	2	16	480
2D-Torus	4	16	64	8	4	32	512
TESH (2,2,0)	4	21	84	10.47	2	8	448
TESH (2,2,1)	4	19	76	9.53	2	16	512
TESH (2,2,2)	4	16	64	7.80	2	32	640
TTN (2,2,0)	6	15	90	7.44	4	8	576
TTN (2,2,1)	6	13	78	6.34	4	16	640
4096 Node							
2D-Mesh	4	126	504	42.67	2	64	8064
2D-Torus	4	64	256	32	4	128	8192
MH3DT (4,4,2,2)	8	18	144	9.37	6	128	12864
TESH (2,3,0)	4	32	128	17.80	2	8	8192
TESH (2,3,1)	4	28	112	14.53	2	16	10240
TTN (2,3,0)	6	24	144	12.60	4	8	10240
TTN (2,3,1)	6	20	120	10.59	4	16	12288
CCC† (9×2^9)	3	22	66	12.75	3	256	6912

†CCC network with 4608 nodes

the following equations:

$$D_{TTN(m,L,0)} = D_{BM(s)} + \max \sum_{i=L}^{2} \left(D_{BM}^{level-move} + D_i \right) + D_{BM(d)}. \quad (5)$$

$D_{BM(s)} = 4$ is the maximum number of hops from the source to the highest level outgoing node. $D_{BM}^{level-move}$ is the number of hops for the immediate lower level outgoing node. D_i is the number of hops in the Level-i routing. $D_i = 7$ for Level-2 and Level-3 routing. $D_i = 13$ for Level-4 and Level-5 routing, because forward and backward nodes are separated by one node-distance by design. $D_{BM(d)} = 4$ is the maximum number of hops from the incoming nodes of destination BM to the destination. $D_{BM(s)}$ and $D_{BM(d)}$ are the diameter of a $(2^m \times 2^m)$ torus network.

4.3 Cost

Inter-node distance, message traffic density, and fault-tolerance are dependent on the diameter and the node degree. The product (*diameter* × *node degree*) is a good criterion for measuring the relationship between cost and performance of a multiprocessor system [5]. An interconnection network with a large diameter has a very low message passing bandwidth, and a network with a high node degree is very expensive. In addition, a network should be easily scalable; there should be no changes in the basic node configuration as we increase the number of nodes. Table 1 shows that the cost of the TTN is less than that of the conventional mesh and torus networks and hierarchical TESH and MH3DT networks. Due to high node degree, with the increase of q the cost of TTN is a little bit higher than that of a TESH network. The cost of TTN(2,2,1) and TTN(2,3,1) is a trivial higher than that of TESH(2,2,1) and TESH(2,3,1) networks, respectively.

4.4 Average Distance

The *average distance* is the mean distance between all distinct pairs of nodes in a network. A small average distance allows small communication latency, especially for distance-sensitive routing, such as store and forward. But it is also crucial for distance-insensitive routing, such as wormhole routing, since short distances imply the use of fewer links and buffers, and therefore less communication contention. We have evaluated the average distances for different conventional topologies by the corresponding formulae, and of different hierarchical networks by simulation. As shown in Table 1, the TTN has a smaller average distance than the conventional mesh and torus networks, and hierarchical TESH network. However, the average distance of the TTN is slightly higher than that of the MH3DT network. The average distance of the TTN(2,2,0) and TTN(2,3,0) are lower than that of TESH(2,2,0) and TESH(2,3,0) networks, respectively. Even it is less than that of TESH(2,2,1) and

TESH(2,3,1) networks.

4.5 Arc Connectivity

Connectivity measures the robustness of a network. It is a measure of the multiplicity of paths between processors. Arc connectivity is the minimum number of links that must be removed in order to break the network into two disjoint parts; it is a measure of connectivity. High connectivity (and thus high arc connectivity) improves performance during normal operation by avoiding link congestion, and also improves fault tolerance. A network is maximally fault-tolerant if its connectivity is equal to the degree of the network. The arc connectivity of various networks is shown in Table 1. Clearly, the arc connectivity of the TTN is higher than that of the mesh and TESH networks, and equal to that of the torus network. However, the arc connectivity of the torus network is exactly equal to its degree. Thus, torus is more fault tolerant than the TTN, and TTN is more fault tolerant than mesh and TESH networks.

4.6 Bisection Width

The *Bisection Width (BW)* of a network is defined as the minimum number of links that must be removed to partition the network into two equal halves. Many problems can be solved in parallel using *binary divide-and-conquer:* split the input data set into two halves, and solve them recursively on both halves of the interconnection network in parallel, then merge the results from both halves into the final result. Small bisection width implies low bandwidth between the two halves, and it can slow down the final merging phase. On the other hand, a large bisection width is undesirable for the VLSI design of the interconnection network, since it implies a lot of *extra chip wires*, such as in hypercube [12]. Table 1 shows that the bisection width of the TTN is lower than that of the mesh, torus, and MH3DT networks [15], and equal to that of the TESH network [12]–[14].

4.7 Wiring Complexity

The *wiring complexity* of an interconnection network refers to its total number of links. It has a direct correlation to hardware cost and complexity. For a Level-L TTN, the wiring complexity is represented by Eq. (6).

$$\left[k^{2(L-1)} \times \left\{2k^2 + 4(2^q)(L-1)\right\}\right] \tag{6}$$

where $k = 2^m$. Table 1 compares the wiring complexity of a TTN with that of several other networks. The total number of physical links in the TTN is higher than in the mesh, torus, and TESH [12] networks; therefore, the cost of physical links is higher for the TTN. But it is lower than that of the MH3DT network [15]. The cost of TTN(2,2,1) and TTN(2,3,1) with respect to the number of physical links is higher than that of TESH(2,2,1) and TESH(2,3,1) networks, respectively. This extra cost of physical links for the TTN yields better dynamic communication performance especially in terms of network throughput than that of the TESH network will be shown in the Sect. 5.

The operating speed of a network is limited by the physical length of links. The distance between two contiguous nodes is unity. With 2D-planar implementation, the maximum lengths of Level-2 and Level-3 TTN are 12 and 48, respectively. These are the wrap-around links of the higher level interconnection. The BM of TTN is a 2D-torus network. Thus, we need some more medium length links whose length is $2^m - 1$. The main demerit of TTN is that we need some medium and high length links. However, this cost yields better performance. To overcome this problem, we have kept in mind as future work, the replacement of the electronic links by optical links, i.e., to study the architecture and performance of opto-electronic-TTN or OTIS-TTN.

5. Dynamic Communication Performance

The overall performance of a massively parallel computer system is affected by the performance of the interconnection network, as well as by the performance of the nodes. Continuing advances in VLSI technologies promise to deliver more power to individual nodes. On the other hand, low performance of the communication network will severely limit the speed of the entire system. Therefore, the success of massively parallel computers is highly dependent on the efficiency of their underlying interconnection networks. The evaluation of dynamic communication performance of the TTN, along with several other networks, is described in this section.

5.1 Performance Metrics

The dynamic communication performance of a massively parallel computer is characterized by *message latency* and *network throughput*. Message latency is the time required for a packet to traverse the network from source to destination. It refers to the time elapsed from the instant when the first flit is injected to the network from the source, to the instant when the last flit of the message is received at the destination. In wormhole routing, it is the average value of the time elapsed between injection of the header flit into the network from the source, and reception of the last unit of the data flit at the destination. Latency is measured in time units. However, when comparing several design choices, the absolute value is not important; because the comparison is performed by computer simulation, latency is measured in simulator clock cycles.

Network throughput is the rate at which packets are delivered by the network for a particular traffic pattern. It refers to the maximum amount of information delivered per unit of time through the network. It also can be defined as the maximum traffic accepted by the network. Throughput depends on message length and network size. Therefore, throughput is usually normalized, dividing it by message length and network size. When throughput of various net-

works are compared by computer simulation and wormhole routing is used for switching, throughput can be measured in flits per node and per clock cycle.

For the network to have good performance, low latency and high throughput must be achieved. Zero-load latency is a lower bound on the average latency of a packet through the network. In zero-load, it is assumed that a packet never contends for network resources with other packets. Under this assumption, the average latency of a packet is its serialization latency plus its hop latency. Throughput is a property of the entire network, and depends on routing and flow control as much as on the topology. Maximum throughput is the upper bound throughput through a network. A resource is in saturation when the demands being placed on it are beyond its capacity for servicing those demands. A channel becomes saturated when the amount of data to be routed over the channel exceeds the bandwidth of the channel. The saturation throughput of a network is the smallest rate of traffic for which some channel in the network becomes saturated. If no channels are saturated, the network can carry more traffic. We also call this saturation throughput "maximum throughput".

5.2 Simulation Environment

To evaluate dynamic communication performance, we have developed a wormhole routing simulator. In our simulation, we use a dimension-order routing algorithm. The dimension-order routing algorithm, which is exceedingly simple, provides the only route for the source-destination pair. For all networks considered in this paper, four virtual channels per physical channel are simulated and the virtual channels are arbitrated by a round robin algorithm. For all of the simulation results, the packet size is 16 flits. Two flits are used as the header flit. In the evaluation of dynamic communication performance, flocks of messages are sent in the network to compete for the output channels. For each simulation run, we have considered that the message generation rate is constant and the same for all nodes. Flits are transmitted at $20,000$ cycles; in each clock cycle, one flit is transferred from the input buffer to the output buffer, or from output to input if the corresponding buffer in the next node is empty. Therefore, transferring data between two nodes takes 2 clock cycles.

Extensive simulations for several networks have been carried out under various traffic patterns: uniform [23], hot spot [24], matrix transpose [25], [26], bit-reversal [27], complement [28], bit-flip [29], and perfect shuffle [31].

5.3 Traffic Patterns

Traffic patterns are pairs of nodes that communicate. In an interconnection network, sources and destinations for messages form the traffic pattern. Traffic characteristics such as message length, message arrival times 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.

The most frequently used, simplest, and most elegant pattern is the uniform traffic pattern where the source and the destination are randomly selected. However, depending on the characteristics of an application, some nodes may communicate with each other more frequently than with others. Consequently, non-uniform traffic patterns are frequent, and cause uneven usage of traffic resources, significantly degrading the dynamic communication performance of the network.

When a hot spot occurs, 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. Hot spots often occur because of the burst nature of program communication and data requirements. Dimension order routing has only one route for a source-destination pair. Bit permutation and computation [30] is a class of non-uniform traffic patterns which are very common in scientific applications, where the source node sends messages to a predefined destination. Both dimension order routing and bit permutation & communication create significant congestion under dimension order routing in the network, and when congestion occurs, the network throughput decreases precipitously.

We have evaluated the dynamic communication performance of the TTN under the uniform and various non-uniform traffic patterns. A very brief description of each traffic pattern is given below.

- **Uniform** – In the uniform traffic pattern, every node sends messages to every other node with equal probability in the network. That is, source and destination are randomly selected.
- **Hot-Spot** – A hot spot is a node that is accessed more frequently than other nodes in the uniform traffic distribution. In hot-spot traffic pattern, each node generates a random number. If that number is less than a threshold, the message is sent to the hot spot node. Otherwise it is sent to other nodes with a uniform distribution.
- **Matrix Transpose** – In matrix-transpose traffic pattern, each node sends messages to a node with an address of the reversed dimension index, i.e., node(x, y) communicates with node(y, x).
- **Bit-reversal** – The binary representation of the node address is $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0. In bit-reversal traffic, the node $\left(b_{\beta-1}, b_{\beta-2} \text{ ... ... } b_1, b_0\right)$ communicates with the node $\left(b_0, b_1, \text{ ... } b_{\beta-2}, b_{\beta-1}\right)$.
- **Complement** – The binary representation of the node address is $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0. In complement traffic, the node $\left(b_{\beta-1}, b_{\beta-2} \text{ ... ... } b_1, b_0\right)$ communicates with the node $\left(\overline{b_{\beta-1}}, \overline{b_{\beta-2}}, \text{ ... } \overline{b_2}, \overline{b_1}, \overline{b_0}\right)$.
- **Bit-flip** – The node with binary coordinates $b_{\beta-1}, b_{\beta-2}$

... ... b_1, b_0 communicates with the node $\left(\overline{b_0}, \overline{b_1}, \text{ ... } \text{ ... } \overline{b_{\beta-2}}, \overline{b_{\beta-1}}\right)$. That is, complement of bit-reversal traffic.

- **Perfect Shuffle** – The node with binary coordinates $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0 communicates with the node $\left(b_{\beta-2}, b_{\beta-3}, \text{ ... ... } b_1, b_0, b_{\beta-1}\right)$. That is, rotate left 1 bit.

5.4 Dynamic Communication Performance Evaluation

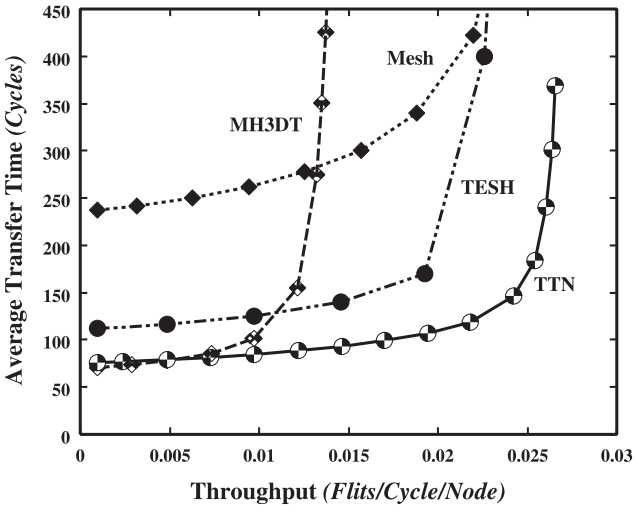
We have evaluated the dynamic communication performance of several 4096-node networks under various traffic patterns. We have evaluated the dynamic communication performance using dimension-order routing algorithm under seven different traffic patterns: uniform, hot-spot, matrix transpose, bit-reversal, complement, bit-flip, and perfect shuffle.

5.4.1 Uniform Traffic

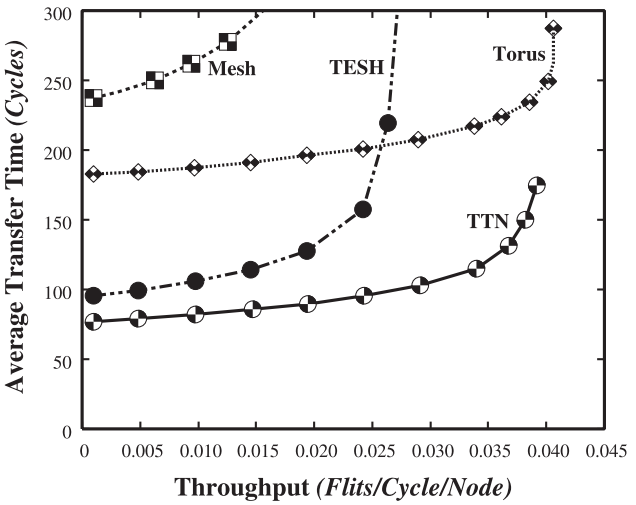
To evaluate the dynamic communication performance of the TESH and TTN, we have used minimum inter-level connectivity, i.e., $q = 0$. For a fair comparison, we allocated 4 virtual channels to the router for performance evaluation. Figure 7 (a) shows the results of simulation under uniform traffic patterns for the various network models. This figure presents the average transfer time as a function of network throughput. Each curve stands for a particular network. As shown in Fig. 7 (a), the average transfer time of the TTN is far lower than that of the mesh network, significantly lower than that of TESH network, and slightly higher than that of the MH3DT network. However, this benefit of latency for MH3DT network is achieved at the cost of extra links (Table 1). The saturation point of each line in the figure represents the maximum throughput. The maximum throughput of the TTN is higher than that of the MH3DT, TESH, and mesh networks. Therefore, the dynamic communication performance of the TTN with minimum inter-level connectivity is better than that of mesh, TESH, and MH3DT networks.

To show the superiority of the TTN over torus network, we have increased the inter-level connectivity from $q = 0$ to $q = 1$. Figure 7 (b) shows the simulation results under uniform traffic patterns for the TTN, along with mesh, TESH, and torus networks. Each curve stands for a particular network. As shown in Fig. 7 (b), the average transfer time of the TTN is far lower than that of the mesh and torus networks, and significantly lower than that of TESH networks. The maximum throughput of the TTN is higher than that of the mesh and TESH network. However, the maximum throughput of the TTN is a slighly lower than that of torus, with the accompanying cost of huge latency. Therefore, TTN achieves better dynamic communication performance than the other conventional networks and the hierarchical network under the uniform traffic.

The mesh and torus networks are deadlock-free using



(a) $q = 0$



(b) $q = 1$

Fig. 7 Dynamic communication performance of dimension-order routing with uniform traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits: (a) $q = 0$ (b) $q = 1$.

1 and 2 virtual channels, respectively. In our experiment, the performance of mesh and torus networks using 1 and 2 virtual channels is far lower than the performance of TTN. For fair comparison we consider here the performance of all networks with 4 virtual channels and the virtual channels are arbitrated by a round robin algorithm.

As mentioned earlier, in uniform traffic source and destination are randomly selected. If the source and the destination are situated in the two halves of the network, then the message has to cross the middle of the network. Thus, the middle of the network is congested, and usually, the maximum throughput of a mesh network under the uniform traffic is the lowest. In the torus network, due to wrap-around links between end-to-end nodes, this congestion is lessened. Therefore, the maximum throughput of the torus network is increased. The TTN is a hierarchical interconnection network, consists of multiple BMs and the BM is a torus net-

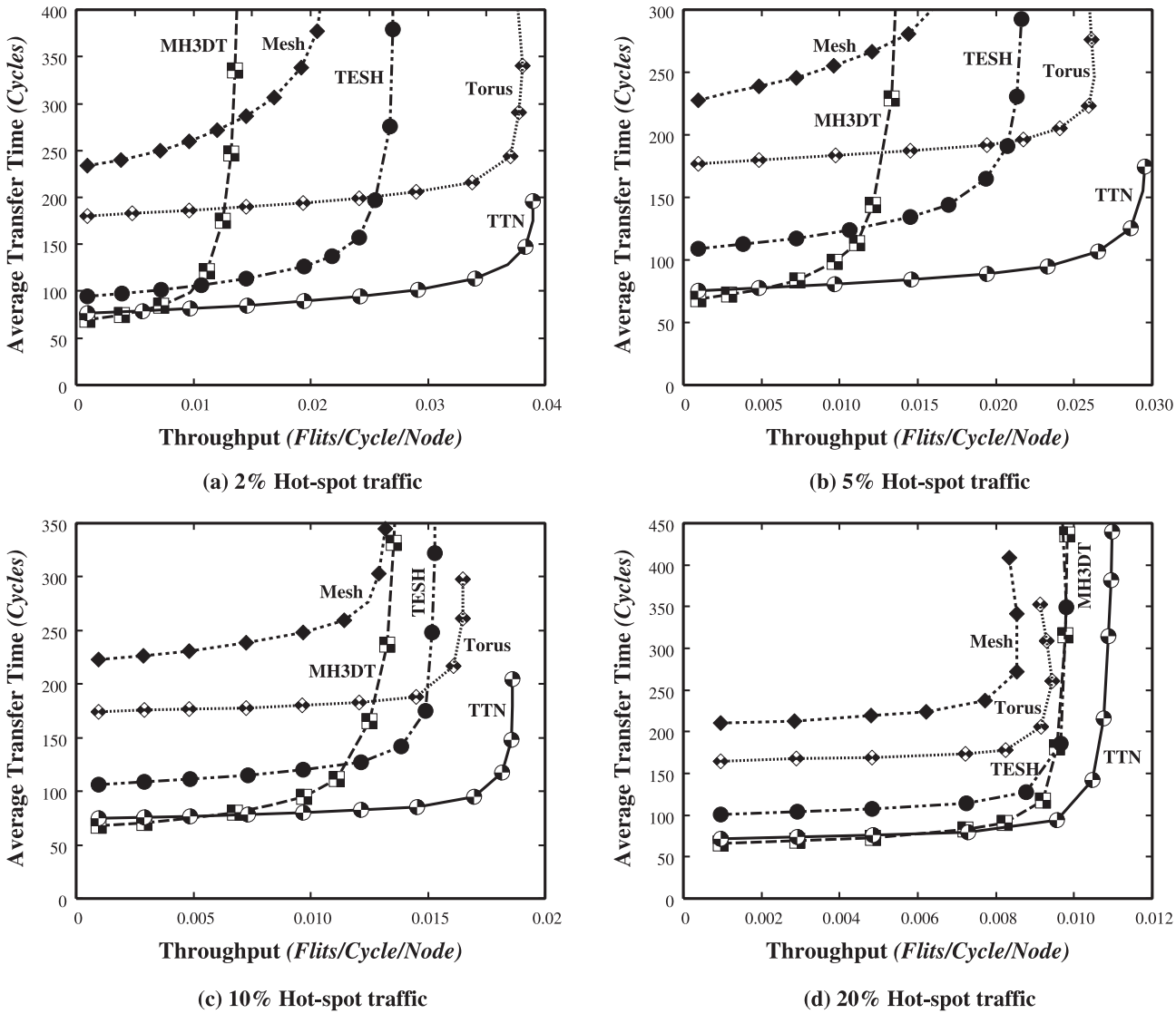


Fig. 8 Dynamic communication performance of dimension-order routing with hot spot traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

work. Due to randomness of uniform traffic pattern, if the source and the destination are situated in the one half of the network or even in the same subnet modules (BM or Level-2), then the middle of the network is not congested. The locality that exists in the uniform traffic pattern and hierarchical nature of the TTN results high saturation throughput than that of mesh network.

5.4.2 Hot-Spot Traffic

For generating hot spot traffic we used a model proposed by Pfister and Norton [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. Here, in uniform distribution, message destinations are chosen randomly with equal probability among the nodes in the network. However, in real application, it may happen that there are some packets (hot-spot packets) which remain in the network, and request rates are very high. Here, the simulations were carried out under the condition that, for TTN and TESH network, Node$(n_5, n_4)(n_3, n_2)(n_1, n_0)$ are source nodes and Node$(n_5, n_4)(0, 0)(0, 0)$ are hot-spot nodes. For mesh and torus networks, we divide the networks into 4×4 matrix. From each part, the node which is closest to the center is assumed to be the hot-spot node. The hot-spot flit generation probability are assumed to be $P_h = 0.02, 0.05, 0.10,$ and 0.20, i.e., the hot-spot percentages are assumed to be 2%, 5%, 10%, and 20% for all networks.

Figure 8 depicts the message latency versus network throughput curves for various hot- spot traffic pattern. Figure 8 (a) represents the result of simulations using 2% hot spot traffic. It is shown that the average transfer time of the TTN is far lower than that of the mesh and torus network, significantly lower than that of TESH networks, and a

slightly higher than that of the MH3DT network. The maximum throughput of the TTN is far higher than that of the mesh, TESH, and MH3DT networks, and noticeablly higher than that of torus network. Figure 8 (b) represents the result of simulations using 5% hot spot traffic. It is shown that the average transfer time of the TTN is far lower than that of the mesh, torus, and TESH networks, and a little bit higher than that of the MH3DT network. The maximum throughput of the TTN is higher than that of the mesh, TESH, and MH3DT networks, and a significantly higher than that of torus networks. Figure 8 (c) represents the result of simulations using 10% hot spot traffic. It is also shown that the average transfer time of the TTN is lower than that of the mesh, torus, and TESH networks, and a little bit higher than that of the MH3DT network. The maximum throughput of the TTN is higher than that of those networks. Figure 8 (d) represents the result of simulations using 20% hot spot traffic. As usual, it is shown that the average transfer time of the TTN is far lower than that of the mesh, torus, and TESH networks, and slightly higher than that of the MH3DT network. The maximum throughput of the TTN is higher than that of those networks. The maximum throughput of the MH3DT network is equal to that of the TESH network. In all hot-spot traffic, the average transfer time of TTN is slightly higher than that of MH3DT network. However, the difference is trivial. This benefit of latency for MH3DT network is achieved with the cost of extra links (Table 1). Therefore, with the hot spot traffic pattern, TTN yields better dynamic communication performance than conventional mesh and torus networks, and hierarchical MH3DT and TESH networks.

One interesting point to be noted here is that the relative difference in maximum throughput between torus and the hierarchical MH3DT and TESH network decreases with the increase of hot spot traffic, and it is shown in Fig. 8 (d) that with 20% hot spot traffic the maximum throughput of the MH3DT and TESH networks are higher than that of the torus network. It is also noted that the relative difference in maximum throughput between TTN and torus network increases with the increase of hot-spot traffic.

5.4.3 Matrix Transpose Traffic

In matrix-transpose traffic pattern, each node sends messages to a node with an address of the reversed dimension index, i.e., $Node(x, y)$ communicates with $Node(y, x)$. In the BM, $Node(x, y)$ sends messages to $Node(y, x)$. At higher levels, the x-coordinates and y-coordinates of the subnet modules at a certain level are transposed. For instance, in the Level-2 TTN, $BM(x, y)$ sends messages to $BM(y, x)$.

The dynamic communication performance of various networks under the matrix transpose traffic pattern is shown in Fig. 9. The figure shows the average transfer time as a function of network throughput for different networks. Each curve stands for a particular network. From Fig. 9, it is seen that the average transfer time of the TTN is far lower than that of the mesh and torus networks, significantly lower than

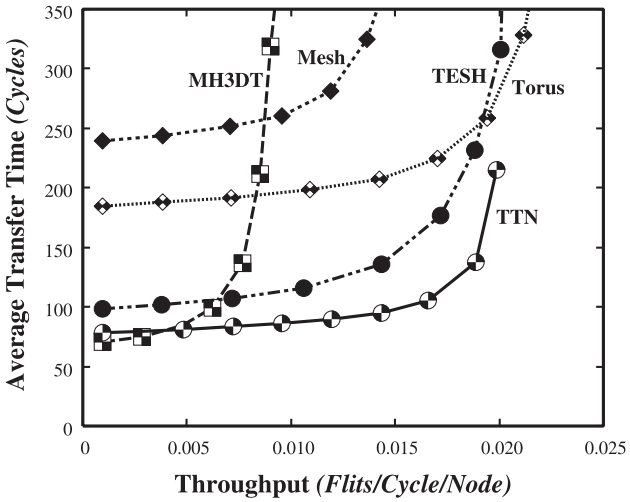


Fig. 9 Dynamic communication performance of dimension-order routing with matrix transpose traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

that of the TESH network, and slightly higher than that of MH3DT network. The maximum throughput of the TTN is higher than that of the mesh, MH3DT, and TESH networks, and slightly lower than that of the torus network. This benefit of throughput of the torus network is achieved with huge cost of message latency. With respect to average transfer time, TTN is better than torus network. However, with respect to throughput, torus network is better than TTN. Therefore, TTN achieves better dynamic communication performance than the hierarchical TESH and MH3DT networks, and the conventional mesh network.

5.4.4 Bit-Reversal Traffic

In a bit reversal traffic pattern, a node with address $Node\left(b_{\beta-1}, b_{\beta-2}\ ...\ ...\ b_1, b_0\right)$ sends messages to $Node\left(b_0, b_1, b_2\ ...\ ...\ b_{\beta-2}, b_{\beta-1}\right)$. Figure 10 depicts the results of simulations under bit reversal traffic pattern for the various network models. From Fig. 10, it is seen that the average transfer time of the TTN is far lower than that of the conventional mesh and torus networks, significantly lower than that of the TESH network, and slightly higher than that of MH3DT network. The maximum throughput of the TTN is far higher than that of mesh, MH3DT, and TESH networks, and slightly higher than that of torus network. Therefore, TTN achieves better dynamic communication performance than the other conventional and hierarchical networks under the bit reversal traffic pattern.

5.4.5 Complement Traffic

The binary representation of the node address is $b_{\beta-1}, b_{\beta-2}\ ...\ ...\ b_1, b_0$. In complement traffic, the $Node\left(b_{\beta-1}, b_{\beta-2}\ ...\ ...\ b_1, b_0\right)$ communicates with the $Node\left(\overline{b_{\beta-1}}, \overline{b_{\beta-2}},\ ...\ ...\ \overline{b_2}, \overline{b_1}, \overline{b_0}\right)$. Figure 11 portrays the results of simulations un-

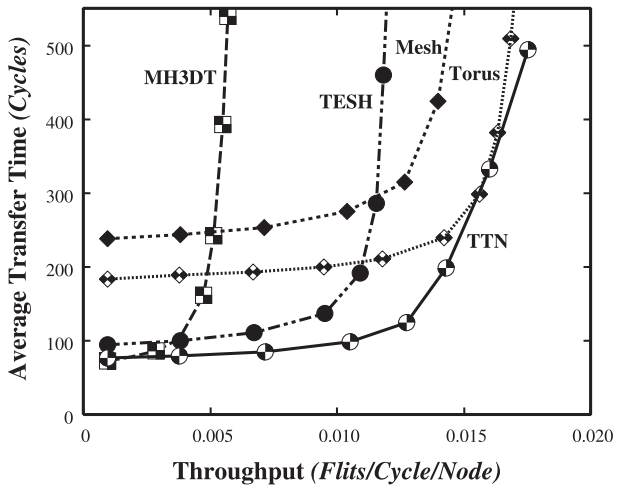


Fig. 10 Dynamic communication performance of dimension-order routing with bit-reversal traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

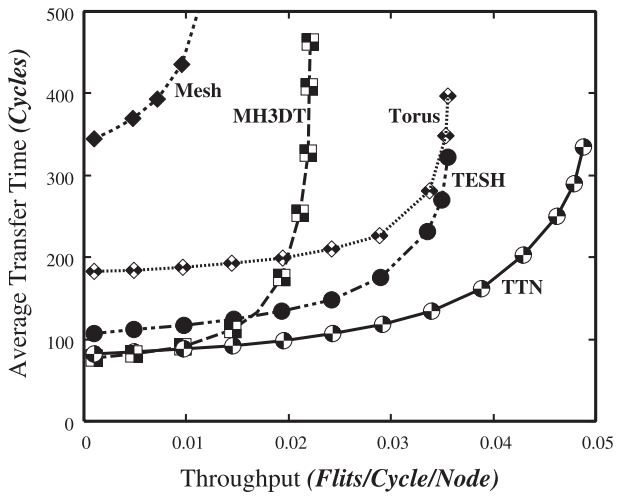


Fig. 11 Dynamic communication performance of dimension-order routing with complement traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

der complement traffic pattern for the various network models. From Fig. 11, it is seen that the average transfer time of the TTN is far lower than that of the mesh and torus networks, significantly lower than TESH network, and a little bit higher than that of MH3DT network (however, the difference is trivial). The maximum throughput of the TTN is far higher than that of conventional mesh and torus networks, and hierarchical MH3DT and TESH networks. Therefore, TTN achieves better dynamic communication performance than the other conventional and hierarchical networks under the complement traffic pattern.

In complement traffic pattern, all the messages cross the bisection of the network. Therefore, the middle of the network is congested, and usually, the maximum throughput of a mesh network under the complement traffic is the lowest, in comparison with other traffic patterns. In the

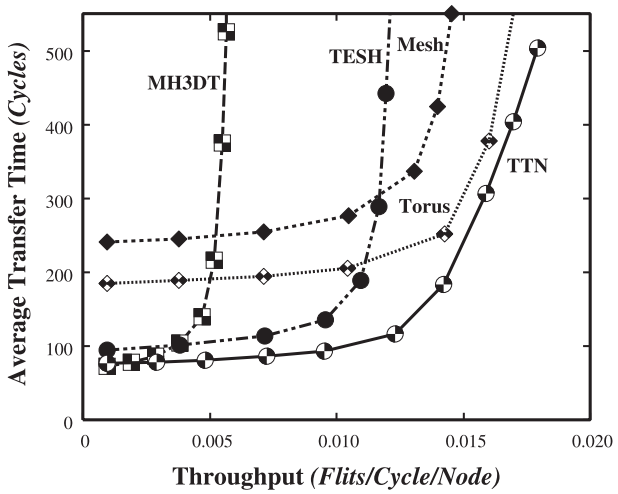


Fig. 12 Dynamic communication performance of dimension-order routing with bit-flip traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

torus network, due to wrap-around links between end-to-end nodes, this congestion is lessened. Thus, the maximum throughput of the torus network is increased. However, throughput is less than that for the uniform traffic pattern. Here, the most interesting point is that under complement traffic pattern, the maximum throughput of the hierarchical interconnection networks MH3DT, TESH, and TTN networks are higher than for the uniform traffic pattern. Among these networks, TTN yields the highest throughput. Therefore, TTN is a suitable network for complement traffic pattern.

5.4.6 Bit-Flip Traffic

In a bit-flip traffic pattern, a node with address $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0 communicates with the $Node\left(\overline{b_0}, \overline{b_1}, \ldots \ldots \overline{b_{\beta-2}}, \overline{b_{\beta-1}}\right)$. That is, complement of bit-reversal traffic. Figure 12 depicts the results of simulations under bit-flip traffic pattern for the various network models. From Fig. 12, it is seen that the average transfer time of the TTN is far lower than that of the conventional mesh and torus networks, slightly lower than that of the hierarchical TESH network, and slightly higher than that of MH3DT network. The maximum throughput of the TTN is far higher than that of mesh, MH3DT, and TESH networks, and significantly higher than that of the torus network. Therefore, TTN yields better dynamic communication performance than the mesh, torus, MH3DT, and TESH networks under the bit-flip traffic pattern.

5.4.7 Perfect Shuffle Traffic

In a perfect-shuffle traffic pattern, a node with address $b_{\beta-1}, b_{\beta-2}$ ... ... b_1, b_0 communicates with the $Node\left(b_{\beta-2}, b_{\beta-3}, \ldots \ldots b_1, b_0, b_{\beta-1}\right)$. That is, rotate left 1 bit. Figure 13 shows the average transfer time versus network

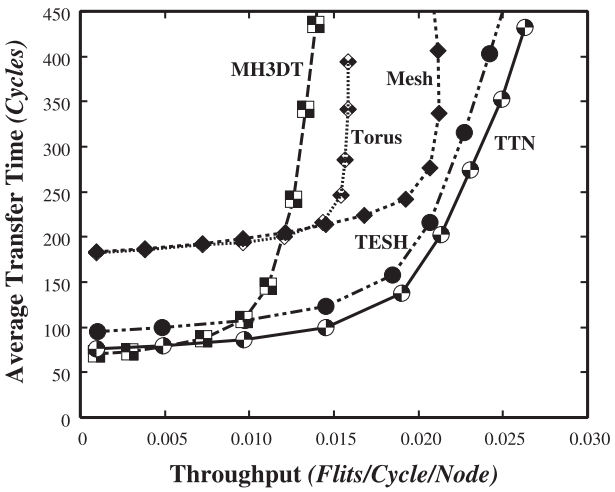


Fig. 13 Dynamic communication performance of dimension-order routing with perfect shuffle traffic pattern on various networks: 4096 nodes, 4 VCs, 16 flits, and $q = 1$.

throughput curve for perfect shuffle traffic. From Fig. 13, it is seen that the average transfer time of the TTN is far lower than that of the mesh and torus networks, noticeably lower than that of TESH network, and slightly higher than that of MH3DT network. The maximum throughput of the TTN is far higher than that of mesh, torus, and MH3DT networks, and significantly higher than that of TESH network. Therefore, TTN achieves better dynamic communication performance than the other conventional and hierarchical networks under the perfect shuffle traffic pattern.

5.5 Summarizing Dynamic Communication Performance

From the performance evaluation, there are some important points to be noted for the TTN. In all the traffic patterns, the average transfer time of the TTN at zero load is a little bit higher than that of MH3DT network. However, the difference is trivial. And the benefit of MH3DT network regarding average transfer time is achieved at the cost of extra physical links, as shown in Table 1. However, the maximum throughput of the TTN outperforms the conventional mesh and torus networks and hierarchical MH3DT and TESH networks. This low latency and high throughput of the TTN will make it a good choice for future generation massively parallel computer systems.

Torus is a suitable network for parallel computers, due to its symmetry and regularity. However, the length of the longest wire is a limiting factor for a network with thousands or millions of nodes. The operating speed of a network is limited by the physical length of links. With the cost of some additional short length links, the dynamic communication performance of the TTN is better than that of torus network under uniform and various non-uniform traffic patterns. An interesting point is that the maximum throughput of the TTN under complement traffic pattern is better than that of uniform traffic pattern.

6. Conclusion

A new hierarchical interconnection network, called **T**ori connected **T**orus **N**etwork (**TTN**), is proposed for the high performance massively parallel computer systems. The architecture of the TTN, addressing of nodes, routing of messages, and static network performance were discussed in detail. From the static network performance, it has been shown that the TTN possesses several attractive features, including constant node degree, small diameter, low cost, high connectivity, small average distance, and moderate (neither too low, nor too high) bisection width.

A deadlock-free routing algorithm using dimension order routing with 4 virtual channels has been proposed for the TTN. By using the routing algorithm described in this paper, and using uniform and various non-uniform traffic patterns, we have evaluated the dynamic communication performance of the TTN, as well as that of several other interconnection networks. The average transfer time of the TTN is lower than that of the conventional mesh and torus networks and hierarchical TESH networks, and it is slightly higher than hierarchical MH3DT network. Maximum throughput of the TTN is also higher than that of conventional mesh and torus networks, and hierarchical MH3DT and TESH networks. A comparison of dynamic communication performance reveals that the TTN achieves better performance than the mesh, torus, TESH, and MH3DT networks. The TTN yields low latency and high throughput with reasonable cost, which are indispensable for high-performance massively parallel computers. Therefore, TTN would be a good choice of interconnection network for next generation massively parallel computers.

This paper focused on the architectural structure, deadlock-free routing, static network performance, and the dynamic communication performance using dimension order routing of the TTN. Issues for future work include the following: (1) assessment of the performance improvement of the TTN with an adaptive routing algorithm, (2) evaluation of the system yield by providing hardware redundancy, i.e., defect-tolerance performance and (3) investigation into embedding other frequently used topologies in the TTN.

Acknowledgements

The authors are grateful to the anonymous expert reviewers for their extremely valuable comments, which helped to greatly improve the clarity of this paper. This paper has benefited greatly from editing by Mary Ann Mooradian, Lecturer of the JAIST, Technical Communication Program.

References

[1] W.J. Dally, "Performance analysis of k-ary n-cube interconnection networks," IEEE Trans. Comput., vol.39, no.6, pp.775–785, June 1990.

[2] Y.R. Potlapalli, "Trends in interconnection network topologies: Hierarchical networks," Int'l. Conf. on Parallel Processing Workshop,

pp.24–29, 1995.
[3] A. El-Amawy and S. Latifi, "Properties and performance of folded hypercube," IEEE Trans. Parallel Distrib. Syst., vol.2, no.1, pp.31–42, 1991.
[4] A. Esfahanian, L.M. Ni, and B.E. Sagan, "The twisted n-cube with application to multiprocessing," IEEE Trans. Comput., vol.40, no.1, pp.88–93, 1991.
[5] J.M. Kumar and L.M. Patnaik, "Extended hypercube: A hierarchical interconnection network of hypercube," IEEE Trans. Parallel Distrib. Syst., vol.3, no.1, pp.45–57, 1992.
[6] N.F. Tzeng and S. Wei, "Enhanced hypercube," IEEE Trans. Comput., vol.40, no.3, pp.284–294, 1991.
[7] S.G. Ziavras, "A versatile family of reduced hypercube interconnection network," IEEE Trans. Parallel Distrib. Syst., vol.5, no.11, pp.1210–1220, 1994.
[8] 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.
[9] M.M. Hafizur Rahman, Y. Miura, and S. Horiguchi, "Dynamic communication performance of hierarchical interconnection network: H3D-mesh," Proc. 2nd ICECE, pp.352–355, Bangladesh, Dec. 2002.
[10] S. Horiguchi and T. Ooki, "Hierarchical 3D-torus interconnection network," Proc. ISPAN'00, pp.50–56, Texas, USA, 2000.
[11] S. Horiguchi and T. Ooki, "Hierarchical 3D-torus interconnection network for massively parallel computers," JAIST Research Report, IS-RR-2000-022, pp.1–15, ISSN 0918-7553, 2000.
[12] V.K. Jain, T. Ghirmai, and S. Horiguchi, "TESH: A new hierarchical interconnection network for massively parallel computing," IEICE Trans. Inf. & Syst., vol.E80-D, no.9, pp.837–846, Sept. 1997.
[13] V.K. Jain and S. Horiguchi, "VLSI considerations for TESH: A new hierarchical interconnection network for 3-D integration," IEEE Trans Very Large Scale Integr. (VLSI) Syst., vol.6, no.3, pp.346–353, 1998.
[14] Y. Miura, Wormhole Routing for Hierarchical Interconnection Networks, Ph.D. Dissertation, School of Information Science, JAIST, 2002.
[15] M.M. Hafizur Rahman, Y. Inoguchi, and S. Horiguchi, "Modified hierarchical torus network," IEICE Trans. Inf. & Syst., vol.E88-D, no.2, pp.177–186, Feb. 2005.
[16] F.P. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," J. ACM, vol.24, no.5, pp.300–309, May 1981.
[17] L.M. Ni and P.K. McKinley, "A survey of wormhole routing techniques in direct networks," Computer, vol.26, no.2, pp.62–76, 1993.
[18] W.J. Dally and C.L. Seitz, "The torus routing chip," Journal of Distributed Computing, vol.1, no.3, pp.187–196, 1986.
[19] 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, 1987.
[20] W.J. Dally, "Virtual-channel flow control," IEEE Trans. Parallel Distrib. Syst., vol.3, no.2, pp.194–205, 1992.
[21] L. Schwiebert and D.N. Jayasimha, "A necessary and sufficient condition for deadlock-free wormhole routing," J. Parallel Distrib. Comput., vol.32, no.1, pp.103–117, 1996.
[22] Y. Miura and S. Horiguchi, "A deadlock-free routing for hierarchical interconnection network: TESH," Proc. 4th Int'l. Conf. HPC Asia, pp.128–133, Beijing, China, 2000.
[23] L. Schwiebert, "A performance evaluation of fully adaptive wormhole routing including selection function choice," IEEE Int'l. Performance, Computing, and Communications Conference, pp.117–123, 2000.
[24] G.F. Pfister and V.A. Norton, "Hot spot contention and combining in multistage interconnection networks," IEEE Trans. Comput., vol.C-34, no.10, pp.943–948, 1985.
[25] A.A. Chien and J.H. Kim, "Planer-adaptive routing: Low-cost adaptive networks for multiprocessors," J. ACM, vol.42, no.1, pp.91–123, 1995.
[26] J.H. Kim, Planer-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors, M.Sc. Thesis, University of Illinois at Urbana-Champaign, 1993.
[27] F. Petrini and M. Vanneschi, "k-ary n-trees: High performance networks for massively parallel architectures," Technical Report TR-95-18, Universita di Pisa, Dec. 1995.
[28] K. Bolding, M. Fulgham, and L. Synder, "The case of chaotic adaptive routing," IEEE Trans. Comput., vol.46, no.12, pp.1281–1292, 1997.
[29] H.H. Najaf-abadi and H. Sarbazi Azad, "The effects of adaptivity on the performance of the OTIS-hypercube under different traffic patterns," Proc. IFIP Int'l. Conf. NPC2004, LNCS, pp.390–398, 2004.
[30] M. Grammatikakis, D.F. Hsu, M. Kratzel, and J.F. Sibeyn, "Packet routing in fixed connection networks: A survey," J. Parallel Distrib. Comput., vol.54, no.2, pp.77–132, 1998.
[31] P.R. Miller, Efficient Communications for Fine-Grain Distributed Computers, Ph.D. Dissertation, Southampton University, U.K., 1991.
[32] J. Xu, Topological Structure and Analysis of Interconnection Networks, Kluwer Academic Publishers, Dordrecht, Netherlands, 2001.

M.M. Hafizur Rahman received his B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Khulna (erstwhile BIT, 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. He is currently serving as an assistant professor in the Dept. of CSE at KUET. He was also a visiting researcher in the School of Information Science at JAIST in 2008. His current research include interconnection networks, especially hierarchical interconnection networks and optical switching networks. Dr. Rahman is member of IEB of Bangladesh.

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.

Yukinori Sato received the BS degree, and the MS and Ph.D. degree in Information Sciences 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 JAIST (Japan Advanced Institute of Science and Technology) 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, and IPSJ.

Susumu Horiguchi received his M.E and D.E degrees from Tohoku University in 1978 and 1981, respectively. He is currently a full professor in the Graduate School of Information Science, Tohoku University. He was a visiting scientist at the IBM Thomas J. Watson Research Center from 1986 to 1987 and a visiting professor at The Center for Advanced Studies, the University of Southwestern Louisiana and at the Department of Computer Science, Texas A&M University in the summers of 1994 and 1997. He was also a professor in the Graduate School of Information Science, JAIST (Japan Advanced Institute of Science and Technology). He has been involved in organizing many international workshops, symposia and conferences sponsored by the IEEE, IEICE and IPS. His research interests have been mainly concerned with interconnection networks, parallel computing algorithms, massively parallel processing, parallel computer architectures, VLSI/WSI architectures, and Multi-Media Integral Systems. Prof. Horiguchi is a senior member of the IEEE Computer Society, and a member of the IPS and IASTED.