A Switch-Tagged Routing Methodology for PC Clusters with VLAN Ethernet

Michihiro Koibuchi, *Member*, *IEEE*, Tomohiro Otsuka,
Tomohiro Kudoh, *Member*, *IEEE Computer Society*, and
Hideharu Amano, *Member*, *IEEE Computer Society*

Abstract—Ethernet has been used for connecting hosts in PC clusters, besides its use in local area networks. Although a layer-2 Ethernet topology is limited to a tree structure because of the need to avoid broadcast storms and deadlocks of frames, various deadlock-free routing algorithms on topologies that include loops suitable for parallel processing can be employed by the application of IEEE 802.1Q VLAN technology. However, the MPI communication libraries used in current PC clusters do not always support tagged VLAN technology; therefore, at present, the design of VLAN-based Ethernet cannot be applied to such PC clusters. In this study, we propose a switch-tagged routing methodology in order to implement various deadlock-free routing algorithms on such PC clusters by using at most the same number of VLANs as the degree of a switch. Since the MPI communication libraries do not need to perform VLAN operations, the proposed methodology has advantages in both simple host configuration and high portability. In addition, when it is used with on/off and multispeed link regulation, the power consumption of Ethernet switches can be reduced. Evaluation results using NAS parallel benchmarks showed that the performance of the topologies that include loops using the proposed methodology was comparable to that of an ideal one-switch (full crossbar) network, and the torus topology in particular had up to a 27 percent performance improvement compared with a tree topology with link aggregation.

Index Terms—Ethernet, routing, deadlock avoidance, interconnection networks, PC clusters.

1 Introduction

Ethernet has been used for interconnection networks of various PC clusters because of its high-performance per cost. Unlike the early Beowulf clusters, recent PC clusters with Ethernet employ system software [1] that supports low-latency zero- or one-copy communication used in system area networks (SANs) [2], [3], [4]. High-throughput (non-blocking) commercial Ethernet switches are now available, and the link bandwidth of Ethernet has rapidly increased, as evidenced by the standardizations of 10-gigabit Ethernet (10 GbE). As of November 2008, GbEs were employed as interconnects on 56 percent of the TOP500 supercomputers [5]. When developing a PC cluster using Ethernet, there are two ways of constructing an intracluster Ethernet: one is to use a switch with several hundreds or more ports, and the other is to connect a number of switches, each having dozens of ports. Since large-scale switches with many ports are expensive, the latter way is preferable to make the best use of cost effectiveness of Ethernet.

Unlike clusters employing SANs, most current PC clusters using Ethernet have employed simple tree-based topologies. This is mainly because topologies that include loops are not allowed in order to avoid broadcast storms which circulate packets forever in layer-2 Ethernet.

There are studies on using IEEE 802.1Q tagged virtual LAN (VLAN) technology for setting up multiple paths between a pair of switches on topologies that include loops, such as mesh [6], [7]. The existing VLAN-based routing method, however, cannot be easily applied to most of current PC clusters with Ethernet, because the message passing interface (MPI) communication libraries used in such PC clusters usually do not support tagged VLAN technology. Consequently, not much work exists on evaluating deadlock-free routing algorithms in Ethernet, or their impact on real PC clusters with many Ethernet switches.

The required number of VLANs has been shown to be proportional to the number of switches in typical topologies such as mesh [8]. Although the IEEE 802.1Q VLAN tag field can identify $4{,}094$ $(2^{12}-2)$ VLANs, commercial cost-effective Ethernet switches support only a limited number of VLANs; this seems to be a limiting factor on implementation and extension of PC clusters.

In this study, we propose a switch-tagged routing methodology for PC clusters whose MPI communication libraries do not need to use tagged VLAN technology. Various deadlock-free routing algorithms can be employed by using at most the same number of VLANs as the degree of a switch; hence, the above limiting factor to building cost-effective PC clusters with many switches can be relaxed.

The proposed methodology has both high portability and a simple host configuration that does not modify the

- *M. Koibuchi is with the Information Systems Architecture Research Division, National Institute of Informatics (NII)/JST, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. E-mail: koibuchi@nii.ac.jp.*
- *T. Otsuka is with the Information Technology Center, Keio University, 3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan. E-mail: terry@am.ics.keio.ac.jp.*
- *T. Kudoh is with the Information Technology Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan. E-mail: t.kudoh@aist.go.jp.*
- *H. Amano is with the Department of Information and Computer Science, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan. E-mail: hunga@am.ics.keio.ac.jp.*

Manuscript received 14 Jan. 2009; revised 17 June 2009; accepted 8 Sept. 2009; published online 2 Apr. 2010.
Recommended for acceptance by M. Ould-Khaoua.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-01-0020.
Digital Object Identifier no. 10.1109/TPDS.2010.73.

MPI communication libraries. It simply configures a switch as follows: it disables the spanning-tree protocol (STP), allocates the VLAN sets, and optionally registers static MAC addresses of hosts for the routing. The proposed methodology can thus be applied to various current PC clusters with Ethernet.

In addition, the proposed methodology together with on/off and multispeed link regulation can reduce the power consumption of Ethernet switches with almost no performance degradation. To save power, it deactivates or slows down links that send and receive fewer frames. The available network resources (switch and links) vary depending on whether the port is selected to be shut down or not. The proposed methodology immediately stabilizes the MAC-address tables of the updated paths on various topologies.

The rest of this paper is organized as follows: In Section 2, we introduce related work. In Section 3, we propose and illustrate the switch-tagged routing methodology. In Sections 4 and 5, we evaluate the proposed methodology and discuss its performance factors. Our conclusions are in Section 6.

2 Related Work

Ethernet has employed a tree topology in order to avoid broadcast storms that circulate broadcast frames or unknown-destination frames forever, and deadlocks of frames that occur when the IEEE 802.3x link-level flow control is enabled [9].

High-performance deterministic routing algorithms that break cyclic channel dependencies have been studied for lossless interconnection networks that can include Ethernet with the IEEE 802.3x flow control [9], [10], [11], [12], and some of them can be implemented on Ethernet by statically registering MAC addresses of hosts without VLAN technology. However, it is difficult to stabilize the management of frames with such a configured Ethernet when a broadcast storm occurs.

Transparent bridges were proposed as a way of allowing loops [13], and their protocol improves network performance. A bridge protocol attempts to find alternate paths called spanning tree alternate routing (STAR)[14], although our methodology can be applied to existing cheap Ethernet switches.

Routing implementation techniques using VLAN technology for topologies that include loops have been developed [6], [7]. The VLAN technology was not intended for increasing network throughput, but for partitioning hosts into multiple groups, and it has been used in intranets and in the Internet backbone for the QoS control [15]. Multiple paths between hosts can be obtained by using VLANs as follows: multiple VLANs, each having a different tree of the physical network, are assigned to a physical network with loops. Each host is configured as a member of each VLAN, i.e., it has a virtual network interface to a VLAN. In this way, all pairs of hosts can communicate with each other via any VLAN tree topology, and there are multiple paths that consist of different link sets between each pair of hosts.

Since each path is assigned to a single VLAN, each source host selects a path by specifying a virtual interface that corresponds to the appropriate VLAN. Each tagged frame is transferred by the usual layer-2 Ethernet mechanism within its VLAN topology. Although each VLAN topology is logically a tree, the physical topologies of layer-2 Ethernet are free from tree structures.

We proposed VLAN topology sets and path assignment methods in a k-ary n-cube mesh and torus for estimating the required number of VLANs for the above VLAN-based routing implementation [8]. It requires k^{n-1} and $\frac{k^{n-1}}{2}+1$ VLANs on a k-ary n-cube mesh to provide balanced minimal paths and partially balanced ones, respectively. Similarly, $2k^{n-1}$ and $k^{n-1}+2$ VLANs are needed on a k-ary n-cube torus. A path optimization to the target application was analyzed and evaluated in VLAN Ethernet [16]. The existing VLAN-based routing implementation often requires a complicated VLAN configuration at each host, involving setting virtual network interfaces in correspondence with the VLAN ID, or managing multiple IP addresses on the virtual interfaces at a single host when using TCP/IP. Most Ethernet switches support IEEE 802.1D STP or 802.1D-2004 Rapid STP (RSTP) to prevent loops in a network. STP and RSTP are not aware of VLANs. When these protocols are enabled, all links out of a spanning tree are automatically disabled. Therefore, STP and RSTP must be disabled when a topology that includes loops is used. The 802.1Q-2003 Multiple STP (MSTP) and Cisco Systems' Per VLAN Spanning Tree (PVST) are STPs which support VLANs. They are quite useful for the VLAN-based routing implementation; however, there are currently only a few cost-effective Ethernet switches that support these protocols.

3 Switch-Tagged Routing Methodology

Here, we present our proposed switch-tagged routing methodology in order to implement various deadlock-free routing algorithms on topologies that include loops. There are two switch-tagged strategies (fixed and renamed VLAN assignments) that enable different path sets to be implemented using a small number of VLANs.

First, we explain the VLAN tagging operation at a switch. Second, we describe the switch-tagged routing methodology. Third, we focus on the methodology's properties.

3.1 Frame Tagging at Switch

A switch behavior of the VLAN tagging operation is as follows: when an untagged frame enters a port, it is tagged with a default VLAN ID tag number (port VLAN ID, PVID). Frames leaving the switch are either tagged or untagged depending on the port's VLAN configuration. If the port is a "tagged" member of a VLAN, the output frame is tagged with the respective VLAN ID. If the port is an "untagged" member of a VLAN, the output frame is left untagged. The VLAN untagged operation is originally intended to connect older equipment that does not support tagged VLAN.

3.2 VLAN Assignment

There are three functions for representing routing algorithms [17]. The simplest routing relation is based on the $N(source) \times N(destination) \mapsto P$ routing relation (all-at-once) [17], where N is the node set and P the path set. The other routing functions are the $N \times N \mapsto C$ routing relation, which only takes into account the current and

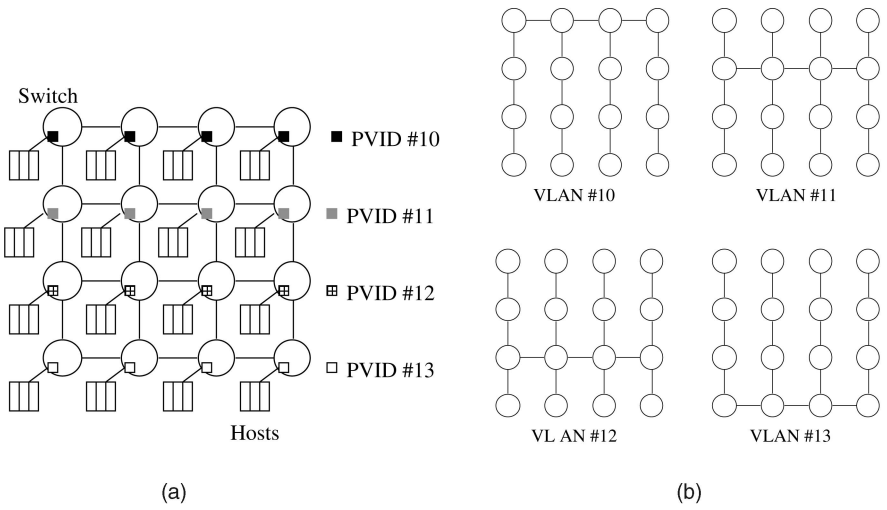


Fig. 1. The DOR fixed VLAN assignment in 4×4 2D mesh. (a) PVID. (b) VLAN set.

destination nodes [17], and the $C \times N \mapsto C$ routing relation, where C is the channel set.

3.2.1 Fixed VLAN Assignment

A path set expressed by the $N \times N \mapsto P$ routing relation where all paths from a host are contained by a single tree can be implemented in fixed VLAN assignment for an n-switch network as follows:

- Step 1: Let t_i be the tree topology that contains all paths from switch i. Let v_i be the VLAN that corresponds to t_i, and it is initialized to null. Let i be $zero$.
- Step 2: If an existing VLAN v includes t_i, let $v_i := v$; otherwise, create a new VLAN that includes t_i and let v_i be the new VLAN.
- Step 3: Set the PVID of the ports of switch i, that connect to hosts, to v_i.
- Step 4: Add each port for connecting a switch in t_i to v_i, and make this port "tagged" members of v_i.
- Step 5: If $i < n - 1$, let $i := i + 1$ and go to step 2.
- Step 6: Register each port connected to a host in all VLANs as an "untagged" member.

The complexity of fixed VLAN algorithm is $O(vn^2)$, where v is the number of VLANs.

Fixed VLAN assignment forwards a frame according to the above procedure as follows: a source host transmits a normal (untagged) frame in the usual way by specifying the IP address or MAC address of a destination host. When an untagged frame from a host enters a port of a switch, it is tagged with the PVID of the port and is regarded as a frame which belongs to the VLAN. Finally, the frame is untagged when it leaves a port connected to the destination host, because such a port is an "untagged" member of the VLAN. The destination host thus receives the usual untagged frame.

Fig. 1 shows the example of fixed VLAN assignment for the dimension-order routing (DOR) shown in [8]. The DOR is popular and goes by several names, e.g., XY (for 2D mesh) or e-cube (for hypercube). It routes frames by crossing dimensions in strictly increasing order, reducing to zero the offset in one dimension before routing in the next one [18]. The figure uses four VLANs in order to express the paths of the DOR.

3.2.2 Renamed VLAN Assignment

Renamed VLAN assignment can implement a path set expressed by the $C \times N \mapsto C$ routing relation. Only untagged frames arrive at input ports in a switch, and they are transferred using their PVID. Each switch that has p ports is configured according to the following procedure:

- Step 1: Let the PVID of port i be VLAN v_i, and register the port in v_i as an "untagged" member. Let i be $zero$.
- Step 2: Register each output port to which frames from the input port i can be routed in v_i as an "untagged" member in order to implement paths.
- Step 3: If $i < p - 1$, let $i := i + 1$ and go to step 2.
- Step 4: Combine two or more VLANs whose member ports are the same into a single VLAN in order to remove the duplication.

Fig. 2a shows an example of renamed VLAN assignment on a fat tree that uses only two VLANs. Hosts 1, 2, 5, 6, 9, 10, 13, and 14 send frames to hosts on different switches via switch $s5$, while the other hosts send them via switch $s6$. Fig. 2b is an example of the above procedure in the fat tree. In step 2, in the case of switch $s1$ in Fig. 2b, since paths from hosts 1 and 2 go through the port to $s5$, the port to $s5$ belongs to VLANs $v2$ and $v3$ in addition to belonging to $v0$.

The complexity of renamed VLAN algorithm is $O(Dn^2)$, where n is the number of switches and D the network diameter.

Table 1 compares fixed and renamed VLAN assignments. The number of used VLANs is taken from Theorem 2 (see the next section).

3.3 MAC Address Management

A problem caused by the proposed methodology is the MAC-address management at switches. Ethernet switches usually learn unknown MAC addresses when they receive frames. When a path from host A to B and one from B to A use different VLANs, the intermediate switches of both paths cannot learn the destination MAC address. This is

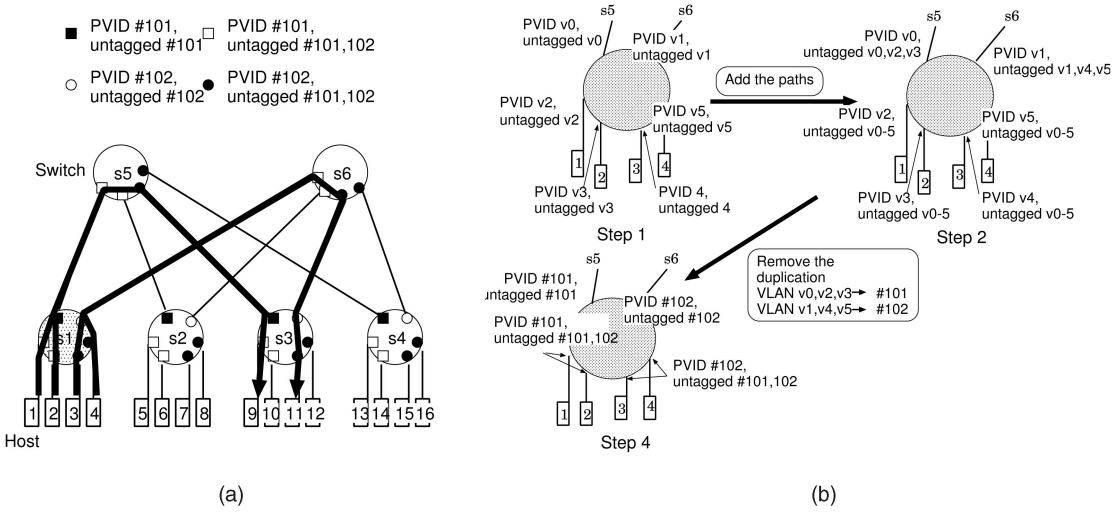


Fig. 2. Renamed VLAN assignment of fat tree. (a) Paths in fat tree. (b) Renamed VLAN assignment of switch s1.

because the MAC address self-learning procedure is independently performed on each VLAN.

This problem can be resolved through static MAC address registration. In recent commercial Ethernet switches, such as Dell PowerConnect 5324, operators can statically register pairs of MAC addresses, VLAN IDs, and output port numbers.

However, static registration cannot use the convenient switch function of address self-learning. To make the best use of the MAC-address self-learning mechanism, we propose the following learning procedure in the case of fixed VLAN assignment; it mitigates the effort required for switch management.

- Step 1: For each VLAN, make a corresponding virtual interface on each host. In the example of the mesh shown in Fig. 1, each host has virtual interfaces for VLANs #10, 11, 12, and 13. In Linux operating systems, virtual interfaces can be made by using the "vconfig" command.
- Step 2: Give an IP address to each virtual interface at all hosts so that the interface has a unique network address that belongs to a different segment on the physical interface.
- Step 3: At each host, broadcast an ICMP or UDP message from each virtual interface so that switches learn the MAC address of the host in each VLAN.

In step 2, the IP address is only used for the MAC address registration at each switch.

The easiest implementation of step 3 is to use the "ping (ICMP echo request)" command. The learning procedure should be retaken before the switch aging time expires. Fortunately, certain commodity Ethernet switches can set the aging time to infinity.

TABLE 1
Comparison of Fixed and Renamed VLAN Assignments in n-Switch Networks (Each Switch has p Ports)

	Fixed assignment	Renamed assignment
Routing relation	part of $N \times N \mapsto P$	$C \times N \mapsto C$
Number of VLANs (worst case)	n	p

3.4 Breaking Cyclic Channel Dependencies

A VLAN topology is a tree, and broadcast or multicast is independently completed within each VLAN in the case of fixed VLAN assignment. Multicast thus causes no broadcast storms. However, in renamed VLAN assignment, each intermediate switch changes the VLAN ID of a frame. This introduces the possibility of broadcast storms among different VLANs when a switch receives frames whose destination MAC addresses are unknown, or when a broadcast occurs.

In addition, a combination of VLANs could cause deadlocks, because VLANs have to share network resources, such as channel buffers. Frames are simply discarded when the buffers of a switch, which does not use the IEEE 802.3x link-level flow control, become filled. An upper layer end-to-end protocol such as TCP usually retransmits the discarded frames; deadlocks do not occur, although the network throughput would decrease. However, when the link-level flow control is enabled for increasing throughput, Ethernet works as a lossless network in which few frames are discarded. In this case, deadlocks could occur in the proposed VLAN routing.

Deadlock-free routing algorithms that break cyclic channel dependencies are used to avoid broadcast storms, deadlocks, and performance degradation, and they have been studied for lossless networks [9], [10], [11], [12].

Theorem 1. *A renamed VLAN assignment with a deadlock-free routing algorithm does not cause broadcast storms.*

Proof. Deterministic deadlock-free routing algorithms break cyclic channel dependency, and the channel dependency is implemented by the combination of VLANs. Thus, when a broadcast occurs, frames never arrive at a port of a switch they have already visited. □

3.5 Number of VLANs

Although the IEEE 802.1Q VLAN tag field can identify $4{,}094$ $(2^{12}-2)$ VLANs, commercial cost-effective Ethernet switches support only a limited number of VLANs. Let us

calculate the maximum number of VLANs required for the proposed methodology.

Theorem 2. *Fixed and renamed VLAN assignments require at most n and p VLANs, respectively, where n is the number of switches and p the switch degree.*

Proof. In the case of fixed VLAN assignment, all paths from a host belong to a VLAN and all paths from hosts connected to a single switch use the same VLAN. Since all switches would have at least a host, fixed VLAN assignment requires at most n VLANs.

Renamed VLAN assignment locally uses the same number of VLANs as the number of PVIDs at a switch, and the number of PVIDs is at most p. Renamed VLAN assignment thus uses at most p VLANs. □

The required number of VLANs is indeed small if we can make the best use of the regularity of the topology (see Section 4.1).

3.6 On/Off and Multispeed Link Regulation for Saving Power

Links of off-chip interconnection networks consume a lot of power even if no data are transferred, and their power consumption is almost constant regardless of the traffic injection rates. For example, IBM InfiniBand 12X LPE TX consumes a nominal power of 0.26 W and a worst-case power of 0.3 W, while its RX takes up nominal a 0.17 W and a worst-case 0.2 W [19]. We measured the power consumption of GbE switches using System Artware, a watthour meter (SHW3A), and it can capture the power consumption at intervals of a second, and its minimum unit is 0.1 W. In our measurement environment, we cannot detect the variation in the power consumption of Ethernet switches depending on the utilization from a 0-Mbps transfer to a 957-Mbps UDP transfer. The power consumption can be regarded as constant values regardless of link utilization in the following section for simply estimating the power consumption of systems.

The power consumption of links can be reduced by using the port-shutdown operation available in most commercial Ethernet switches. Their operation was not originally intended to reduce power consumption; it is normally used to block the injection of unexpected frames from neighboring switches. In addition to port-shutdown, power consumption is also reduced when the link-speed operation reduces the link speed to 100 or 10 Mbps.

Here, we propose to use the port-shutdown and link-speed operation for reducing the power consumed by switches.

Table 2 lists their results, and the power consumption of each switch except ports is constant regardless of the number of activated links. We measured the power consumption of switches when links had been physically removed from a switch, and the values are the same as in Table 2. Thus, the port-shutdown operation completely reduces the power of the port, even if a physical link is connected to the port in switches we measured. Although additional power could be needed for the operation to deactivate or activate links, it did not capture the behavior the power increases just after links turn off or on in all switches we measured.

In the table, the "GbE/port" column refers to the power consumption of a single port whose speed is full (1 Gbps), while the "100 M/port" column refers to that when the speed is set to 100 Mbps. The "All except ports" is the power consumption of switches when all the ports are shutdown, and "Max (port ratio)" is the power consumption when all ports are activated with 1 Gbps. "PC5324," "PC6224," and "PC6248" stand for Dell PowerConnect 5324, 6224, and 6248 nonblocking switches, respectively, while "SF-420" is Planex Communications SF-0420G. In the case of the SF-0420G, the power of the port is reduced when the opposite port of the link is also shutdown. The PC6224 and PC6248 switches are layer-3 switches, and they provide a larger number of services in comparison with layer-2 switches. Thus, we consider that they consume more power than the PC5324 and SF-0420G switches do.

TABLE 2
Power Consumption of GbE Switches (W)

	GbE/ port	100M/ port	10M/ port	All except ports	Max (port ratio)
PC5324	1.2	0.9	0.5	15.0	42.9 (65%)
PC6224	2.0	0.9	0.8	42.5	91.1 (53%)
PC6248	2.1	1.3	1.1	56.8	155.2(63%)
SF-420	1.0	0.2	0.0	32.6	55.4 (41%)

To monitor and manage ports of switches, most of Ethernet switches support the standard management information base (MIB). The standard MIB gives IP, UDP, and TCP traffic information, including the number of input and output frames, and the total amount at each port. A host can obtain them via the simple network management protocol (SNMP).

In addition, there is a large amount of monitoring and sampling software available for Ethernet, such as IPTraf [20], and such software can be used for traffic prediction. Thus, we can use on/off and multispeed link regulation to adjust the number of activated links so that the traffic load of every channel is uniform.

Ethernet has a unique MAC address management feature for switches that suits a tree topology. The feature makes it difficult to implement on/off link regulation algorithms on various topologies. To make on/off and multispeed-link interconnection techniques to Ethernet, the proposed methodology stabilizes the path update as follows:

1. Link Status: On to Off: The following procedure for a path reconfiguration deactivates a target link.

Step 1: Use an existing routing algorithm to calculate the path set so that it avoids the target link.

Step 2: Implement the path set in step 1 using a procedure in Section 3.2.

Step 3: Deactivate the target link.

2. Link Status: Off to On: The following procedure reactivates a target deactivated link.

Step 1: Activate the target link.

Step 2: Use an existing routing algorithm to calculate the path set so that it uses the target link.

Step 3: Implement the path set in step 2 using a procedure in Section 3.2.

The delay of the link wake-up and negotiation tends to be several seconds (see Table 6). However, the path

modification overhead of updating the PVID of the port is almost *zero*. Thus, the above procedures minimize the communication interruption.

3. Changing the Link Speed: Besides using the on/off link regulation, slowing down the links further reduces the power consumption of switches.

   The following procedure changes the speed of the target link.

Step 1: Use an existing routing algorithm to calculate the path set so that it avoids the target link.

Step 2: Implement the path set in step 1 using a procedure in Section 3.2.

Step 3: Change the speed of the target link.

Step 4: Use an existing routing algorithm to calculate the path set so that it uses the target link.

Step 5: Implement the path set in step 4 using a procedure in Section 3.2.

Step 2 employs temporal paths during changing the link speed in order to minimize the communication interruption.

3.7 Limitations of Existing Commodity Switches

1. Applicable Commercial Switches: Commodity GbE switches cost from under 100 dollars to 10,000 dollars. The cheapest switches do not support VLAN technology, or few functions of VLANs, and hence, they cannot employ our methodology. The proposed methodology can be applied to commercial switches that support the operations of the IEEE 802.1Q standard described in Section 3.1.
2. Upper Limit on Number of Hosts: The number of hosts is limited to size of the MAC address table in Ethernet switches. Each entry of an MAC address table consists of the destination MAC address, VLAN ID, and port. The maximum number of hosts H is thus estimated as follows:

$$H = \frac{T}{V}, \tag{1}$$

where T is the number of table entries at a switch and V the number of used VLANs. T is usually around 10,000 ($8k$ or $12k$) in cost-effective commodity switches, such as Dell PowerConnect 5324. Since V depends on the VLAN assignment (fixed or renamed), the upper limit on the number of hosts is strongly affected by the VLAN assignment. The details about the required number of VLANs are shown in Section 4.1.

TABLE 3
Number of VLANs

	Fixed VLAN assignment	Renamed VLAN assignment
Fat tree (u,d,r)	u^r	u
Mesh (k-ary n-cube)	k^{n-1}	n
Mesh (k-ary n-cube, 1-link deactivation)	$(k+1)k^{n-2}$	n
Torus (k-ary n-cube)	$2k^{n-1}$	$3n$
Torus (k-ary n-cube, 1-link deactivation)	$2(k+1)k^{n-2}$	$3n$

4 Fundamental Evaluation

Here, we show the fundamental evaluation results using a PC-cluster testbed and discuss the performance factors of the switch-tagged routing methodology.

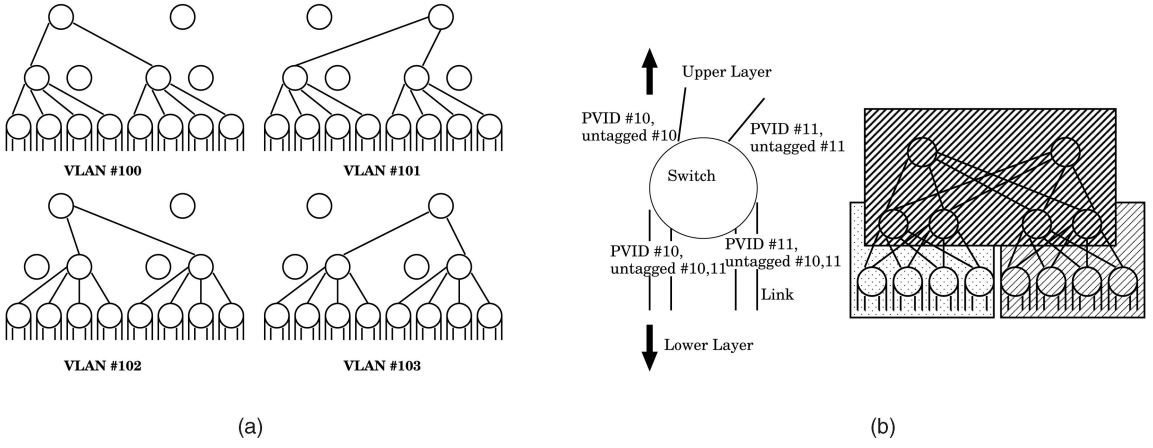
First, we show the number of VLANs in typical regular topologies and the overhead of VLAN operations at a switch. Second, we show the performance factors.

4.1 Number of VLANs

Table 3 shows the required number of VLANs in typical topologies. We assume that DOR is used, and two links between switches are used for breaking cycles in the torus.

As shown in Fig. 3a, fixed assignment requires u^r VLANs on a fat tree (u, d, r) whose switch has u upper links and d lower links, and the number of its layers is r. Fig. 3b shows that renamed assignment uses u VLANs.

In the case of fixed VLAN assignment on a mesh and torus, we use the assignments based on the method [8], [21]. As listed in Table 3, topological and routing regularity reduces the required number of VLANs especially in the



(a) (b)

Fig. 3. VLAN assignment in fat tree (2,4,2). (a) Fixed assignment. (b) Renamed assignment.

TABLE 4
Latency of Switch (in Microseconds)

	PC 5324			PC 6224			GSM7212			SF-0420G		
	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max
U-U	2.47	2.74	2.79	3.03	3.86	4.34	2.47	2.77	2.79	3.38	3.42	3.48
T-T	2.47	2.76	2.79	3.13	3.88	4.38	2.47	2.76	2.79	3.38	3.42	3.48
U-T	2.49	2.76	2.78	3.35	3.84	4.38	2.46	2.75	2.78	3.38	3.42	3.48
T-U	2.46	2.76	2.81	3.19	3.87	4.41	2.46	2.78	2.81	3.38	3.42	3.45
U-(T)-U	2.47	2.75	2.79	3.19	3.85	4.41	2.43	2.73	2.75	3.38	3.42	3.45

case of renamed VLAN assignment, compared with the worst case listed in Table 1.

Now let us consider the required number of VLANs under the condition that a single horizontal (x-directional) link is deactivated and the north-last turn model is employed to avoid it [22]. Table 3 shows that the required number of VLANs that use all activated links slightly increases when the link is deactivated.

4.2 Overhead of VLAN Operations

We compared the overhead of tagged VLAN operations in typical cost-effective switches. We measured the latencies of nonblocking layer-2 switches, Dell PowerConnect 5324, 6224, Netgear GSM7212, and Planex SF-0420G by using the ping command (ICMP message) between two hosts via GtrcNET-1 [23] which is a programmable network testbed that can also carefully monitor Ethernet traffic.

Table 4 lists the resulting switch latencies. "T-U" indicates the case in which an input tagged frame is transferred through its VLAN and its VLAN tag is removed at the output port of a switch, and "U-(T)-U" indicates the case in which an input untagged frame is transferred through the VLAN of the PVID, and the output frame is untagged at a switch. "U-(T)-U" is used at every switch in the case of renamed VLAN assignment. The results listed in the table show that the tagged VLAN operation does not affect the latency at these switches. The latency overhead of the PowerConnect 6224 is higher than that of the other switches, since it is a layer-3 switch that provides a larger number of services that would require additional internal operations in comparison with layer-2 switches, such as GSM7212.

We also measured the bandwidth using Tperf 1.5 software [24]. Table 5 shows the bandwidth of the nonblocking layer-2 switches. Since the VLAN tag (4 byte) is added to each frame, the ratio of raw data in a frame slightly decreases. The bandwidth of the tagged frame transfer (T-T and T-U) is thus slightly lower than that of untagged frame transfer in all switches. Renamed VLAN assignment, (U-(T)-U), has no bandwidth overhead, since it transfers untagged frames between switches.

Since the overheads of fixed and renamed VLAN assignments are almost the same, we evaluate only fixed VLAN assignment in the following sections.

TABLE 5
Bandwidth of Switch (in Megabits Per Second)

	PC5324	PC6224	GSM7212	SF-0420G
U-U (UDP)	957.0	957.1	957.1	956.9
T-T (UDP)	954.4	954.5	954.1	954.5
U-T (UDP)	956.9	956.9	957.0	956.9
T-U (UDP)	954.6	954.5	954.6	954.4
U-(T)-U(UDP)	957.0	957.4	957.1	957.1
U-U (TCP)	941.1	940.5	941.0	941.4
T-T (TCP)	936.9	938.1	938.0	937.9
U-T (TCP)	940.1	938.0	928.1	941.0
T-U (TCP)	937.7	938.4	938.1	938.1
U-(T)-U(TCP)	941.1	941.0	941.0	941.0

4.3 Overhead of On/Off and Multispeed Link Regulation

We measured the overhead of the on/off link and multispeed link operations at a switch. The proposed on/off link regulation can be decomposed into a port-shutdown operation and a VLAN operation at the switches in order to update the paths.

We measured the overhead of the port-shutdown and multispeed operations, while the ping command (ICMP message: 64 bytes) between two hosts is executed at intervals of 0.1 second. The switch continuously performed shutdown, and no-shutdown (resume) of the port, or changed link speed. The overhead of the on/off link or link-speed operation was taken to be while the link was inactive and the communication (ping frame) interrupted. "PC5324," "PC6224," and "PC6248" stand for Dell PowerConnect 5324, 6224, and 6248 nonblocking switches, respectively, while the "SF-420" means the Planex Communications SF-0420G.

Table 6 shows that the overhead of the on/off and multispeed link operation is usually several seconds. The overhead of Ethernet switches varies depending on the services provided by the commercial products. For example, some switches have unique functions for setting up the port, such as port mirroring for traffic monitoring, that would affect the overhead. In addition, the overhead is strongly affected by the switch options concerning auto-negotiation that is sometimes required by 1 Gbps link speed. We thus show the minimum overhead of the on/off and multispeed link operation in Table 6.

Fig. 4 shows the topology used for evaluating the overhead of VLAN operations. The PVID of the port to the sender host is updated at a switch, while the ICMP messages are transferred between two hosts at 0.1 second intervals.

The communication could be interrupted while the port status is being updated, and we take this to be the overhead of the VLAN operations. We monitor and measure the overhead using the GtrcNET-1 [23] for 10 seconds, and Table 7 lists its results.

The "fixed path" represents frame transfers using VLAN A in Fig. 4, while the "dynamic path" represents frame transfers using VLANs A and B that are dynamically changed at every a few seconds. Table 7 indicates that the overhead of the path update was too small to detect by our measurement equipment. In addition to the latency, we measured the frame bandwidth between two hosts by using the GtrcNET-1. Frame bandwidth is measured at every 100 $m$seconds, and it

TABLE 6
Overhead of On/Off and Link Speed Operation (in Seconds)

	On/Off	Gb → 100M	Gb ← 100M	100M → 10M	100M ← 10M	Gb → 10M	Gb ← 10M
PC5324	3.2	4.1	2.4	2.3	3.3	2.5	2.4
PC6224	2.6	2.8	5.7	2.6	2.6	2.6	5.7
PC6248	2.2	2.7	3.3	2.8	3.1	2.5	2.3
SF-420G	1.2	2.7	2.5	2.4	2.3	2.8	2.3

includes not only body but also control information of frames. The results show that the performance of latency and throughput was unaffected even though path set is switched at every a few seconds.

4.4 Behavior of Dynamic On/Off Link Regulation

Table 7 showed that the influence of the path update between a single source-and-destination pair is trivial. Here, we show that on multiple data transfers using a topology that includes loop. To analyze the behavior of dynamic on/off link regulation, we monitored the bandwidth at every 100 m seconds using GtrcNET-1 on the topology in Fig. 5, and the results are shown in Fig. 6. We used the testbed cluster for the evaluations, and Table 8 lists the specifications of each host. We used the PowerConnect 5324s for the switches.

In this monitor, first, we used topology B, and hosts 0, 1, 2, and 3 sent to 2, 3, 0, and 1 with a highest injection rate using Tperf [24], respectively. Host i is connected to switch i, where $0 \leq i \leq 3$, and we omit them in Fig. 5. After 5 seconds, host 3 sends data to 2 with a highest injection rate, and the topology is updated to topology A by reactivating the link. After 10 seconds, the data transfer from host 0 to 2 is terminated, and the topology is changed to topology B by deactivating the link.

Fig. 6 demonstrates that the frame bandwidth varied from 0 to 970 Mbps during the on/off link operation, and the communication is momentarily unstable.

The PVID of each switch is sequentially and manually updated in this simple measurement. If a tool to automatically update PVIDs in switches or if an efficient network reconfiguration is applied [25], the influence of the transition between old and new paths will be minimized.

There is a possibility to cause deadlocks between the path sets of topologies A and B during the update, if both paths make cyclic channel dependencies under the UDP data transfer with the link-level flow control [21]. To avoid deadlocks during the path update, dynamic reconfiguration techniques have been recommended for lossless interconnection networks [25], and they would be developed in Ethernet.

In this evaluation, to avoid deadlocks, we simply used a TCP transfer with the link-level flow control, and a UDP transfer with a single switch that disables the link-level flow control [21].

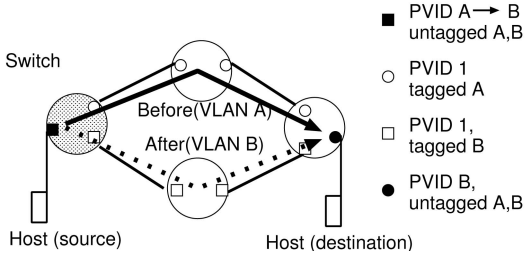


Fig. 4. Topology considered in evaluations.

4.5 Impact of Traffic Congestion and Path Hops

We evaluated the impact of traffic congestion and path hops on MPI-level bandwidth and latency, which are essential parameters for parallel processing on PC clusters.

We used the testbed cluster whose specification is listed in Table 8. Each host used SCore cluster system software [1] version 5.8.2. SCore is an open-source cluster system software package that provides various parallel programming environments, such as a low-level communication library PM and an MPI library MPICH-SCore based on MPICH-1.2.5. In addition, Tperf-1.5 [24] and Intel MPI benchmarks (IMBs) 2.3 [26] were used for measuring the performance of TCP/UDP transfer and MPI-level transfer between pairs of hosts, respectively.

First, we evaluated the effect of the IEEE 802.3x link-level flow control on bandwidth in the presence of path congestion. Fig. 7 represents the average bidirectional bandwidth between eight pairs of hosts using the IMB Multi-PingPing benchmark. "FC None" indicates that the link-level flow control is disabled, while "FC All" indicates that the flow control is enabled at every link. Each path between a host pair includes the same two switches and a link between the two switches; thus, all paths conflict on the interswitch link.

Since the figure illustrates that IEEE 802.3x link-level flow control is efficient for sharing bandwidth among conflicting paths, the link-level flow control should be used in the proposed methodology for increasing network performance.

Next, we focused on the path hops. Fig. 8 represents MPI-level one-way latency as measured by the IMB PingPong benchmark and bidirectional bandwidth as

TABLE 7
Performance of Path Modification (Testbed)

	Latency (μsec)			Frame bandwidth (Mbps)		
	Min	Ave	Max	Min	Ave	Max
Fixed path	8.21	8.24	8.28	967.9	971.8	973.2
Dynamic path	8.21	8.24	8.28	968.0	972.7	981.1

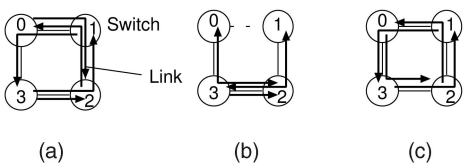


Fig. 5. Cyclic topologies considered in evaluations. (a) Topology A: all links are activated using two VLANs. (b) Topology B: a link is deactivated using a VLAN. (c) Cycles between paths on topologies A and B.

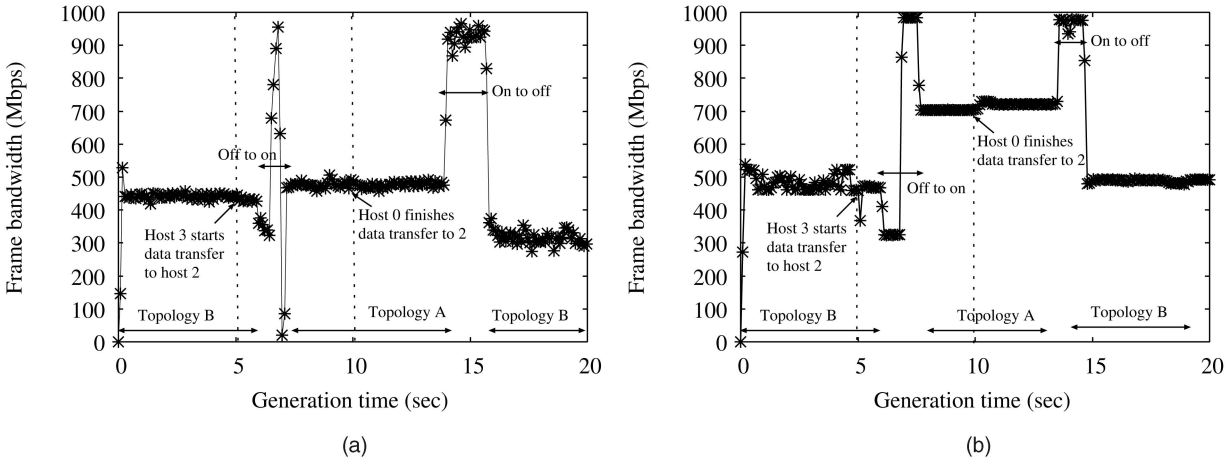


Fig. 6. Behavior of dynamic on/off link regulation. (a) TCP transfer. (b) UDP transfer.

measured by the IMB PingPing benchmark for varying numbers of intermediate switches between hosts, when the link-level flow control is enabled. The figure illustrates that the number of path hops is a crucial factor for the bandwidth as well as latency between hosts.

5 Evaluation Using PC Clusters

We show the evaluation results of various topologies by the proposed methodology.

5.1 PC Clusters

We used three types of PC clusters; the testbed described in the previous section, a 66-host PC cluster using six GbE switches (Dell PowerConnect 6248, 48 ports), and a 225-host PC cluster using the eight same GbE switches. The second cluster, called Misc, uses multicore processors that occur inter- and intraprocessor MPI communications, and its specifications are listed in Table 9. The third cluster, called SuperNova, only occurs the interprocessor communications among single-core processors. This cluster, which is at Doshisha University, Japan, was 93rd in the TOP500 ranking as of November 2003 [5], and its specifications are listed in Table 10. The testbed with 16 switches was used for performing various topologies, while the Misc and SuperNova clusters with six or eight switches were mainly used

TABLE 8
Specifications of Each Host (Testbed)

CPU	Intel Xeon 2.8GHz × 2 (SMP)
Memory	PC2-3200 DDR2 SDRAM 1Gbytes
Chipset	Intel E7520
PCI	64bit/133MHz PCI-X
NIC	Intel PRO/1000 MT server adapter
NIC Driver	Intel e1000 6.2.15
OS	Fedora Core 1 (kernel 2.4.21)

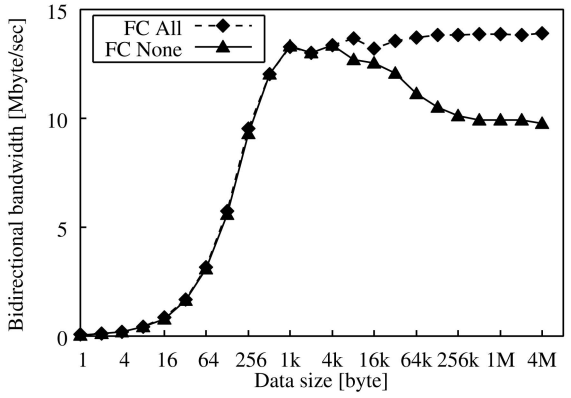


Fig. 7. Impact of traffic congestion.

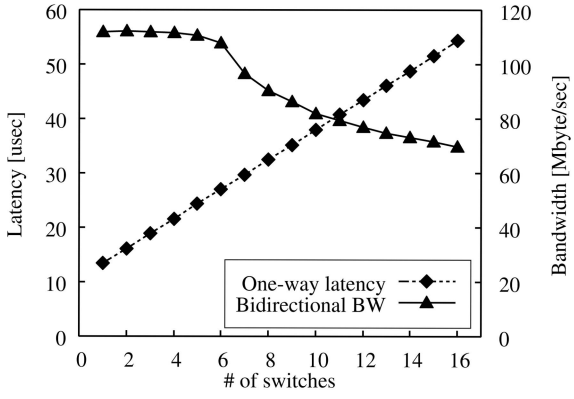


Fig. 8. MPI latency and bandwidth.

TABLE 9
Specifications of Each Host (Misc Cluster)

CPU	Quad-Core AMD Opteron 2.3GHz
Memory	DDR2 667 MHz 8GB
NIC & driver	Broadcom BCM95721, Tigon3
OS	CentOS 4.6
Kernel	2.6.9-67.0.15.ELsmp
MPICH	1.2.7p1

TABLE 10
Specifications of Each Host (SuperNova Cluster)

CPU	AMD Opteron 1.8 GHz × 2
Chipset	AMD 8131+8111
Memory	PC2700 Registered ECC 2 GB
OS	Debian GNU/Linux 4.0
Kernel	2.6.18-4-amd64
MPICH	1.2.7p1

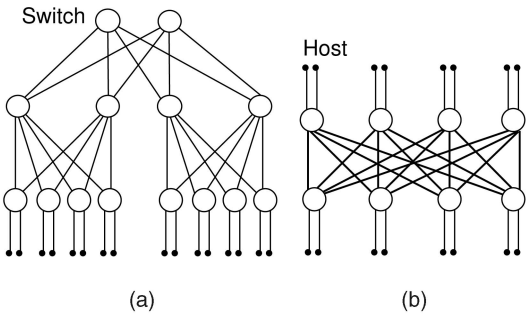


Fig. 9. Evaluated indirect topologies. (a) Fat tree (2,4,2). (b) Myri-clos (4×4).

for measuring the performance relationship between the proposed methodology and link aggregation. Each switch in the Misc cluster connects to 11 hosts, whereas each switch in the SuperNova cluster connects to 28 or 29 hosts. The Misc and SuperNova clusters use TCP/IP with MPICH 1.2.7p1, and MAC addresses of hosts are registered by self-learning before the measurements were made. The IEEE 802.3x link-level flow control was enabled at every port.

5.2 Topologies and Path Sets

We constructed both of indirect topologies (fat tree and Myrinet-clos in Fig. 9) and direct topologies (2D mesh and 2D torus) on the testbed. Since the indirect topologies make the best use of the acyclicity of the tree structures, the minimal routings on them are always deadlock-free. The direct topologies used the DOR.

Table 11 lists the topologies evaluated on the testbed. "Avg.H" represents the average number of switches that compose a path. For the purpose of comparison, we employed the "M-tree" topology, which is a mesh-based direct network (the same as VLAN #10 in Fig. 1) with no VLANs.

5.3 Traffic Patterns

We first evaluated the network throughput of each topology for typical traffic patterns with collective communication.

Collective communication is frequently used in parallel programming using MPI. Interconnection networks in parallel computers and certain SANs, such as QsNET [3], thus employ tree-based (hardware) or path-based multicasting, both of which decrease the number of frames per broadcast or multicast operations [18].

On the other hand, Ethernet usually uses unicast-based multicasting. Various MPI functions, such as MPI_Alltoall, MPI_Reduce, MPI_Scatter, and MPI_Barrier, are widely used, and these communication characteristics are important factors in designing efficient path sets.

TABLE 11
Evaluated Topologies in Testbed

Topology	# sw	# links	Avg.H
Mesh (4×2)	8	10	2.75
Torus (4×2)	8	12	2.50
Myri-clos	8	16	2.25
M-tree	16	15	4.81
Mesh (4×4)	16	24	3.50
Torus (4×4)	16	32	3.00
Fat tree	14	24	3.75

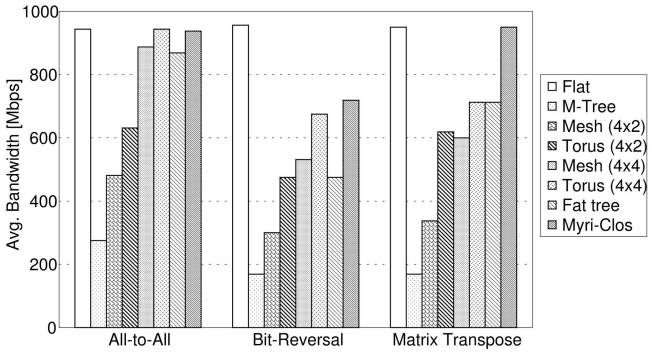


Fig. 10. Traffic pattern results (testbed).

We measured the network throughput using traffic of the all-to-all operation, which is difficult to optimize for the topologies using unicast-based multicasting because all processes communicate with each other at almost the same time. Throughput was also measured for two other synthetic traffic patterns: bit-reversal and matrix transpose. In bit-reversal traffic, a host with the identifier $(a_0, a_1, \ldots, a_{n-1})$ sends a packet to the host whose identifier is the bit reversal $(a_{n-1}, \ldots, a_1, a_0)$ of the source host. In matrix transpose traffic, a host (x, y) sends a packet to the host $(k-y-1, k-x-1)$ (k is the number of hosts in each dimension) or $(k-x-1, k-y-1)$ when $x+y=k-1$.

The number of hosts was 16 in all topologies on the testbed. Thus, in Mesh (4×2) and Torus (4×2), eight switches were each connected to two hosts. The UDP transfer of Tperf-1.5 [24] was used for measuring the bandwidth of each transfer pair, and the sender and receiver processes were run on each host. The frame size was set to the maximum UDP datagram size, 1,470 bytes.

Fig. 10 shows the average bandwidth of all transfer pairs on each topology. We also evaluated a flat 1-switch network (flat, nonblocking full crossbar) in which all 16 hosts were connected to a single switch. Note that such a flat topology is an ideal one for providing full bisection bandwidth; however, it is hardly possible to employ such a topology in a large-scale cluster with thousands of hosts.

Fig. 10 illustrates that path congestion drastically affects the bandwidth between hosts. However, topologies supported by the proposed methodology outperform the single tree-based topology with no VLANs (M-tree). In addition, the performance of four switch-tagged topologies, 4×4 mesh, torus, fat tree, and Myrinet-clos, are comparable to that of an ideal flat topology in all-to-all traffic.

5.4 NAS Parallel Benchmarks

5.4.1 Impact of Topology

We evaluated some of the topologies that had better traffic patterns results on NAS parallel benchmarks (NPBs) 3.2 [27]. The number of processes for parallel execution was fixed to 16 in the case of the PC-cluster testbed, while the SuperNova cluster used 128 processes in CG, FT, IS, LU, and MG and 225 processes (the maximum size) in SP and BT. We compiled all the benchmarks using gcc/g77 3.3 with the -O3 option.

Fig. 11 shows relative performances normalized by the performance of the flat topology and tree topology, respectively. The unit of performance is the execution time (second), and the y-axis is its relative value. A lower value is

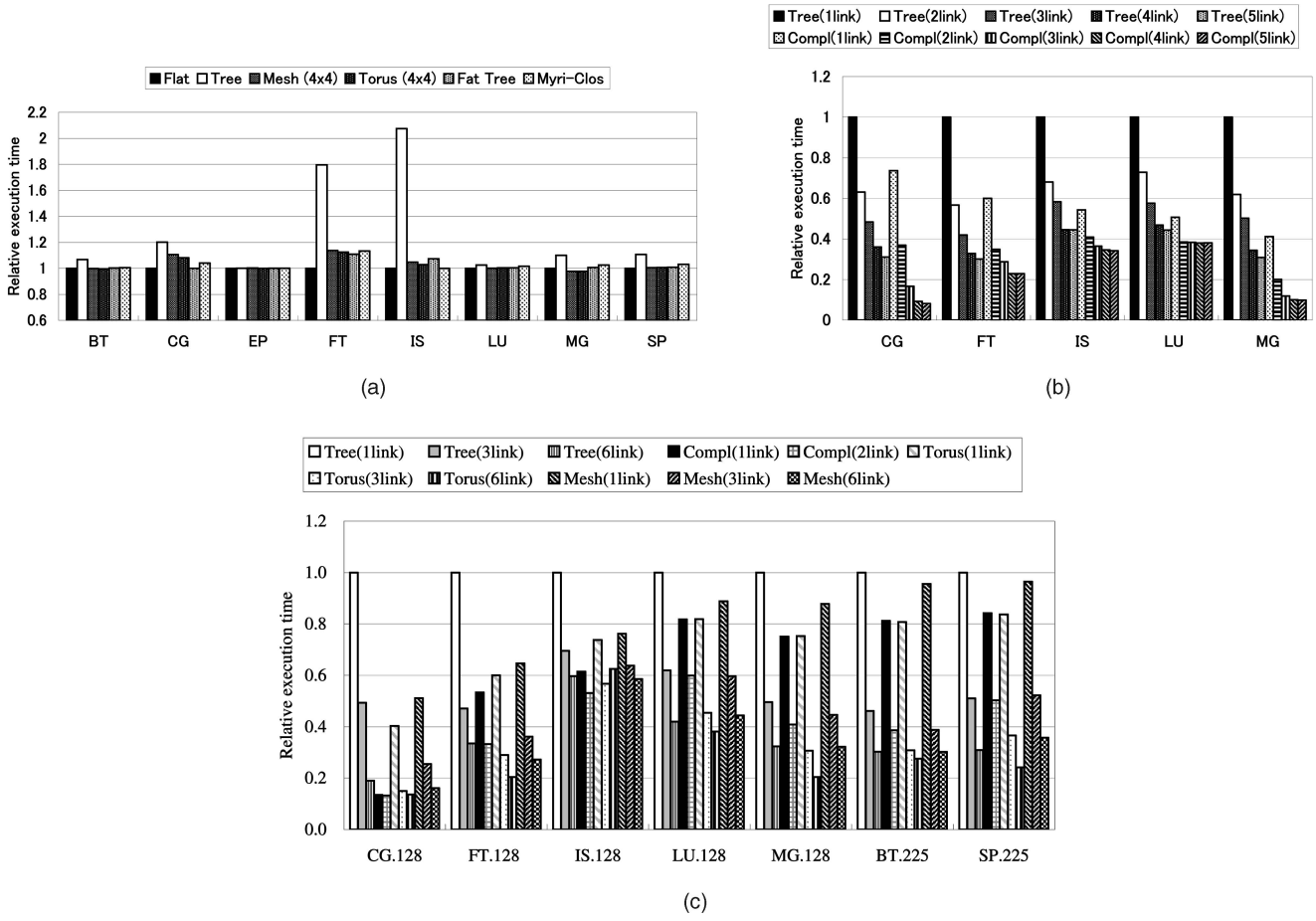


Fig. 11. NAS parallel benchmark results. (a) Testbed, class B, 16 processes. (b) Misc cluster, class C, 128 processes. (c) SuperNova cluster, class C, 128-225 processes.

thus better. "Tree (6link)" stands for the tree topology that uses six links between switches using link aggregation, and "Compl(2link)" stands for a completely connected topology that uses two links between switches in Fig. 11b. In Fig. 11a, the topologies with the proposed methodology have almost the same performance as the flat topology on most benchmarks, while the performance of M-tree is especially degraded on FT and IS. It is known that FT and IS frequently perform the MPI_Alltoall function, and thus, require a large bisection bandwidth.

Fig. 11b shows the results of the NPB on the Misc cluster. Compared with the results on the SuperNova cluster that uses single-core processors in Fig. 11c, the impact of topology on the performance increases. This is because the computation power of each host is higher than that of the SuperNova cluster, even though both clusters used the same Ethernet switches.

Fig. 11c shows that topologies with the proposed methodology improve the performance by up to 650 percent and the tree topology with link aggregation is inferior to the topology including loops. Torus topology with three links between switches, that uses 36 links in total, achieves up to a 27 percent performance improvement compared with the tree topology with six links that uses 42 links in total.

Table 12 gives an itemization of the average computation and communication time of processes. The computation time is almost constant regardless of the topologies used, but communication time is strongly dependent on type of topologies. These results clearly show that the NPB performance is affected by the communication overhead of topologies.

5.4.2 Impact of On/Off Link Regulation

We used a simple static on/off link selection algorithm in order to show the impact of on/off link regulation using the

TABLE 12
Itemized Statement of FT, LU, and SP Benchmarks (SuperNova Cluster) (in Seconds)

	FT.128			LU.128			SP.225		
	Total	Comput	Comm	Total	Comput	Comm	Total	Comput	Comm
Tree(1link)	189.0	8.6	180.4	119.4	23.2	96.2	312.9	16.6	296.3
Compl (2link)	62.7	8.6	54.1	71.7	23.2	48.5	158.0	16.7	141.2
Mesh(6link)	51.4	8.6	42.8	53.2	23.2	30.0	111.7	16.7	95.0

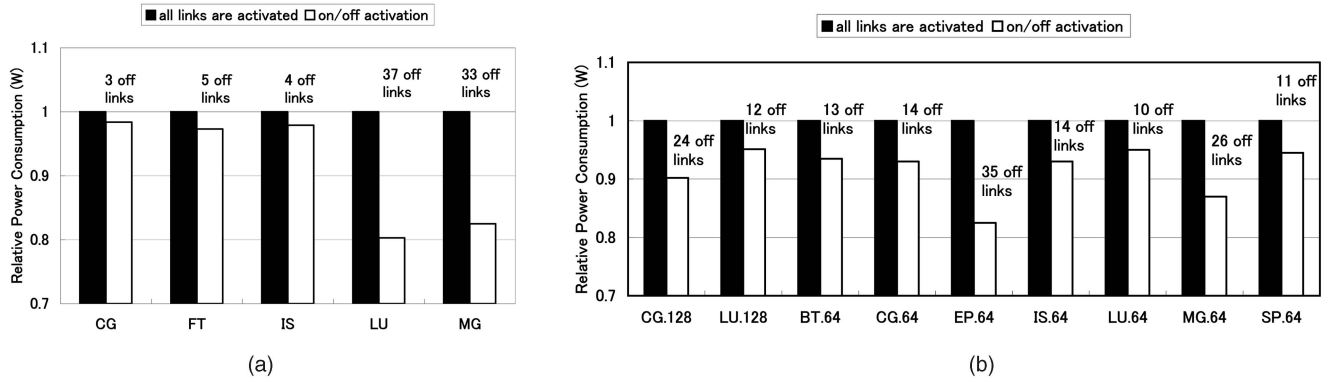


Fig. 12. Power consumption of on/off link regulation (class C). (a) SuperNova cluster, 128 processes. (b) Misc cluster.

proposed methodology on the Misc and the SuperNova clusters, although we could have used more sophisticated means of on/off link selection or multispeed interconnection [28], [29], [30], [31]. The simple static on/off link selection algorithm selects the deactivated links as follows: 1) Estimate the amount of the application traffic in each channel when all the links are activated. The maximum amount of traffic on a single channel is calculated. 2) Reduce the number of links between switches under the condition that the amount of the application traffic on every channel is less than the maximum amount of traffic on a channel calculated in step 1.

Notice that a link consists of two unidirectional channels and there are a large amount of Ethernet monitoring and sampling software, such as IPTraf [20], for measuring the traffic to each destination. In this evaluation, we used the trace files of the applications with the smallest class (W or A) for the traffic estimation in the on/off link selection algorithm for step 1, and we used the unit, byte, in MPI-level communication.

Fig. 12a shows the power consumption of all switches in the Misc cluster. The plot was calculated using the results in Table 2. The numerical values in Fig. 12a show the numbers of deactivated links on the completely connected topology. We confirmed that the execution time of each parallel application with the on/off link regulation is the same as when all links are activated. Thus, the simple on/off link regulation maintained the performance. It is important to keep the cluster's performance close to that when all links are activated, since the execution time of the application strongly affects the total power consumption of the cluster. The simple on/off link regulation does not affect the execution time of the parallel programs, and Fig. 12a illustrates that the power consumption of all switches is reduced by up to 20 percent.

Fig. 12b shows the power consumption of all switches in the SuperNova cluster. The numerical values in the figure are the numbers of deactivated links on the torus topology. In the case of 128 processes with class C, each switch connects to 16 hosts, while each switch connects to eight hosts with class C in the case of 64 processes. Each host executes a single process. As shown in Fig. 12b, the power consumption of all switches is reduced by up to 19 percent, while maintaining the performance.

For the case of 64 hosts in the previous section, each switch used 23 ports: eight ports for connecting hosts and the remaining ports for connecting the neighboring switches. Thus, we can use 24-port switches, i.e., Dell PowerConnect 5324s, instead of PowerConnect 6248s in the 64-host evaluations on the cluster. Fig. 13 plots the estimated total power consumptions based on the results in Table 2.

As shown in Fig. 13, the power consumption can be reduced by up to 25 percent in the case of PowerConnect 5324s. These results show that the on/off link regulation significantly reduces the power consumption of commercial Ethernet switches, while maintaining the performance. Fig. 12b shows that the on/off link regulation more efficiently reduces the power consumption of the simple layer-2 switch PowerConnect 5324 compared with that of the high-functional layer-3 switch (such as PowerConnect 6248), since various services that consume power are always running on the high-functional switches regardless of the link status. In this evaluation, the static on/off link regulation was used so as to adjust the number of links between switches according to the application traffic patterns for simply illustrating its impact on the performance and power consumption. An aggressive on/off link selection algorithm that frequently changes path sets would be needed, when there are few links between switches, or when there are a large number of alternative paths. Such an algorithm can be implemented with the proposed methodology.

6 Conclusions

In this study, we proposed a switch-tagged routing methodology in order to implement various deadlock-free routing algorithms on PC clusters with Ethernet whose MPI

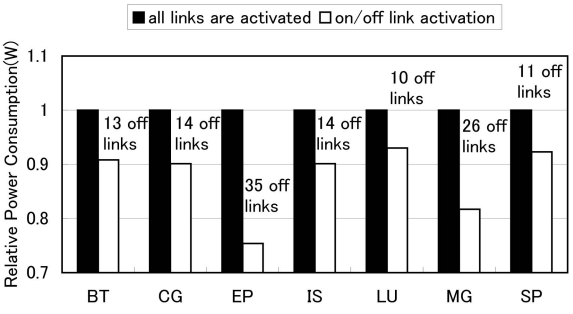


Fig. 13. Power consumption of on/off link regulation (SuperNova cluster, 64 processes, PC5324).

communication libraries do not support tagged VLAN technology. The proposed methodology simply configures each Ethernet switch; it disables the STP, allocates the VLAN sets, and optionally registers static MAC addresses of hosts. Since the MPI communication libraries do not need to perform VLAN operations, the proposed methodology has a simple host configuration and high portability. Fixed and renamed VLAN assignments require at most n and p VLANs, respectively, where n is the number of switches and p the degree of a switch. Thus, renamed VLAN assignment can make a larger PC cluster than that of fixed VLAN assignment, because each VLAN requires different MAC address entries to a single host for routing in each switch which only stores the limited number (e.g., 8,000) of MAC addresses. Fixed VLAN assignment expresses only the $N \times N \mapsto P$ routing relation (all-at-once) where all paths from a host are contained in a tree, while renamed VLAN assignment expresses the $C \times N \mapsto C$ routing relation, where N is the node set, P the path set, and C the channel set. Thus, they have different pros and cons.

Evaluation results using NAS parallel benchmarks showed that the performance of the topologies with loops using the proposed methodology was comparable to that of an ideal 1-switch (full crossbar) network, and the torus topology achieves up to a 27 percent performance improvement compared with the tree topology using link aggregation. In addition, the proposed methodology with the on/off and multispeed link regulation reduces the power consumption of switches by up to 25 percent with almost no performance degradation in the PC clusters that we tested. To minimize the influences of the transition between old and new path sets on the network performance, we plan to develop a switch configuration tool that supports an efficient network reconfiguration in Ethernet with link-level flow control.

ACKNOWLEDGMENTS

This work was partially supported by JST CREST (ULP-HPC: Ultra Low-Power, High-Performance Computing via Modeling and Optimization of Next Generation HPC Technologies). The authors would like to thank Professor Tomoyuki Hiroyasu, Doshisha University, for allowing them to use SuperNova and Misc clusters.

REFERENCES

[1] PC Cluster Consortium, http://www.pccluster.org/, 2010.
[2] InfiniBand Trade Association, http://www.infinibandta.org/, 2010.
[3] F. Petrini, W.C. Feng, A. Hoisie, S. Coll, and E. Frachtenberg, "The Quadrics Network (QsNet): High-Performance Clustering Technology," *Proc. Hot Interconnets 9,* pp. 125-130, Aug. 2001.
[4] M. Koibuchi, K. Watanabe, T. Otsuka, and H. Amano, "Performance Evaluation of Deterministic Routings, Multicasts, and Topologies on RHiNET-2 Cluster," *IEEE Trans. Parallel and Distributed Systems,* vol. 16, no. 8, pp. 747-759, Aug. 2005.
[5] Top 500 Supercomputer Sites, http://www.top500.org/, 2010.
[6] T. Kudoh, H. Tezuka, M. Matsuda, Y. Kodama, O. Tatebe, and S. Sekiguchi, "VLAN-Based Routing: Multi-Path L2 Ethernet Network for HPC Clusters," *Proc. IEEE Int'l Conf. Cluster Computing (Cluster),* Sept. 2004.
[7] S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh, "Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks," *Proc. IEEE INFOCOM,* pp. 2283-2294, Mar. 2004.
[8] T. Otsuka, M. Koibuchi, A. Jouraku, and H. Amano, "VLAN-Based Minimal Paths in PC Cluster with Ethernet on Mesh and Torus," *Proc. Int'l Conf. Parallel Processing (ICPP),* pp. 567-576, June 2005.
[9] F.D. Pellegrini, D. Starobinski, M.G. Karpovsky, and L.B. Levitin, "Scalable Cycle-Breaking Algorithms for Gigabit Ethernet Backbones," *Proc. IEEE INFOCOM,* pp. 2175-2184, Mar. 2004.
[10] A. Jouraku, M. Koibuchi, and H. Amano, "An Effective Design of Deadlock-Free Routing Algorithms Based on 2D Turn Model for Irregular Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 3, pp. 320-333, Mar. 2007.
[11] A. Mejia, J. Flich, J. Duato, S.-A. Reinemo, and T. Skeie, "Boosting Ethernet Performance by Segment-Based Routing," *Proc. 15th EUROMICRO Int'l Conf. Parallel, Distributed and Network-Based Processing,* pp. 55-62, Feb. 2007.
[12] S.-A. Reinemo and T. Skeie, "Effective Shortest Path Routing for Gigabit Ethernet," *Proc. IEEE Int'l Conf. Comm. (ICC),* pp. 6419-6424, June 2007.
[13] R. Garcia, J. Duato, and J.J. Serrano, "A New Transparent Bridge Protocol for LAN Internetworking Using Topologies with Active Loops," *Proc. Int'l Conf. Parallel Processing (ICPP),* pp. 295-303, 1998.
[14] K. Lui, W. Lee, and K. Nahrstedt, "STAR: A Transparent Spanning Tree Bridge Protocol with Alternate Routing," *ACM SIGCOMM Computer Comm. Rev.,* vol. 32, no. 3, pp. 33-46, July 2002.
[15] S. Urushidani, S. Abe, Y. Ji, K. Fukuda, M. Koibuchi, M. Nakamura, S. Yamada, K. Shimizu, R. Hayashi, I. Inoue, and K. Shiomoto, "Design of Versatile Academic Infrastructure for Multilayer Network Services," *IEEE J. Selected Areas in Comm.,* vol. 27, no. 3, pp. 253-267, Apr. 2009.
[16] S. Miura, T. Boku, T. Okamoto, and T. Hanawa, "A Dynamic Routing Control System for High-Performance PC Cluster with Multi-Path Ethernet Connection," *Proc. Workshop Comm. Architecture for Clusters (CAC) in IPDPS,* pp. 1-8, Apr. 2008.
[17] W.D. Dally and B. Towles, *Principles and Practices of Interconnection Networks.* Morgan Kaufmann, 2003.
[18] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach.* Morgan Kaufmann, 2002.
[19] V. Soteriou and L.-S. Peh, "Exploring the Design Space of Self-Regulating Power-Aware On/Off Interconnection Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 3, pp. 393-408, Mar. 2007.
[20] IPTraf: IP Network Monitoring Software, http://iptraf.seul.org/, 2010.
[21] T. Otsuka, M. Koibuchi, T. Kudoh, and H. Amano, "A Switch-Tagged VLAN Routing Methodology for PC Clusters with Ethernet," *Proc. Int'l Conf. Parallel Processing (ICPP),* pp. 479-486, Aug. 2006.
[22] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," *Proc. Int'l Symp. Computer Architecture (ISCA),* pp. 278-287, May 1992.
[23] GtrcNET-1, http://projects.gtrc.aist.go.jp/gnet/, 2009.
[24] Tperf, http://www.am.ics.keio.ac.jp/~terry/tperf/, 2010.
[25] O. Lysne, J.M. Montañana, J. Flich, J. Duato, T.M. Pinkston, and T. Skeie, "An Efficient and Deadlock-Free Network Reconfiguration Protocol," *IEEE Trans. Computers,* vol. 57, no. 6, pp. 762-779, June 2008.
[26] Intel MPI Benchmarks, http://www3.intel.com/cd/software/products/asmo-na/eng/219848.htm, 2010.
[27] The NAS Parallel Benchmarks, http://www.nas.nasa.gov/Software/NPB/, 2010.
[28] L. Shang, L.-S. Peh, and N.K. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," *Proc. Int'l Symp. High-Performance Computer Architecture,* pp. 91-102, Jan. 2003.
[29] J.M. Stine and N.P. Carter, "Comparing Adaptive Routing and Dynamic Voltage Scaling for Link Power Reduction," *IEEE Computer Architecture Letters,* vol. 3, no. 1, pp. 14-17, Jan. 2004.
[30] M. Alonso, J.M. Martinez, V. Santonja, P. Lopez, and J. Duato, "Power Saving in Regular Interconnection Networks Built with High-Degree Switches," *Proc. IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS),* Apr. 2005.
[31] M. Koibuch, T. Otsuka, H. Matsutani, and H. Amano, "An On/Off Link Activation Method for Low-Power Ethernet in PC Clusters," *Proc. IEEE Int'l Symp. Parallel and Distributed Processing (IPDPS),* May 2009.

Michihiro Koibuchi received the BE, ME, and PhD degrees from Keio University, Japan, in 2000, 2002, and 2003, respectively. He was a visiting researcher at the Technical University of Valencia, Spain, and a visiting scholar at the University of Southern California, in 2004 and 2006, respectively. He is currently an associate professor at the National Institute of Informatics (NII) and the Graduate University for Advanced Studies, Japan. His research interests include the areas of high-performance computing and interconnection networks. He is a member of the IEEE and the IEEE Computer Society.

Tomohiro Otsuka received the BE, ME, and PhD degrees in 2001, 2003, and 2009, respectively, from Keio University, Japan, where he is currently a technical staff in the Information Technology Center. His research interests include interconnection networks and communication architecture of high-performance computing platforms.

Tomohiro Kudoh received the PhD degree from Keio University in 1992. In 2002, he joined the National Institute of Advanced Industrial Science and Technology (AIST), where he currently serves as the group leader of the Grid Infraware Research Group of Information Technology Research Institute. In the past few years, his research has focused on network as a Grid infrastructure. His recent work also includes the GridMPI project, which focuses on development of high-performance MPI executed over Grid environment, GbE/10GbE hardware network testbed GtrcNET, and the G-lambda project which target is to define an interface to manage network as a Grid resource. He is a member of the IEEE Computer Society.

Hideharu Amano received the PhD degree in 1986 from Keio University, Japan, where he is currently a professor in the Department of Information and Computer Science. His research interests include the areas of parallel processing and reconfigurable systems. He is a member of the IEEE Computer Society.