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

Tomohiro Otsuka† Michihiro Koibuchi‡ Tomohiro Kudoh* Hideharu Amano†
†Keio University ‡National Institute of Informatics (NII)
{terry,hunga}@am.ics.keio.ac.jp koibuchi@nii.ac.jp
*National Institute of Advanced Industrial Science and Technology (AIST)
t.kudoh@aist.go.jp

Abstract

Ethernet has been used for connecting hosts in the area of high performance-per-cost PC clusters. Although L2 Ethernet topology is limited to a tree structure, various routing algorithms on topologies suitable for parallel processing can be employed by applying IEEE 802.1Q VLAN technology. However, communication library used in current PC clusters does not always support VLANs, so the current design of VLAN-based routing method cannot be applied for such PC clusters.

In this study, we propose a switch-tagged VLAN methodology to flexibly set the route of frames on such PC clusters. Since each host does not need to process VLAN tags, the proposed method has advantages in both simple host configuration and high portability. Evaluation results using NAS Parallel Benchmarks showed that performance of topologies supported by the proposed method was comparable with that of an ideal 1-switch (full crossbar) network in the case of a 16-host PC cluster.

1. Introduction

Ethernet has been used for interconnection networks in various PC clusters because of its high performance-per-cost. Unlike the early Beowulf clusters, recent PC clusters with Ethernet employ system software[11] that supports low-latency zero- or one-copy communication used in system area networks (SANs)[9][3]. High-throughput (non-blocking) commercial Ethernet switches are now also available, and the link bandwidth of Ethernet has rapidly increased, such as 10-Gigabit Ethernet (10GbE) standardization.

When developing a large-scale PC cluster using Ethernet, there are two ways of constructing an intra-cluster Ethernet network: one is to use a switch with a large number of (several hundred or more) ports, and the other is to connect a number of switches each of which has only a few ports. Because of high cost and low availability of such large-scale switches with many ports, the latter way is preferable to make the best use of cost-effectiveness of Ethernet.

However, unlike SANs, most PC clusters using Ethernet currently have employed simple tree-based topologies for their interconnection networks of switches. This is mainly because topologies including loops are not allowed in layer-2 Ethernet, and the routing in Ethernet network has not been focused on research fields concerning PC clusters.

Recently, there are a few works on applying IEEE 802.1Q tagged VLAN technology to the area of metropolitan networks and PC clusters using Ethernet, so as to employ multiple paths between each pair of hosts under various topologies including loops, such as Fat tree and Hyper crossbar[14][6]. They have also illustrated that the VLAN-based routing makes efficient use of link bandwidth under well-distributed paths in small-scale networks.

The existing VLAN-based routing method, however, cannot be easily applied for most of current PC clusters with Ethernet, because the system software and communication library used in such PC clusters do not usually support tagged VLAN technology. There therefore is little work[7] to show the methodology of implementing the VLAN-based routing, its feasibility, and the impact of the VLAN-based routing on large-scale PC clusters with a number of switches.

In this study, we propose a switch-tagged VLAN methodology for PC clusters whose system software and communication library do not support VLAN tagging. By this method, the path between each pair of hosts can be flexibly set and a large bisection bandwidth provided by the VLAN-based routing can be obtained without any modification of the system software at hosts.

The proposed methodology has an advantage of not only such high portability but also simple host configuration. The existing VLAN-based routing method requires a complicated VLAN configuration at each host, i.e. setting vir-

tual network interfaces each of which corresponds to a VLAN ID, or managing multiple IP addresses on the virtual interfaces at a single host when using TCP/IP. On the other hand, the proposed method only requires a VLAN configuration at switches, which is similar to that in the original VLAN-based routing method.

The rest of this paper is organized as follows. In Section 2, we briefly explain the VLAN-based routing method. In Section 3, we describe a summary of the proposed switch-tagged routing method with an example. In Section 4, we propose and illustrate the switch-tagged VLAN routing methodology. In Sections 5 and 6, we evaluate the proposed method and discuss performance factors of the VLAN-based routing. Finally, in Section 7, we describe conclusions.

2. VLAN-based Routing Method

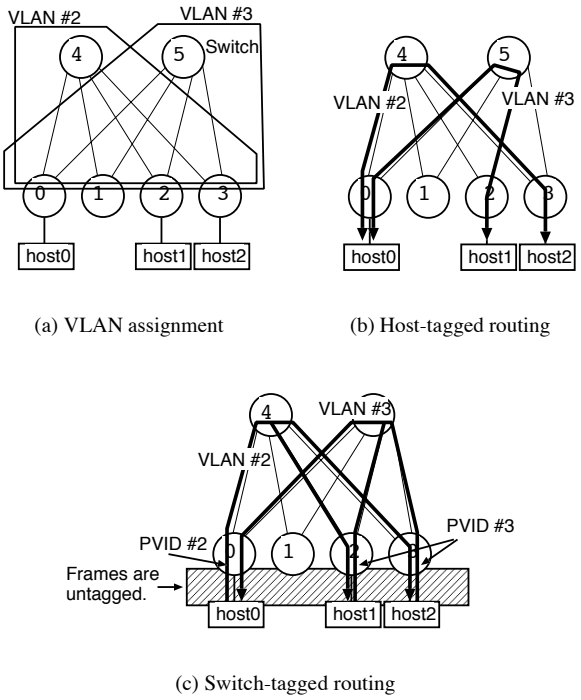
In this section, we briefly explain the original VLAN-based routing method proposed by Kudoh et al.[6].

In the VLAN-based routing method, multiple paths between hosts are obtained as follows: in a physical network including loops, multiple VLANs, each of which is a different spanning tree of the physical network from others, are assigned. Each host is configured as a member of all the VLANs, i.e. it has virtual network interfaces corresponding to all VLANs. In this way, all pairs of hosts can communicate with each other via any VLAN topology, and multiple paths that consist of different link sets are available 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 Ethernet frame tagged with the specified VLAN ID is transferred by usual L2 Ethernet mechanism within the VLAN topology. Although each VLAN topology is logically a tree, physical topologies of L2 Ethernet are thus free from tree-based structures, and various routing algorithms on a flexible path set including all existing links can be employed by introducing multiple VLANs[10].

An example of the VLAN-based routing is shown in Figures 1(a) and (b). There are two VLANs, VLAN #2 and #3, each of which is a spanning tree of the physical network including loops. Each host has two virtual interfaces and can send frames to all destination hosts via either of two VLANs as shown in Figure 1(b).

Note that most Ethernet switches support IEEE 802.1D STP (Spanning Tree Protocol) or IEEE 802.1w RSTP (Rapid STP) to prevent loops in a network. Unfortunately, STP and RSTP are not aware of VLANs, so when these protocols are enabled, all links out of a spanning tree are automatically disabled. Therefore, STP and RSTP must be disabled when the VLAN-based routing is used. IEEE 802.1s



(a) VLAN assignment

(b) Host-tagged routing

(c) Switch-tagged routing

Figure 1. VLAN-based routing on a Fat tree

MSTP (Multi STP) and Cisco Systems' PVST (Per VLAN Spanning Tree) are both STPs which support VLANs. They are quite useful for the VLAN-based routing, but currently there are only a few cost-effective Ethernet switches which support these protocols.

3. Summary of Switch-tagged Routing

In this section, we show the concept and an example of the proposed switch-tagged VLAN routing method.

3.1. Frame Tagging in VLAN Switches

The switch behavior in VLAN tagging operation is as follows; when an untagged Ethernet frame enters a port, it is tagged with the default VLAN ID tag number (port VLAN ID, PVID), while a tagged frame enters a port with no effect on the tag.

On the other hand, 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 untagged.

3.2. Switch-tagged Routing Method

In the switch-tagged VLAN routing method, all paths from a host belong to a single VLAN regardless of a destination host. Both VLAN tagging and untagging operations are performed at each switch port connected with a host, by configuring as follows.

- Set PVID of each port to the ID of the VLAN that is used by the connected host when sending frames.
- Register each port as an "untagged" member of all VLANs in the whole network.

A source host transmits a normal (untagged) frame in the usual way by specifying IP address or MAC address of a destination host. When a frame from a host enters a port of a switch, it is tagged with PVID of the port and is regarded as a frame which belongs to a VLAN indicated by the ID tag number. The frame is then transferred by L2 Ethernet mechanism as well as the normal VLAN-based routing. Finally, the frame is untagged when it leaves a port which is connected to the destination host, because the port is an "untagged" member of the VLAN. The destination host thus receives a usual untagged frame, and the communication library can easily handle it.

In this way, all hosts can communicate with each other on various topologies, even if a VLAN tagging operation is not supported by communication library on hosts.

An example of the switch-tagged routing is shown in Figures 1(a) and (c). The port of switch 0 connected with host 0 is configured as PVID #2, and the ports of switches 2 and 3 connected with hosts 2 and 3 respectively both have PVID #3. Thus, frames transmitted by host 0 are routed in VLAN #2 for all destinations, while those by hosts 1 and 2 use VLAN #3.

3.3. Restrictions of the Switch-tagged Routing

The switch-tagged routing method imposes restrictions on a path set compared with the original VLAN-based routing; that is, all paths from each host must be contained by a single VLAN. For example, the path set by the original method shown in Figure 1(b) cannot be employed by the switch-tagged routing, because paths from host 0 use two VLANs, #2 and #3.

This routing restriction would introduce difficulty in establishing balanced minimal paths on complicated topologies. However, for simple topologies used in parallel computers, a VLAN set with such balanced paths can easily be assigned (see Section 4.3).

Another problem introduced by the switch-tagged method is about MAC address learning on switches. Ethernet switches usually learn unknown MAC addresses when receiving frames. However, since the VLAN-based routing method can employ multiple VLANs, there are many cases where a path from host A to B and that from B to A use different VLANs. In these cases, the intermediate switches of both paths cannot learn the destination MAC address even if these two paths consist of the same intermediate switches, because the MAC address learning is performed on each VLAN independently.

This problem is also applicable to the original VLAN-based routing, however, in the switch-tagged routing, only one VLAN in the whole topology is allowed to avoid using different VLANs between each pair of hosts. Fortunately, in recent commercial Ethernet switches, such as DELL PowerConnect 5324, pairs of MAC address and output port number can be statically registered to the table. Therefore, we assume such static MAC address registration on each switch for the proposed switch-tagged routing method.

4. Switch-tagged Routing Methodology

In this section, we illustrate a switch-tagged VLAN methodology to flexibly set the route of frames. First, we express possible path sets of the switch-tagged routing method. Second, we show a deadlock-free operation of routing algorithms in order to avoid a deadlock problem which the VLAN-based routing method newly introduces. Thirdly, we state the switch-tagged routing method.

4.1. Routing Relation

In general, there are three functions for representing routing algorithms. The simplest routing function is based on $N(source) \times N(destination) \mapsto P$ routing relation (all-at-once)[1], where N and P are the node set and the path set respectively. The other routing functions are based on $N \times N \mapsto C$ routing relation, which only takes into account the current and the destination nodes[2], and $C \times N \mapsto C$ routing relation, where C is the channel set. In the case of SANs or interconnection networks in parallel computers, the format of (distributed) routing tables at routers corresponds to the routing relation.

On the other hand, in the case of switch-tagged VLAN method, the above routing relations do not fit with the possible sets of frame routes. Therefore, we introduce a new routing relation, $V \times N \mapsto C$, so as to express possible routing algorithms, where V is the VLAN set.

Note that $V \times N \mapsto C$ routing relation can express some simple deterministic routing algorithms used in regular topologies, such as mesh (see Section 4.2).

4.2. Performing Deadlock-free Operation

All path sets that are expressed by $V \times N \mapsto C$ routing relation can be employed in the proposed methodology. However, this adaptability introduces possibility of deadlocks of frames even in a L2 Ethernet network that employs store-and-forward as a switching technique.

In general, Ethernet frames are simply discarded when buffers of a switch or a network interface of a host are filled. An upper end-to-end protocol such as TCP usually retransmits the discarded frames and a deadlock does not occur, even though the network throughput is decreased. However, when IEEE 802.3x link-level flow control is used as well as SANs, frames are hardly discarded. In this case, deadlocks can occur and the network throughput is drastically decreased, while the link-level flow control is efficient for sharing bandwidth among conflicting paths (see Section 5.3).

To avoid deadlocks of frames, deadlock-free deterministic routing algorithms with no virtual channels used in parallel computers can be used[2][13]. They guarantee deadlock freedom by breaking all cyclic dependencies in the channel dependency graph (CDG). We show the impact of deadlocks on bandwidth in Section 5.2.

4.3. VLAN Methodology for Routing Algorithms

A path set which is expressed by $V \times N \mapsto C$ relation can be implemented in the switch-tagged VLAN method with the following simple procedure on the n-switch Ethernet network.

1. For every switch, make a logical tree which is just composed of paths from the switch to all other switches. n logical trees, each of which includes n switches connected with $n-1$ links, are totally built.

2. If the number of logical trees is not greater than the number of VLANs supported by switches, assign each logical tree to a unique VLAN and finish this procedure.

3. Combine two or more logical trees into a single tree, if their topologies are the same, then go back to step 2. If there is no pair of the same two logical trees, the target path set cannot be used in the Ethernet switches[1].

IEEE 802.1Q VLAN tag field can identify 4,094 ($2^{12} - 2$) VLANs, and current cost-effective Ethernet switches usually support at least several hundred VLANs. So the number of VLANs used in the network would be a trivial matter in the case of most PC clusters with Ethernet.

[1]Note that a combined topology of two different logical trees always includes cycles.

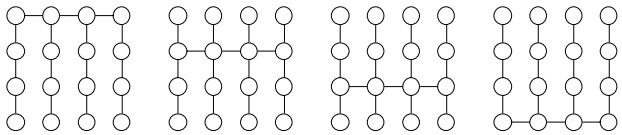


Figure 2. The DOR VLANs in 4 × 4 2-D mesh

As an example of the switch-tagged routing with small number of VLANs, we show the path set for the dimension-order routing (DOR), which is a typical minimal deadlock-free deterministic algorithm on mesh and torus. The DOR switch-tagged VLAN set with the smallest number of VLANs consists of the following N VLANs in an $N \times N$ mesh[10][2]; $\{\mathrm{VL}(-, y) \mid 0 \leq y < N\}$ as shown in Figure 2. Figure 2 shows that four VLANs, VL$(-, 0)$, VL$(-, 1)$, VL$(-, 2)$ and VL$(-, 3)$, are employed to take paths that are along the DOR in the 4×4 2-D mesh. A path from a source switch (x, y_S) is assigned to the VLAN VL$(-, y_S)$. For M-dimensional mesh, at least N^{M-1} VLANs are required, where N is the number of switches per dimension and M is the number of dimensions. Note that, at switch (x, y_S), a PVID corresponding to the VLAN VL$(-, y_S)$ is set to all ports that are connected with hosts.

5. Fundamental Evaluation

In this section, we show the fundamental evaluation results using a real PC cluster and demonstrate performance factors of the proposed switch-tagged methodology.

5.1. Evaluation Environment

Table 1 lists the specifications of each host in the cluster. We used DELL PowerConnect 5324 (Gigabit Ethernet × 24 ports, non-blocking) for switches.

Table 1. Specifications of each host

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)

[2]The result of the above procedure is indeed the same as the VLAN set shown in [10]. We thus briefly show the example.

In each host, SCore Cluster System Software[11] version 5.8.2 is employed. SCore is an open-source cluster system software package which 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[8]. In addition, Tperf-1.5[16] and Intel MPI Benchmarks (IMB) 2.3[4] are used for measuring performance of TCP/UDP transfer and MPI-level transfer between each pair of hosts respectively.

5.2. Impact of Deadlocks on Ethernet

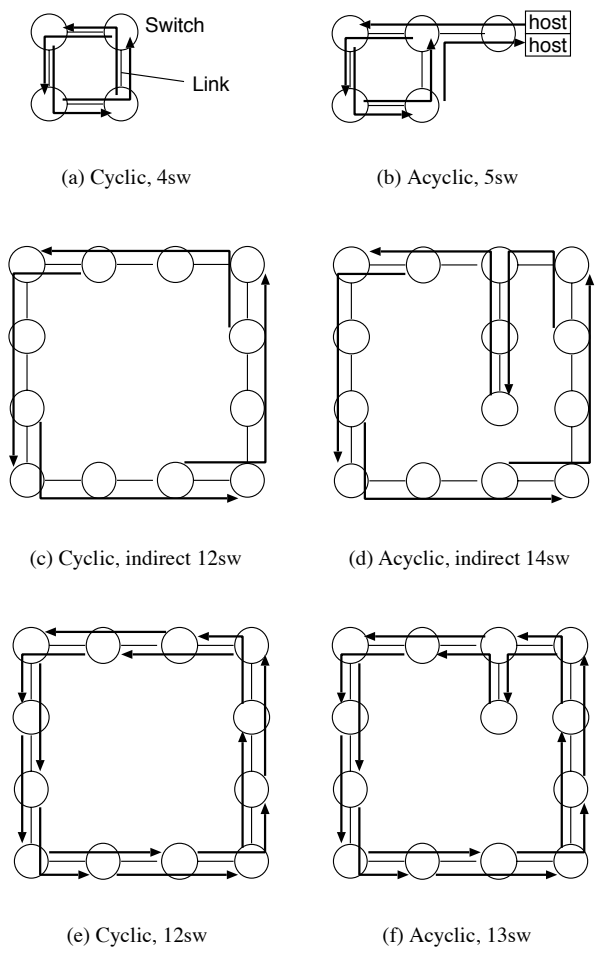
In Section 4.2, we described that a deadlock-free routing should be employed even in L2 Ethernet. In this subsection, to certify the requirement of deadlock freedom, we demonstrate the influences of deadlocks on bandwidth using the real system.

Six frame transfer patterns shown in Figure 3 are employed in order to investigate the impact of various cycles in a network. In each of cyclic frame transfers shown in Figures 3(a), (c) and (e), there is a cycle that would introduce deadlocks. On the other hand, in each of acyclic (deadlock-free) frame transfers shown in Figures 3(b), (d) and (f), no cycles of frames are created. The number of conflicting transfer on a channel is up to two in both cases, so the conditions for cyclic and acyclic frame transfers are the same, except for the cyclic dependencies.

Table 2 lists the average bandwidth in $Mbps$ and frame loss ratio between each sender and receiver pair under various conditions of link-level flow control. "FC None" is the case when the link-level flow control is disabled, while "FC All" is the case where the flow control is enabled at every link. "FC Host" is the case when the flow control is enabled only at links between a host and a switch, and "FC SW" is the case when the flow control is enabled only at links between switches.

Table 2 illustrates that the bandwidth is degraded in most of cyclic transfer patterns. In particular, when link-level flow control between switches is enabled ("FC All" and "FC SW"), the bandwidth is drastically degraded and is often close to zero.

We have analyzed the frames transferred between switches in pattern (a) using GtrcNET-1[5], which is a programmable network testbed and can also precisely monitor Ethernet traffic. Then we found that a large number of PAUSE frames which stop transfer of data frames are transferred between each link within the cycle in cases of "FC All" and "FC SW". Since PAUSE frames are transferred independently of VLAN topologies, we consider that the drastic degradation of bandwidth in cyclic transfer patterns are mainly caused by looping of PAUSE frames. Therefore, when IEEE 802.3x link-level flow control is used, deadlock freedom is required to avoid frame looping.



(a) Cyclic, 4sw (b) Acyclic, 5sw

(c) Cyclic, indirect 12sw (d) Acyclic, indirect 14sw

(e) Cyclic, 12sw (f) Acyclic, 13sw

Figure 3. Cyclic and acyclic frame transfers

Moreover, even if flow control between switches are disabled ("FC None" and "FC Host"), the bandwidth in acyclic transfers is improved up to 17% compared with that in cyclic transfers in all UDP cases. These results demonstrate that deadlock-free routing is quite efficient for not only avoiding deadlocks but also fully using the link bandwidth. We illustrate the effectiveness of link-level flow control in Section 5.3.

5.3. Impact of Path Hops and Traffic Congestion

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

Figure 4 represents MPI-level one-way latency measured by IMB PingPong benchmark and bidirectional bandwidth measured by IMB PingPing benchmark, changing the number of intermediate switches between hosts. The

Table 2. Bandwidth and frame loss ratio with cyclic and acyclic transfers

	FC None	FC All	FC Host	FC SW
(a)/UDP	345.2	209	477.2	0.3
	(64%)	(23%)	(0.07%)	(99.9%)
(b)/UDP	403.3	317.7	506.4	230.1
	(58%)	(0.02%)	(7%)	(76%)
(c)/UDP	339.0	464.9	477.2	0.3
	(65%)	(0.09%)	(0.1%)	(99.9%)
(d)/UDP	344.1	476.8	477.5	247.0
	(64%)	(0.02%)	(0.04%)	(74%)
(e)/UDP	349.1	1.9	479.4	19.2
	(64%)	(11%)	(1%)	(98%)
(f)/UDP	377.8	166.7	486.6	109.1
	(60%)	(0%)	(3%)	(89%)
(a)/TCP	445.0	0.8	454.6	90.4
(b)/TCP	465.4	317.6	440.9	345.0
(c)/TCP	462.2	469.3	466.4	401.4
(d)/TCP	443.2	464.1	469.7	427.9
(e)/TCP	465.6	47.8	447.0	153.9
(f)/TCP	472.4	158.3	416.6	227.5

figure illustrates that the number of path hops is crucial for bandwidth as well as latency between hosts. We consider that this is mainly because end-to-end flow control and Ack/Nack operations are performed by the PM communication library.

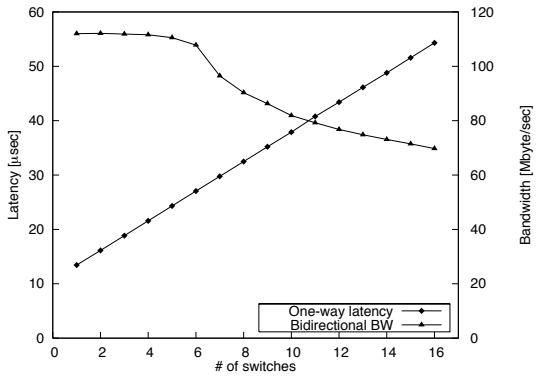


Figure 4. MPI latency and bandwidth

We also evaluated the effectiveness of IEEE 802.3x link-level flow control on bandwidth under the situation of path congestion. Figure 5 represents the average bidirectional bandwidth between eight pairs of hosts using IMB Multi-PingPing benchmark, in cases of "FC None" and "FC All" described in Section 5.2. Each path between the host pairs includes the same two switches and a link between the two switches, so all paths are conflicting on the inter-switch link.

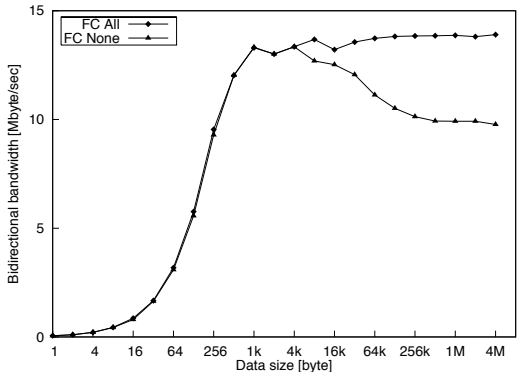


Figure 5. Impact of traffic congestion

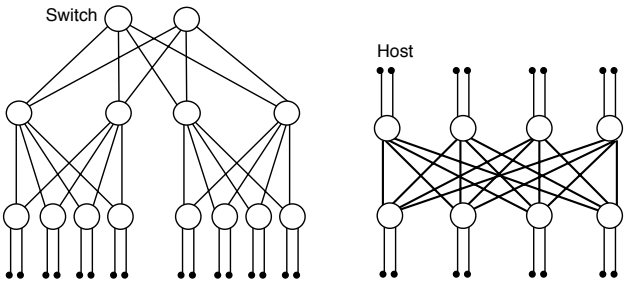
The figure illustrates that IEEE 802.3x link-level flow control is efficient for sharing bandwidth among conflicting paths. Note that the maximum bandwidth of "FC All" case (13.9Mbyte/sec) is about 1/8 of that of the two-switches case in Figure 4 (112.2Mbyte/sec).

6. Evaluation of Switch-tagged Routing

In this section, we show the evaluation results of the proposed switch-tagged routing method on various topologies.

6.1. Topologies and Path Sets

We constructed both of indirect topologies (Fat tree and Myrinet-Clos in Figure 6) and direct topologies (2-D mesh and 2-D torus). Since these indirect topologies make the best use of acyclicity of tree structures, all minimal routings on them are always deadlock-free and mostly expressed by $V \times N \mapsto C$ relation. In the direct topologies, the DOR algorithm shown in Section 4.3 is employed.



(a) Fat tree (2,4,2) (b) Myrinet-Clos (4×4)

Figure 6. Evaluated indirect topologies

Table 3 lists the evaluated topologies. "Avg.H" represents the average number of switches that compose a path, while "Max.H" is the maximum. "CP" represents the maximum number of conflicting paths on a single channel in the case of all-to-all communication among 16 hosts (see Section 6.2). For the purpose of comparison, we employed "M-Tree" topology, which is a mesh-based direct-network (the same as VL$(-, 0)$ in Figure 2) and requires no VLAN support.

Table 3. Evaluated topologies

Topology	#sw	#link	Avg.H	Max.H	CP
Mesh (4×2)	8	10	2.75	5	24
Torus (4×2)	8	12	2.50	3	32
Myri-Clos	8	16	2.25	3	12
M-Tree	16	15	4.81	10	64
Mesh (4×4)	16	24	3.50	7	16
Torus (4×4)	16	32	3.00	5	12
Fat tree	14	24	3.75	5	16

We statically register MAC addresses of hosts at each switch on each VLAN in order to avoid flooding caused by unknown MAC addresses. IEEE 802.3x link-level flow control is enabled at every link.

6.2. Traffic Patterns

We first evaluate the network throughput of each topology under some typical traffic patterns of collective communication.

Collective communication is frequently used in parallel programming using MPI. Interconnection networks in parallel computers or some SANs, such as QsNET[12], thus employ tree-based (hardware) or path-based multicasting, both of which decrease the number of frames per broadcast or multicast operations[2].

On the other hand, in the case of Ethernet, only a unicast-based multicasting has been employed. Various types of MPI functions, such as MPI_Alltoall, MPI_Reduce, MPI_Scatter and MPI_Barrier, have been widely used, and these communication characteristics are important factors in designing efficient path sets.

We measured the network throughput using a traffic of all-to-all operation, which is difficult to optimize using unicast-based multicasting because all processes communicate with each other at almost the same time. Two other synthetic traffic patterns, bit-reversal and matrix transpose, are also used. These traffic patterns frequently appear in parallel programs, especially with unicast-based collective communication[2].

The number of hosts used is 16 on all topologies, so in Mesh (4×2) and Torus (4×2), each of eight switches are connected with two hosts. The UDP transfer of Tperf-1.5 is used for measuring the bandwidth of each transfer pair, and sender and receiver processes are both run on each host. Each frame size is set to the maximum UDP datagram size, 1470 bytes.

Figure 7 represents the average bandwidth of all transfer pairs on each topology. We also evaluated a flat 1-switch network (Flat, non-blocking full crossbar) in which all 16 hosts are connected to a single switch. Note that such a flat topology is an ideal one for providing full bisection bandwidth, but it is hardly possible to employ such a topology in a large-scale cluster with thousands of hosts (see Section 1).

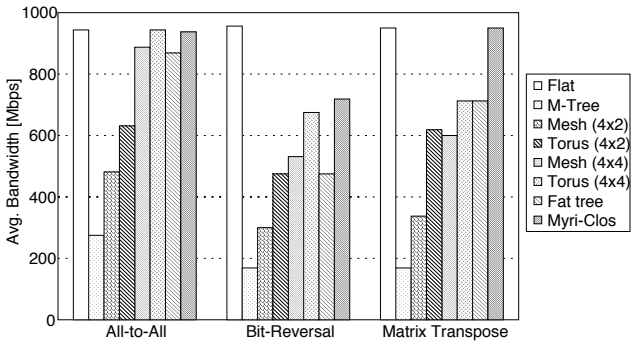


Figure 7. Traffic patterns results

Figure 7 illustrates that a path congestion drastically affects the bandwidth between hosts. However, topologies supported by the proposed switch-tagged method outperform a single tree-based topology with no VLANs (M-Tree). In addition, the performance of four switch-tagged topologies, 4×4 mesh and torus, Fat tree and Myrinet-Clos, are comparable with that of an ideal flat topology in all-to-all traffic.

6.3. NAS Parallel Benchmarks

Next, we evaluate some topologies which had better results in the traffic patterns results using NAS Parallel Benchmarks 3.2[15]. The problem size is Class B and the number of processes for parallel execution is fixed to 16. We compiled all the benchmarks using gcc/g77 3.3.2 with -O3 option.

Figure 8 shows a relative performance normalized by the performance of Flat topology. Topologies by the proposed switch-tagged method yield almost the same performance of Flat topology in most benchmarks, while performance of M-Tree is sometimes degraded especially in FT and IS benchmarks. It is known that FT and IS frequently perform MPI_Alltoall function and thus require a large bisection bandwidth.

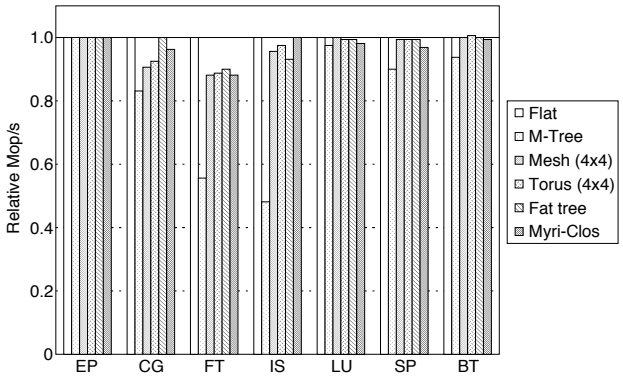


Figure 8. NAS Parallel Benchmarks results

These results conduct that the proposed switch-tagged routing method makes efficient use of link bandwidth with a large number of switches and provides scalability on PC clusters with Ethernet.

7. Conclusions

In this paper, we proposed a switch-tagged VLAN methodology to flexibly set the route of frames on PC clusters with Ethernet, in which the system software does not support VLANs. Since each host does not need to process VLAN tags, the proposed method has advantages of both simple host configuration and high portability. The flexibility of VLAN-based paths may introduce loops in the cyclic dependency graph (CDG) of the network and the loop degrades bandwidth even in the case of Ethernet, so the proposed method is based on a deadlock-free path set.

Evaluation results using NAS Parallel Benchmarks showed that the proposed switch-tagged routing improves performance in most applications compared with the existing routing with no VLANs. The performance of 2-D mesh, 2-D torus and Myrinet-Clos topologies supported by the proposed method is comparable with that of an ideal 1-switch (full crossbar) network which can be employed only in small-scale clusters. These results conduct that the proposed routing method makes efficient use of link bandwidth with a large number of switches and provides scalability on PC clusters with Ethernet.

As a future work, we are currently planning to measure the overhead of VLAN tagging at switches, and to compare performance of proposed switch-tagged method with the original VLAN-based routing or some other tree-based topologies with no VLANs. In addition, to ensure that the VLAN-based routing can operate stably and safely, we are also planning to introduce a fault tolerant strategy using link aggregation for arbitrary topologies.

Acknowledgments

This work was supported by Joint Research Fund, "Parallel and Distributed Processing using Ethernet", National Institute of Informatics.

References

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

[2] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: an engineering approach*. Morgan Kaufmann, 2002.

[3] InfiniBand Trade Association. `http://www.infinibandta.org/`.

[4] Intel Cluster Toolkit. `http://www.intel.com/cd/software/products/asmo-na/eng/cluster/clustertoolkit/`.

[5] Y. Kodama, T. Kudoh, R. Takano, H. Sato, O. Tatebe, and S. Sekiguchi. GNET-1: Gigabit Ethernet Network Testbed. In *Proc. of 2004 IEEE International Conference on Cluster Computing (Cluster2004)*, Sept. 2004.

[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. In *Proc. of 2004 IEEE International Conference on Cluster Computing (Cluster2004)*, Sept. 2004.

[7] S. Miura, T. Okamoto, T. Boku, M. Sato, and D. Takahashi. Low-cost High-bandwidth Tree Network for PC Clusters based on Tagged-VLAN Technology. In *Proc. of the 8th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2005)*, pages 84–93, Dec. 2005.

[8] MPICH. `http://www-unix.mcs.anl.gov/mpi/mpich/`.

[9] Myricom. `http://www.myri.com/`.

[10] T. Otsuka, M. Koibuchi, A. Jouraku, and H. Amano. VLAN-based Minimal Paths in PC Cluster with Ethernet on Mesh and Torus. In *Proc. of the 2005 International Conference on Parallel Processing (ICPP-05)*, pages 567–576, June 2005.

[11] PC Cluster Consortium. `http://www.pccluster.org/`.

[12] F. Petrini, W. C. Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network (QsNet): High-Performance Clustering Technology. In *Proc. of Hot Interconnets 9*, pages 125–130, Aug. 2001.

[13] S.-A. Reinemo and T. Skeie. Ethernet as a Lossless Deadlock Free System Area Network. In *Proc. of Third International Symposium on Parallel and Distributed Processing and Applications (ISPA'05)*, pages 901–914, Nov. 2005.

[14] S. Sharma, K. Gopalan, S. Nanda, and T. cker Chiueh. Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks. In *Proc. of 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, pages 2283–2294, Mar. 2004.

[15] The NAS Parallel Benchmarks. `http://www.nas.nasa.gov/Software/NPB/`.

[16] Tperf. `http://www.am.ics.keio.ac.jp/~terry/tperf/`.