Ethernet-Based Interconnections for Massively Parallel Clusters[1]

Vara Varavithya and Thongchai Thepuatrakul
Department of Electrical Engineering
King Mongkut's Institute of Technology North Bangkok
1518 Piboonsongkram Rd., Bangsue Bangkok 10800, Thailand.
E-mail: vara@hpc.ee.kmitnb.ac.th

ABSTRACT -- Large production volume of the devices results in very low equipment cost based on Ethernet Technology. Advances mix-signal VLSI chip, DSP and analog signal, lead to two order of magnitude of bandwidth improvement in the Ethernet Network. The performance of the Ethernet technology is further enhanced by the development of Ethernet switches in which the aggregated bandwidth is much larger than the broadcast hub. The large scale clusters interconnect using Ethernet is considered as a economical solution. The Ethernet topologies for massively parallel clusters are discussed in the paper. Based on classical network topologies, we proposed Stack Ring and Stack Mesh topologies for the large systems. The stack of Ethernet switch is considered as a single lump node and connected using ring and mesh topologies. A set of the processors are assigned to perform the forwarding tasks. The topology definitions are defined and the IP assignment algorithms for both ring and mesh networks are presented. The effects of forwarding overhead is evaluated and HPL benchmark was tested on the system.

Keywords -- Cluster of workstations, network topology, Ethernet, high performance computing.

1. Introduction

A parallel high performance computing platform is made more accessible by interconnecting a group of workstations via a high speed interconnection network [1, 2, 3]. Examples of applications that can benefit from a cluster of computers, include computational fluid dynamics, weather forecast, bioinformatics, transaction computing, and Internet information servers. A Beowulf-class cluster [4] adopts commodity products, both hardware and software, to construct high performance parallel systems. It has been estimated in [5] that there are currently more than 100,000 clusters around the globe. The improvement in microprocessor and network technologies further drives the realization of these clusters.

A network with low latency and high bandwidth is required to sustain high performance in multicomputers [6]. These systems require that the network latency is in order of a few micro seconds and bandwidth is in order of a few Gbps. System area networks (SANs) [7, 8, 9, 10], are designed to transfer information at very high data rate in a relatively short distance. Although SAN offers very low latency and high bandwidth, the price of the SAN is expensive. Currently, the cost of a single SAN network interface card can exceed the cost of the computing node itself. ATM technology is another candidate as cluster interconnect. The cost of ATM equipments is still relatively high. Comprehensive treatments on interconnection networks for multicomputers is presented in [11].

The *Ethernet* technology is a strong candidate when the cost is considered as an important design requirement. Because of commodity products has very large production volumes the price of Ethernet devices is relatively low. The bandwidth of Ethernet technology has evolved from 10Mbps, 100Mbps, and 1Gbps where 10Gbps Ethernet is around the conner. The performance of current generation Ethernet network is lower than that of system area networks due to hardware speed and heavy communication library [12]. Higher bandwidth in the next generation hardware and supporting some of the communication protocol in the network interface cards will reduce the performance gap between the Ethernet and SANs. However, Ethernet-connected clusters have proven to be suitable for computational intensive applications and have been widely implemented [13].

Carrier Sense Multiple Access (CSMA), adopted in Ethernet, is a contention-based protocol in which network performance is severely degraded in high volume traffic. The collision problem is partially solved using hardware switch [14] at the data link level. *Ethernet Switch* has more aggregation bandwidth compared to the broadcast bus where, with no output port conflict, multiple communication messages can be exchanged in parallel. Most of the Ethernet devices today are shipped as switch-based devices. The contention Ethernet hub is obsolete from the market. In [15], multiple network interface units are implemented in the computing nodes to increase aggregate bandwidth. The channel bonding technique was proposed to provide alternative paths from the source to the destination

where the computing nodes are connected in the mesh-like topology.

This paper investigates the Ethernet topologies for large scale cluster interconnections that take advantage of contemporary consumer switching devices. There are several interesting issues in exploring Ethernet technology as an interconnection in closely connected cluster. A certain class of Ethernet switches has stackable capability. A set of Ethernet switches can be connected together using special backplane connection cables. The number of ports is multiply increased using this technique without performance degradation. We consider a group of stackable switches as a basic building block. The scalability of the Ethernetconnected cluster is therefore limited by the number of ports in one stack of switches, typically 80-120 ports. An Ethernet network for a large cluster requires more complicate details of implementation to maintain performance. Two classical network topologies, ring and mesh, are applied to the stacks of Ethernet Switches. We propose the *Stack-Ring* (SR) and the *Stack-Mesh* (SM) as interconnection topologies for Ethernet-connected massively parallel clusters. The SR/SM interconnects groups of *stackable switches* in ring/mesh with wrap around links. All destinations are being covered by assigning routing tasks to the nodes in a distributed manner. Beowulf-class clusters can be configured using regular IP addressing and forwarding schemes. The forward selection and routing setup schemes are presented for the proposed topologies. The communication overheads associated with the proposed topology were measured and its scalability is evaluated. The performance results of the real applications based on HPL benchmark were tested and compared.

This paper is organized as follows. Section 2 presents preliminaries. The proposed Ethernet topologies are discussed in Section 3. The network parameters and performance are compared in Section 4. Concluding remarks are drawn in Section 5.

2. Preliminaries

A cluster is a collection of workstations interconnecting via an interconnection network. Figure 1 (a) shows the general architecture of a cluster. A computing node is an autonomous computer which has its own processor(s), memory, hard drive, network interface, shown in Figure 1 (b). Communication between computing nodes is accomplished by passing messages. The performance of these computers (nodes) varies from the PCs to the high-end workstations. Due to high aggregation bandwidth and lower cost, the Ethernet switch is usually adopted as a core network devices. The developments of parallel applications can be efficiently achieved using parallel programming development platforms such as PVM [2] and MPI [16]. The system softwares that support networking and parallel programming need to be install on all the nodes. Another important service provided by the operating system is the network file system (NFS) that transparently services the file system to all the computing nodes.

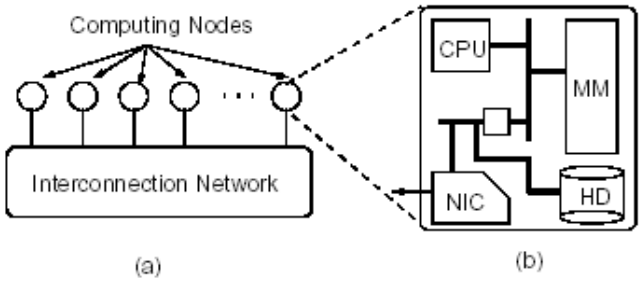

Figure 1. Cluster architecture: (a) Clusterarchitecture, (b) Computing node architecture

Relevant terminologies and network parameters are defined as follows. A *cluster interconnection network*, G is a strongly connected graph, $G(V,C)$, where V represents the set of vertices and C represents a set of physical links connecting the nodes. A set of vertices V in the cluster is divided into two sets, a set of *nodes* (computers) N_c and a set of *switches* N_s. A *node degree* I_c is the number of network interface at each node and a switches degree I_s is the number of ports at each switch. The switch backplane has peak bandwidth of *SP*. The maximum number of switches in a single stack is equal to SK_n. A group of switches in a single stack is considered as a lump switching device called *Stack Switch Box* (SSB). The maximum number of ports in one SSB is therefore equal to $I_s \times SK_n$.

The *network diameter* D is a shortest distance between any two remotest nodes in the network. Each link has *b* bit-per-second bandwidth. *Bisection Bandwidth BW* is the amount of information that can flow between two equal halves of the nodes in the network [11]. The performance metrics for cluster interconnects include *communication latency* and *message throughput*. The communication latency is the time elapsed between the initiation of the message and the reception of the entire message at the destination. The message throughput is the number of the messages delivered per unit time.

3. Ethernet Topologies

In this section, we present several Ethernetconnected topologies as interconnection for massively parallel clusters. For the completeness of the paper, some materials from [17] are reiterated.

3.1 Star Topology

In a small cluster, the nodes can be interconnected using a single Ethernets switch, as shown in Figure 2 (a) where there is only one switch delay between any nodes. The switch entity can be either single switch or a group of stackable switches. A message is first forwarded from the source node to the switch and then from the switch to the destination node. Only one network interface (I_c = 1) is required at each node. The maximum number of nodes N_c in the cluster is therefore limited by the number of switch ports I_s, typically 12-36 ports.

The Ethernet switches usually offer fully connection backplane where, in absent of output port conflict, all messages can be concurrently forwarded without contention.

We can increase bandwidth of the system using additional switches and network interfaces. The cluster can be configured such that the traffic is distributed among these interfaces. As shown in Figure 2 (b), each node is equipped with two network interfaces. The traffic can be distributed by equally dividing the nodes that generate messages to each interface or by differentiating traffic into subclasses. Two important classes of traffic in the cluster are intra-node communication and NFS communication. Separate intra-node and NFS communication enhances performance of the system especially for applications with a lot of file activities [18]. The number of interfaces I_c is limited by the number of I/O slots at the node, typically 4 to 6 PCI slots.

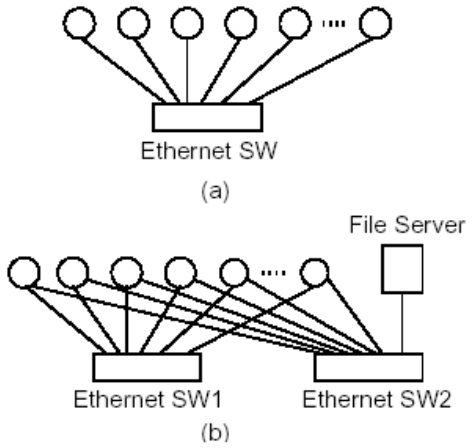

Figure 2. Cluster interconnection: (a) Simple cluster connection, (b) Separate network for NFS.

A certain type of Ethernet switches has stackable capability. The stackable switch has a backplane connection for interconnecting a group of switches together. In some technologies, the backplane bandwidth of an SSB still able to handle full connectivity. In current technology, four to seven switches can be stacked together. The number of nodes in a single SSB cluster is therefore in range of one hundred nodes.

3.2 Tree Topology

The size of the cluster can be exponentially scaled up by interconnecting switches according to the tree topology. The routing between the nodes is accomplished by switch learning. The 3-level tree network is presented in Figure 3 (a). In *M*-level tree, the nodes are connected to the (*M* - 1)-level switches. At least one port in the (*M* - 1)-level switch is used to connect upward to the root.

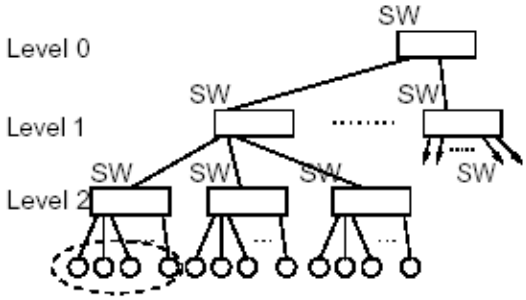

Figure 3. Tree topology.

The tree topology can provide connection up to very large size cluster. The maximum number of nodes in the system can be represented by $N_c = (I_s - 1)^{(M)} + (I_s - 1)^{(M - 1)}$ nodes which is astronomically large even for a small number of tree levels. The tree topology is not suitable for communication intensive applications since links that connected toward the root node become bottlenecks in the communication paths between nodes in the different switches. Some switch technologies allow a set of links to be bundled. Using bundle links, a cluster can be build according to fat tree topology which alleviates the bottleneck problems. However, the number of bundled links is bounded by two to four links.

3.3 Stack-Ring Topology

As previously mentioned, the scalability of both single-hop and tree topologies is limited. Based on the ring topology, we propose a *Stack-Ring topology* (SR) to build a large scale Ethernet-connected cluster. Figure 4 shows the 5-stage SR network. In the SR topology, the computing nodes responsible for not only executing applications but also forwarding messages. The SR network has S_t stages. Two network interfaces are implemented at each node labeled as East (E) interface and West (W) interface. A single SSB belongs to one stage. The SSB i ($i \in (0, \ldots, S_t - 1)$) services communication requests from the nodes at the stage i and $|i-1| \; mod \; S_t$. The dashed oval in Figure 4 show the group of nodes serviced by the SSB in the stage 1. Each stage has the maximum of N_r nodes which is equal to $\frac{I_s \times SK_n}{2}$. The node r at the stage i is labeled as (r, i) where $0 \le r \le (N_r - 1)$ and $0 \le i \le (S_t - 1)$.

The traffic is divided into two classes, intrastage traffic and inter-stage traffic. The intra-stage traffic is the communication between the nodes within the same SSB, accomplished via the hardware Ethernet switch. The inter-stage traffic is the communication between the nodes that do not have direct data link level connections. For interstage communication, the nodes are responsible for forwarding parts of messages to their destinations. The message forwarding process is performed in software. Since the software routing incurs higher overhead compared to hardware routing, the shortest path from the

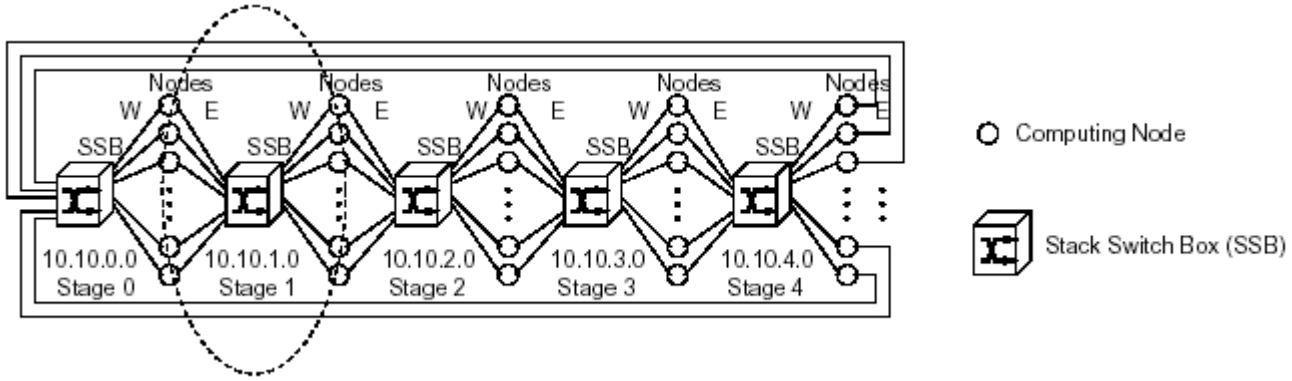

Figure 4. Stack Ring topology (SR): The topology consists of 5 stages with wrap around communication links. Each computing node has two network interface called East channel and West Channels.

source to the destination is selected using the algorithm shown in Figure 5.

The algorithm selects the interface that leads the destination with the minimum number of hops. The relative distance from the source to the destination is compared to provide routing information. For example, consider the nodes in the stage 1, Figure 4, the interface Wis selected for destinations belong to stage 0 and 4 and the interface E is selected for destinations in the stage 2.

Assuming that TCP/IP protocol is adopted in communication subsystem in the cluster, every node (x, i) is assigned an IP address to each interface. The nodes within the same stage are in the same subnetwork with IP:[10.10.*i*.*x*] for the interface A and IP:{[10.10.(i - 1).*x*], $i \neq 0$ and [10.10.(St - 1).*x*], $i = 0$} for the interface W. Hence, this IP configuration can support up to 127 nodes at each stage which covers available ports in the stack switch configurations. The routing table at each node is assigned according to information obtained from the interface selection algorithm. The load balancing of forwarding tasks is managed by the following scheme. The x^{th} node restricts the x^{th} nodes of the next/previous stage as its gateways to route messages to the nodes in other stages. Therefore the forwarding tasks are evenly distributed among all the nodes.

The table 1 shows an example of routing table assignment for the nodes in the stage 1.

Table 1. IP Forwarding table in the SR network.

Network	Interface	Gateway
10.10.0.0	W	10.10.0.x
10.10.1.0	E	N/A
10.10.2.0	E	10.10.1.x
10.10.3.0	W	10.10.0.x

3.4 Stack-Mesh Topology

The scalability of the SR topology is limited by the network diameter. The communication delay incurred in software forwarding process through the ring can degrade the performance if the message has to pass a large number of stages. Several popular topologies are classified as *orthogonal topology*. The nodes in the orthogonal networks can be arranged in the orthogonal n-dimensional space [11]. Due to their scalable properties, the orthogonal n-dimensional topologies are the basic topologies used in most contemporary multicomputers. Two important orthogonal n-dimensional topologies are the *n-dimensional mesh* and *k-ary n-cube* topologies and are defined as follows [6]:

Definition 1: An *n-dimensional mesh network* is defined as an interconnection network that has $k_0 \times k_1 \times k_2 \times \ . \ . \ . \ \times k_{n-1}$ nodes where k_i is the network radix of dimension i and n is the network dimension. The particular node is identified by the position in each dimension which can be represented by vector $(x_1, x_2, x_3, \ . \ . \ . \ , x_n)$. Two nodes, $(x_1, x_2, x_3, \ . \ . \ . \ , x_n)$ and $(y_1, y_2, y_3, \ . \ . \ . \ , y_n)$ are neighbors to each other if and only if there exists an i such that $x_i = y_i + 1$, and $x_j = y_j$ for all $i \neq j$.

Definition 2: An *k-ary n-cube network* is de- fined as an interconnection network that has n dimensions having k nodes in each dimension. The particular node in *k-ary n-cube* is identified by the position in each dimension which can be represented by vector $(x_1, x_2, x_3, \ . \ . \ . \ , x_n)$. Two nodes, $(x_1, x_2, x_3, \ . \ . \ . \ , x_n)$ and $(y_1, y_2, y_3, \ . \ . \ . \ , y_n)$ are neighbors to each other if and only if there exists an i such that $x_i = (y_i + 1) \ mod \ k$, and $x_j = y_j$ for all $i \neq j$. There are wraparound channels in the k-ary n-cube, with are not present in the ndimensional mesh networks. If $k = 2$, then every node has n neighbors. If $k > 2$, then every node has $2n$ neighbors.

Interface Selection Algorithm for SR
Inputs:
1. Address of the source node (R_s, X_s)
2. Address of the destination node (R_d, X_d).
Output:
Selected Interface
Procedure:

```
begin
    if(X_s ≤ S_t/2)
        if((X_s ≤ X_d) and ((X_s + S_t/2) > X_d)
            Select interface E;
            return;
        else
            Select interface W;
            return;
    endif
    if(X_s > S_t/2)
        if((X_s > X_d) and ((X_s − S_t/2) ≤ X_d)
            Select interface W;
            return;
        else
            Select interface E;
            return;
    endif
end
```

Figure 5. An interface selection algorithm for Stack-Ring topology.

The SM topology conforms to the *k-ary 2-cube network* since the *k-ary n-cube* network is regular and symmetric since all the nodes are identical compared to formal definition of the Mesh network. The node degree in the mesh network depends on its location. The utilization of channels in the center area of the mesh is higher than the channels near the edges. Figure 6 shows the four by four Stack-Mesh Topology (SM). The computing node has four network interfaces connected to the nearby SSBs. Interfaces are labeled as *N, E, S*, and *W*, according to their directions. The computing nodes in the systems is addressed as (*x, i, j*). *x* represents the node's rank in the group. The node group is surrounded by the dash line in the Figure 6. The location of the group of nodes in the network is specified by (*i, j*).

There are two issues need to be considered in the configuration of SM network, addressing and distribution of forwarding task. Assuming that the cluster adopts IP protocol in the message passing communication, the IP subnetwork is assigned to each node group. There are several approaches in assigning the IP addresses to SM network. We selects a simple method to ease of configuration. The node (*x, i, j*) is assigned the IP address of [10, *j, i, x*]. Using this IP addressing scheme, the SM network can scale up to 256×256 mesh.

The forwarding table can be filled up using the algorithm shown in Figure 7. The shortest paths from the source to the destinations are selected. The message follows the dimension-order path in which the *X* dimension is traversed first and followed by the *Y* dimension. Similar to SR network, the forwarding tasks are distributed among the node. The gateways for the node [*x, x, x, i*] are configured according to the scheme in the Figure 7 with the IP address of [*y, y, y, i*] . The number of entries in the forwarding table is equal to the product of *N* and *M*.

4. Performance Comparisons

Both hardware switches and software forwarding are adopted in both SR and SM networks. To have better understanding on the overhead incurred in software forwarding, we have evaluated several communication subsystem performance tests on different network configurations using Netpipe. The maximum cluster size based on current technology is estimated for each topology. The HPL benchmark were tested and their results are discussed in the last part of the performance comparisons.

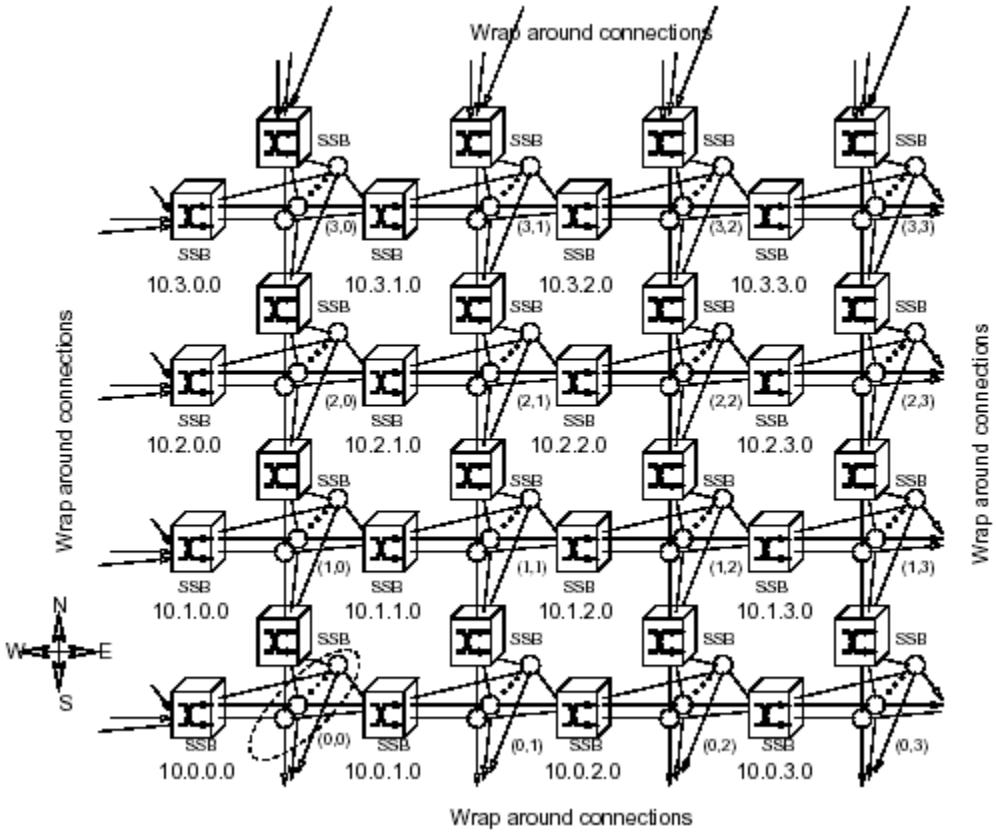

Figure 6. 4-by-4 Stack-Mesh topology for large scale clusters.

4.1 Software Forwarding Overheads

The network performance of several network configurations was measured using NetPIPE [19]. A set of Pentium-II 400MHz with 384MB of memory running LINUX Kernel version 2.2.9-27mdk and the Intel EtherExpress 520T switches were used in the experiments. The MPI routines were called to send/receive messages between a pair of nodes.

The differences in the transfer time of hardware-based and software-based are reflected in Figure 8. For the block size of 1 Kilo-bit, the transfer delays for direct connection, hardware Ethernet switch, software forwarding are measured as 146 μs, 151 μs, 208 μs, respectively. The hardware switching is 1.37 time faster than the software forwarding. In the Ethernet throughput graph, Figure 9 (a), the throughput of the hardware switch are very close to the direct connection (cross wire). The performance of two switch in series is slightly lower than the single switch. This is because the switching delay is small compared to the software overhead incurred during packet initiation and reception.

The throughput of software forwarding is 10-20% less than the hardware switch for the medium block sizes. The throughput of the communication further decrease when two-pairs of nodes send messages simultaneously. Similar trend is observed in the Ethernet signature graph, in Figure 9 (b). This is because, in software forwarding, the message is passing through two switches and one software forwarding. The message is first sent from the node to the switch and then is to the gateway node. The gateway node performs forwarding functions and then sends message to the second switch which is in turn forwarding the message to the destination node. From the results of the experiment, communication performance degrades considerably using software forwarding. The intergroup communication in both SR and SM networks should be minimized.

4.2 Scalability

We studied the scalable performance of Ethernet technology by estimating the maximum cluster size for different topologies. The 24-port fast Ethernet switch is considered as the network device building block. We make an assumption that the maximum number of switches in the same stack is equal to five and at most four network interfaces can be installed at each node. The maximum sizes of the star, tree, SR, SM topologies are estimated as follow.

Interface Selection Algorithm for $N \times M$ network
Inputs:
1. Address of the source node (R_s, X_s, Y_s)
2. Address of the destination node (R_d, X_d, Y_d)
Output:
Selected Interface
Procedure:

```
begin
    // Step 1: Forward in X dimension;
    if(X_s < X_d)
        if((X_d - X_s) < N/2)
            Select interface E;
            return;
        else
            Select interface W;
            return;
    if(X_s > X_d)
        if((X_s - X_d) < N/2)
            Select interface W;
            return;
        else
            Select interface E;
            return;
    // Step 2: Forward in Y dimension;
    if(Y_s < Y_d)
        if((Y_d - Y_s) < N/2)
            Select interface N;
            return;
        else
            Select interface S;
    if(Y_s > Y_d)
        if((Y_s - Y_d) < N/2)
            Select interface N;
            return;
        else
            Select interface S;
            return;
end
```

Figure 7. An interface selection algorithm for Stack-Mesh topology.

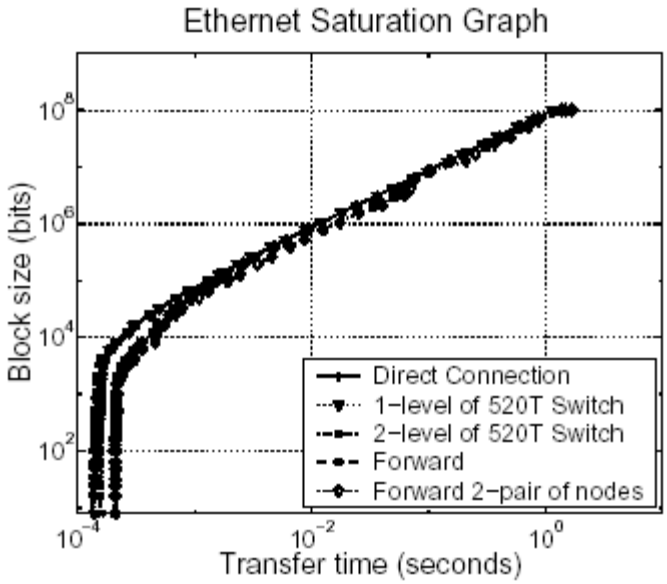

Figure 8. Network performance comparison: Block size versus transfer time.

Table 2. Summarize the scalability of the proposed Ethernet topologies for massively parallel cluster.

Topology	# of Switches	# of Nodes	Software Forwarding	Bisection Bandwidth
Star	5	120	none	1.2 Gbps
Tree	61	1416	none	1.2 Gbps
SR	25	300	1	12 Gbps
SM	160	960	1	48 Gbps

For the star topology, the maximum number of nodes connected to the cluster is equal to the product of the number of port I_s (24 ports) and the number of switch in the stack (5), 120 (24×5) nodes in our case. The bisection bandwidth BW can be represented as $min(I_s \ / \ 2 * b, SP) \times \#ofNI_s$. Under parameters considered, the value of $I_s / 2 * b$ is 1.2 Gbps.

For the two-level tree topology, the connection between the root node and the level one comprises of two bundled links to increase bandwidth to 200Mbps. The interconnection network consists of a single switch as the root and sixty switches in the level 1 (twelve groups of five switches). The total of 1416 nodes can be connected to the cluster as leave nodes with $min(1.2Gbps, SP)$ bisection bandwidth. Although a large number of nodes can be connected to the tree topology, available bisection bandwidth BW located at the root switch is not well balanced with the number of nodes.

A 5-stage SR network consists of 25 switches (5 stackable switches at each stage) and the total of 300 nodes (60 nodes per stages). The peak bisection bandwidth is 12Gbps ($60 \times 2 \times 100M$). The network diameter D is 4 hops. A message has at most one forward operation between any pair of source and destination. While the communication delay in the SR topology is higher due to software forwarding, the aggregated bisection bandwidth of the proposed topology significantly higher by a factor of 10 and communication locality of intra-stage nodes can benefit from the directly connected hardware switch. We believe that as performance of microprocessor continuing to increase and the adoption of a fast processor in the network interface, the software forwarding overhead will decrease in the near future.

The 4×4 SM networks consists of 32 SSBs which are comprised of 160 switches. The number of computing nodes in a single group is equal to 60 nodes therefore the total number of nodes is $60 \times 16 = 960$ nodes. The peak bisection bandwidth is 48Gbps. Each group of nodes has 6Gbps bandwidth. The network has 4 groups in one dimension. Therefore the total of $6 \times 4 = 24$Gbps bisection bandwidth. With the wrap around channel, the bisection bandwidth is doubled to 48 Gbps At most one forward operation is required between any pair of source and destination. The scalability results are summarized in Table 2.

4.3 HPL Benchmark Experiments

To investigate the validity of the proposed topologies, HPL-A portable implementation of the High Performance Linpack

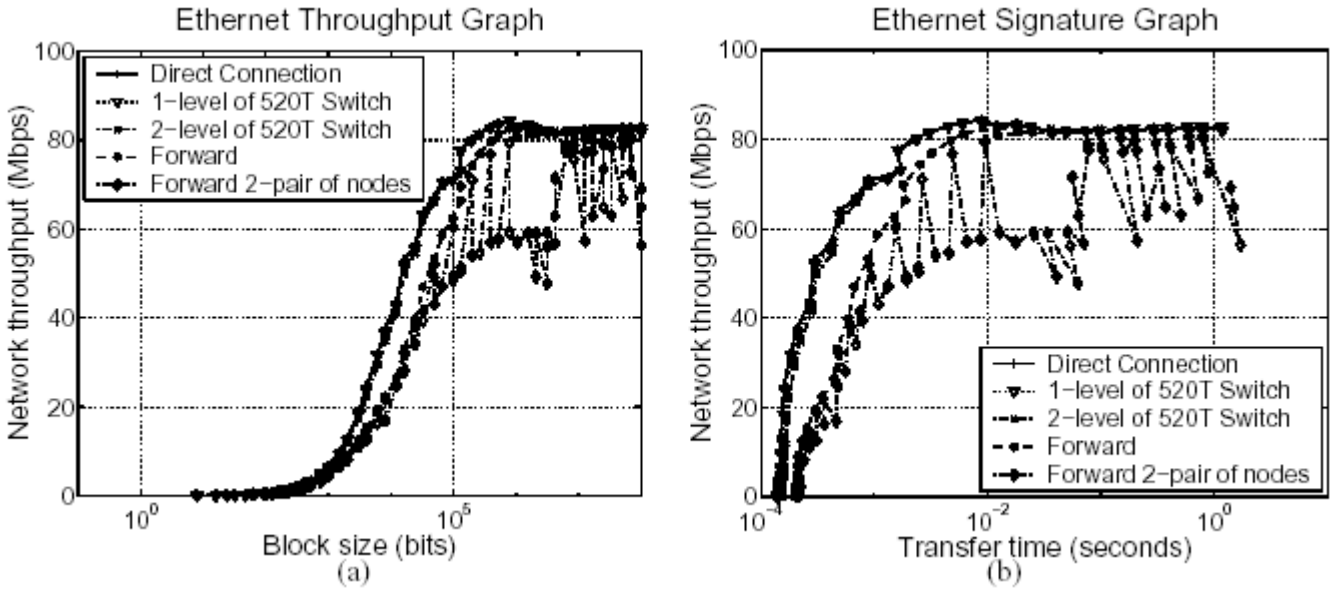

Figure 9. Network performance comparison (a) Network throughput versus message block size, (b) Network throughput versus transfer time. The test results show the performance of the hardware switching is 1.37 times and the software forwarding. The software forwarding shows higher communication overhead at the same block size.

benchmark for distributed- Memory computers was tested on different network topologies. Due to limited resources, the clusters consists only eight nodes. The same computer configurations were used. The HPL results cannot reflect the scalability of the proposed system since the number of nodes is small but we test the HPL to show the implementation of the proposed topology. The following parameters are selected: problem size = 10000, Block size $NB = 64$, and $P \times Q = 1 \times 8$. The performance results are shown in Table 3.

Table 3. HPL performance results.

Topology	Times	Gflops
Star	436.78	1.527
Tree	438.09	1.522
SR	5785.75	0.1153
SR with extra node	455.14	1.465

In the star topology, eight nodes are connected to a single Ethernet switch. The performance of the start topology is the best for the eight-node system. For the tree topology, two sets of four nodes are connected to two level-1 switches. The level-1 switches have a single link connected to the level-0 switches. The SR topology consists of four stages. Each stage consists of two computing nodes. The performance of the SR topology is very poor. The HPL benchmark has considerable communication activities. The nodes in the SR topology have to process the forwarding tasks.

The degradation in performance results from the forwarding overhead and the context switch overhead. However, the HPL results for the 8- node system is not a fair comparison. The SR topology is designed for the system that the star topology cannot accommodate. In the larger system, the intra-stage traffic can communicate through hardware. The number of nodes at each stage is equal to 60 which can perform a certain amount of task. The larger number of nodes means better distribution of forwarding task of interstage traffic. Also the large problems are usually divided into multiple levels of hierarchies. The communication between subproblem is less.

To relieve the forwarding overhead, the extra node is added to each stage to perform the forwarding task or act as a gateway. The system consists of 12 nodes, 8 nodes for computation and 4 nodes for forwarding. The performance is improved by an order of magnitude. As the network interface technology and microprocessor technology continues to evolve, the software forwarding overhead will decrease and slowly migrate to hardware level.

5. Concluding Remarks

The Ethernet network topologies for large scale clusters were studied. The contemporary Ethernet switches can be stacked together and have large backplane bandwidth. We proposed the SR and SM topologies for medium to large size clusters

implemented with stackable Ethernet switches. In SR network, the groups of stack switches are connected in series with wraparound links. The mesh network with wraparound links is used as a basis of the SM topology.

The software forwarding overheads were measured and compared to the hardware overhead. The results show that the hardware switch is faster than the software forwarding (adopted in the proposed topology) by a factor of 1.37. The aggregate bisection bandwidth of the SR network is more than other topologies by a factor of ten. The forwarding tasks are distributed among all nodes. The SR/SM networks of 300/960 nodes with 12/48Gbps bisection bandwidth are shown. The realization of the SR networks was tested using HPL benchmark. For the small cluster, the results from HPL benchmarks in SR network is very poor. However, we believe that applications run in a large cluster, the subproblems are classified into several levels. The communication intensive parts of the task are allocated to the same stage. Therefore the interstage traffic can be reduced. The proposed approach is a promising interconnect solution for a cluster-based supercomputer not only as the main interconnection but also the back-up network of the high speed network for better system availability.

References

[1] T. Agerwala, J. L. Martin, J. H. Mirza, D. C. Sadler, D.M. Dias, and M. Snir, "SP2 system architecture," *IBM Systems Journal*, vol. 34, no. 2, pp. 152–184, 1995.

[2] J. Dongarra, G. Geist, R. Manchek, and V. Sunderam, "Integrated PVM framework supports heterogeneous network computing," *Computers in Physics*, April 1993.

[3] M. Litzkow, M. Livny, and M. W. Mutka, "Condor–a hunter of idel workstations," in *Proceedings of the Eighth International Conference of Distributed Computing Systems*, pp. 104–111, 1988.

[4] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawak, and C. V. Packer, "Beowulf: A parallel workstation for scientific computation," in *International Conference on Parallel Processing*, 1995.

[5] I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, 1999.

[6] P. Mohapatra, "Wormhole routing techniques for directly connected multicomputer systems," *To appear in the ACM Computing Survey*.

[7] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, and C. P. Thacker, "Autonet: A high-speed self-configuring local area network using point-to-point links," tech. rep., SRC Research Report 59, 1990.

[8] C. B. Stunkel, D. G. Shea, and B. Abali, *et al*, "The SP2 high-performance switch," *IBM System Journal*, vol. 34, pp. 185–204, February 1995.

[9] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawlk, C. L. S. ad J. N. Seizovic, and W. Su, "Myrinet: A Gigabit-per-second local area network," *IEEE Micro*, vol. 15, pp. 29–36, February 1994.

[10] R. W. Horst, "TNet: A reliable system area network," *IEEE Micro*, vol. 15, pp. 37–45, February 1995.

[11] J. Duato, S. Yalmachili, and L. M. Ni, *Interconnection networks: An Engineering Approach*. Los Alamitos, California: IEEE Computer Society, 1997.

[12] V. Karamcheti and A. A. Chien, "Software overhead in messaging layers: Where does the time go?," in *Proceedings of ASPLOS-VI*, (San Jose, California), October 1994.

[13] T. Sterling, "How to build a beowulf: Assembling, programming, and using a clustered pc do-it-yourself supercomputer," in *Supercomputing*, (San Jose), 1997.

[14] T. W. Giogis, "29 switching hub save the bandwidth," *BYTE*, pp. 162–169, July 1995. [15] C. Reschke, T. Sterling, and D. Ridge, "A design study of alternative network topologies for the beowulf parallel workstation," in *High Performance and Distributed Computing*, 1996.

[16] "MPI: Message passing interface forum." Message passing interface forum, http://www.mpi-forum.org, 1994.

[17] V. Varavithya and T. Thepuatrakul, "High bandwidth ethernet topology for large scale clusters," in *EECON 23*, CMU, December 2000.

[18] V. Varavithya, P. Lousangfa, and C. Ngamphiw, "Effects of network configurations on performance of beowulf-class clusters," in *EECON 23*, CMU, December 2000.

[19] Q. O. Snell, A. R. Mikler, and J. L. Gustafson, "Netpipe: A network protocol independent performance evaluator," in *IASTED Conference*, (http://www.scl.ameslab.gov/netpipe/).