Analysis of 100Mb/s Ethernet for the Whitney Commodity Computing Testbed[1]

Samuel A. Fineberg and Kevin T. Pedretti

NAS Technical Report NAS-97-025
October 1997

MRJ, Inc.
Numerical Aerospace Simulation Division
NASA Ames Research Center, M/S 258-6
Moffett Field, CA 94035-1000

Abstract

We evaluate the performance of a Fast Ethernet network configured with a single large switch, a single hub, and a 4x4 2D torus topology in a testbed cluster of "commodity" Pentium Pro PCs. We also evaluated a mixed network composed of ethernet hubs and switches. An MPI collective communication benchmark, and the NAS Parallel Benchmarks version 2.2 (NPB2) show that the torus network performs best for all sizes that we were able to test (up to 16 nodes). For larger networks the ethernet switch outperforms the hub, though its performance is far less than peak. The hub/switch combination tests indicate that the NAS parallel benchmarks are relatively insensitive to hub densities of less than 7 nodes per hub.

1. Work performed under NASA Contract NAS 2-14303

1.0 Introduction

Recent advances in “commodity” computer technology have brought the performance of personal computers close to that of workstations. In addition, advances in “off-the-shelf” networking technology have made it possible to design a parallel system made purely of commodity components, at a fraction of the cost of MPP or workstation components. The Whitney project, being performed at NASA Ames Research Center, integrates these components in order to provide a cost effective parallel testbed.

One of the key components of Whitney is the means of interconnecting the nodes (each of which is an off the shelf PC). There are many custom, semi-custom, and commodity technologies available for networking. These include Ethernet, Fast Ethernet, Gigabit Ethernet, Myrinet, HiPPI, FDDI, SCI, etc. The most attractive of these choices, however, is currently Fast Ethernet, due to its good performance and extremely low cost.

Combining a large number of systems into a high performance parallel computer requires the careful selection of both network technology and topology. The Whitney project is currently evaluating different network technologies and topologies in a testbed cluster of “commodity” Intel Pentium Pro PCs. This paper will report on the implementation and performance of Fast Ethernet, in a single hub and a single switch, a combination of hubs and switches, and in a 4x4 routed 2D torus[2] topology.

The remainder of this paper is organized as follows. Section 2 will provide the configuration details for the networks we tested. In section 3, the actual hardware configuration of the testbed system will be discussed. Section 4 presents the results of the hub, switch, and torus experiments. Section 5 describes the effect of using hubs to connect multiple nodes to each switch port. Finally, section 5 presents final conclusions along with directions for further research.

2.0 Network Configuration

Fast Ethernet [Iee95] is a ten times faster version of the original Ethernet standard. The increase of the bit rate to 100 million bits per second (Mbp/s) and modifications to the physical layer of the Ethernet standard are the only major changes. This has greatly helped manufacturers in bringing products to market quickly and also has created a large consumer market because of Ethernet's familiarity. As a result, the price of Fast Ethernet equipment has fallen dramatically since its introduction. A typical PCI Fast Ethernet adapter costs $50-$80, and hubs cost approximately $75 per port. In addition, because the most common physical layer for Fast Ethernet (i.e., 100baseTX) utilizes

2. For the purposes of this paper, the 4x4 routed 2D torus tested will often simply be referred to as a torus.

inexpensive cabling technology, category 5 unshielded twisted pair (UTP), wiring costs are also very low.

2.1 Connection Options

To build a Fast Ethernet network, machines must be attached using either a hub, switch, or “crossover” cable. In a hub, all systems share a single broadcast network, so only one host can send and one or more hosts may receive at one time. When more than one host attempts to use the network at the same time, a “collision” occurs. The systems then retry their messages using a “carrier sense media access with collision detection” (CSMA/CD) algorithm with exponential backoff. This mechanism for handling shared network access is common to all Ethernet based systems. This means that in a hub connected system the maximum bisection bandwidth is limited to 100Mbp/s (12.5 MBytes/sec), and is often lower when more than one host is contending for access, regardless of the number of nodes in the network. While this is hardly adequate for a parallel system, we performed measurements on this configuration to see how it would perform.

To increase the bisection bandwidth of the system, one must increase the number of simultaneous connections possible and “break” the ethernet into multiple segments. This can be done either with an Ethernet switch or by adding TCP/IP routers. The advantage of Ethernet switching is that there still appears to be a single Ethernet network, though it will now support multiple simultaneous senders and receivers. In addition, some Ethernet switches allow nodes to operate in “full duplex” mode where they simultaneously send and receive data. This is especially useful for acknowledgment and flow control packets that must flow from a receiver to a sender. The disadvantage, however, is that Ethernet switches are expensive, $300-$700 per port, and they do not scale past 100-200 nodes. Further, switches do have a limited bisection bandwidth, though they can typically deliver 1-2Gbp/s of aggregate bandwidth.

A second choice, however, is to utilize TCP/IP based routing where either some or all nodes forward packets between subnets. This scheme increases the aggregate bandwidth of the network without purchasing additional switching hardware (the nodes are the switches). In addition, if nodes are attached directly using “crossover” cables rather than hubs, full duplex operation is possible. However, router nodes must have more than one Ethernet card, nodes must spend CPU time forwarding packets between other nodes, and the performance of TCP/IP routing is usually lower than that of Ethernet switches.

In this paper we chose to test a hub connected system, a switch connected system, a routed topology, and a combination of hubs and switches. While these networks are all viable for the small system we tested, no single hub or switch can scale to 500+ nodes. Therefore, these networks are meant to be used for comparison. The routed topology and hub/switch combination networks, however, will scale to

500+ nodes, so some combination of these is likely to be used in the final Whitney system.

The routed topology we chose, a 2D torus, requires all nodes to perform routing. Further, because links are implemented with crossover cables (i.e., the network does not include any hubs), all connections can operate in full duplex mode.

The 2D torus was chosen for two reasons. The first reason was scalability, a mesh or torus network can be expanded to any size system by increasing either one or more dimensions. This is particularly important because the planned size for Whitney is 400-500 nodes. In addition, by increasing both dimensions not only is the size of the mesh increased, but also the bisection bandwidth. The only limitation is that as the size increases, so does the diameter of the network. We chose to minimize this effect by keeping the mesh square and providing the wraparound connections.

The second reason for choosing a 2D torus was for physical and cost reasons. The nodes we used in the experiments had only 5 PCI slots. Utilizing single port Ethernet cards, this means that no more than 5 other systems may be attached to each node. While there are two and 4 port Ethernet cards, the per port cost is 2-4 times the cost of single port cards. Because we wanted an arbitrarily scalable network, we could not use a hypercube (we could only have up to 2^5, 32, nodes), and we would need 6 links for a 3D mesh/torus.

2.2 Torus Network

Figure 1 illustrates the 2D torus configuration. Each of the sixteen nodes was

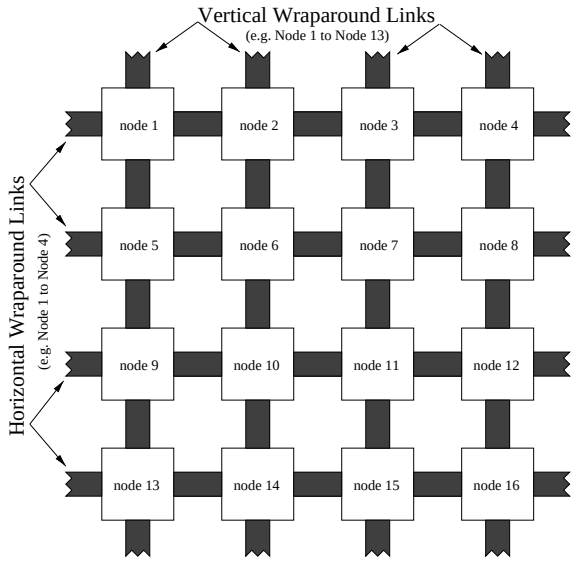


FIGURE 1. ***16 node 2D torus network configuration***

directly linked to its four nearest-neighbors via a 100 Mbs bidirectional Fast

Ethernet connection. Thus, the torus was partitioned into thirty-two distinct TCP/IP subnetworks. The size of the torus was restricted to 16 nodes because we did not have enough ethernet cards to build a larger torus. Even at this size, the torus network required 64 Ethernet cards.

Links between neighbor interfaces (for the torus configuration) used standard category 5 unshielded twisted pair wiring that was crossed over (null modem). The wiring was tested and certified for 100 Mbs operation to ensure good connections. All links were direct so no dedicated hubs, routers, repeaters, switches, or other devices were used in the torus.

In addition to the topology depicted in Figure 1, an additional node was connected to a fifth network interface in node 1. It's major functions were to serve as a front-end for starting jobs on the cluster and to work as an NFS server for the processing nodes. Shared disk I/O, while important in a production system, was not a significant factor in any of the benchmarks which were used in this paper. The final Whitney system will have a parallel file system implemented across multiple I/O nodes.

2.3 Hub

For the hub experiments, all nodes were attached to three “stacked” Bay Networks Netgear FE 516 Hubs. By stacking the three 16 port hubs they act like a single 48-port hub. Each node had only a single Ethernet card and all nodes plus the front end were on a single TCP/IP subnet.

2.4 Switch

For the switch based experiments we used a single Cisco Catalyst 5500 switch with 48 ports. The switch was operated at 100Mbp/s in “full duplex” mode for the switch only experiments, and all nodes were attached directly to the switch including the front end system. The switch operated as a single TCP/IP subnet, but because no links were shared there was no possibility for ethernet collisions.

2.5 Hub/switch combination

To reduce the cost and increase scalability of a switch based ethernet network, we attach multiple nodes to each switch port. This can be done by attaching several nodes to an ethernet hub and attaching an “uplink” connection (i.e., a cross-over cable) from the hub to a switch port. This means that we can substantially reduce the number of switch ports needed (i.e., 2 nodes per hub means we only need 18 switch ports for a 36 node system, 3 per hub requires 12 ports, etc.) In addition, because hubs are so much cheaper than switches, even with the uplink ports the overall cost of a hub/switch based network should be lower. Of course, the disadvantage is that the overall aggregate bandwidth will likely be reduced and nodes can only operate in “half duplex” mode.

To test this configuration we had 13 4-port Linksys EtherFast 100BaseTX hubs. Each of these could be stacked to form an 8-port hub. We also used the NetGear hubs for the experiments where we needed more than six 8-port hubs (i.e., for 4 nodes/hub and 5 nodes/hub). The front end node was directly attached to the ethernet switch. Therefore, we were able to test configurations with 1-7 compute nodes attached to each hub, with each hub having one uplink to the Cisco switch as shown in Figure 2.

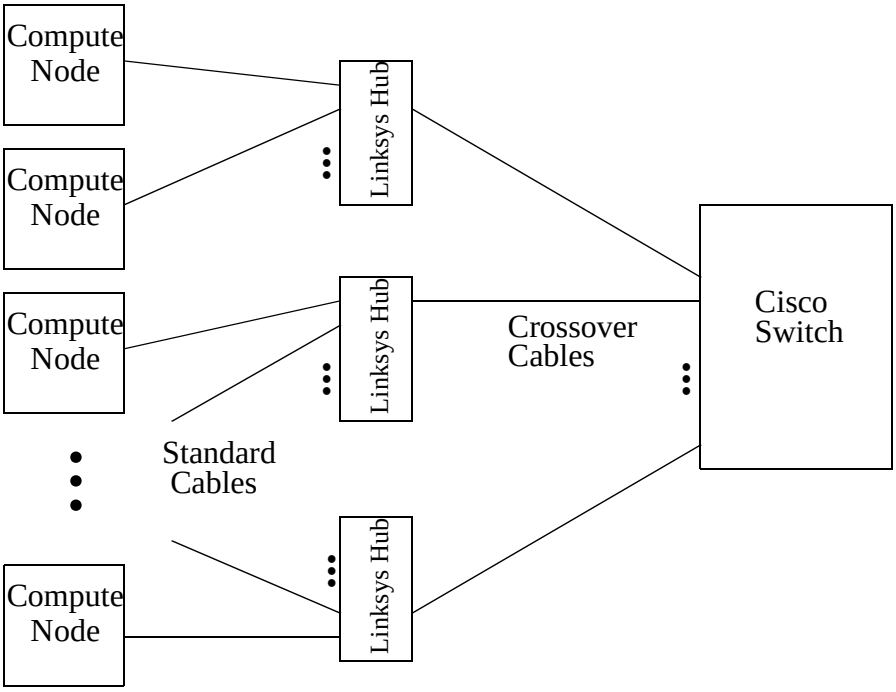


FIGURE 2. ***Hub to switch network configuration***

The performance of this network configuration should be better than the Hub based system, but worse than a pure switch based network. The performance of this network will be discussed in Section 5.

3.0 The Whitney Prototype

3.1 Hardware

The Whitney prototype consists of 39 compute nodes (though only 36 were used in these experiments) and one front end node. The compute nodes consist of the following hardware:

- Intel Pentium Pro 200MHz/256K cache
- ASUS P/I -P65UP5 motherboard, Natoma Chipset
- ASUS P6ND CPU board
- 128 MB 60ns DRAM memory

- 2.5 GB Western Digital AC2250 hard drive
- 1 or 4 Cogent/Adaptec ANA-6911/TX ethernet cards[3]
- Trident ISA graphics card (used for diagnostic purposes only)

For this paper, we chose to concentrate on Fast Ethernet based networks, subsequent research will evaluate the cost/performance trade-offs of more esoteric networks such as Myrinet.

3.2 Software

Red Hat Linux 4.1[4] (RHL) was installed on each of the processing nodes. The kernel included with RHL, version 2.0.27, was replaced with the newest version at the time - 2.0.30. The kernel was compiled with ip forwarding turned on so that the routing mechanism of Linux could be used. Both the de4x5 v0.5 and the tulip v0.76 Ethernet drivers were tested. The de4x5 driver was used initially and exhibited some inconsistent performance characteristics. The final torus configuration on which all benchmarks were run used the tulip development driver.

For the torus network, a program executed at boot-time set up the routing tables on each node with static routes to non-local subnets using an X-Y routing scheme. Packets addressed to non-neighbor nodes were forwarded through the appropriate interface towards their destination. The shortest-hop distance was maintained in all cases. For the switch, hub, and hybrid networks only a single TCP/IP subnet was utilized, so no TCP/IP routing was needed within the system. Instead, all routing is done within the switches at the physical layer.

The MPI message passing system [Mes94] was used for communication between nodes. MPICH (version 1.1.0) [GrL96] was the specific MPI implementation utilized. It was built using the P4 device layer, so all communication was performed on top of TCP sockets. Programs were started on the system by the *mpirun* program [Fin95] which resided on the front-end. *mpirun* takes the name of the program and the number of processing nodes to use and then remotely spawns the appropriate processes on the cluster. All of the benchmarks mentioned in this report used MPI for communication.

4.0 Performance

The first benchmark run on the cluster measured the message latency and bandwidth of point to point links. The second benchmark measured the performance

---

3. Only one network card was installed in each node for the hub and switch based tests. The torus required four Ethernet cards per node. In addition, for the torus node 1 contained an additional fifth ethernet card. The additional card was connected to the front-end node.

4. Red Hat Linux is available from http://www.redhat.com.

of collective communication. Finally, the NAS Parallel Benchmarks version 2.2 were run. These are a set of benchmarks that approximate the MPI performance of a parallel architecture on “real world” tasks (i.e., CFD codes).

4.1 Point to point message passing

To measure point-to-point message passing performance, a MPI ping-pong benchmark was utilized. This benchmark simply sent a message of a fixed size from one node to another than back. The time for this operation was then divided by two to get the time to send a message one way. The message size was varied from 1 byte to 1 Mbyte, and all experiments were repeated 20 times. Figure 3

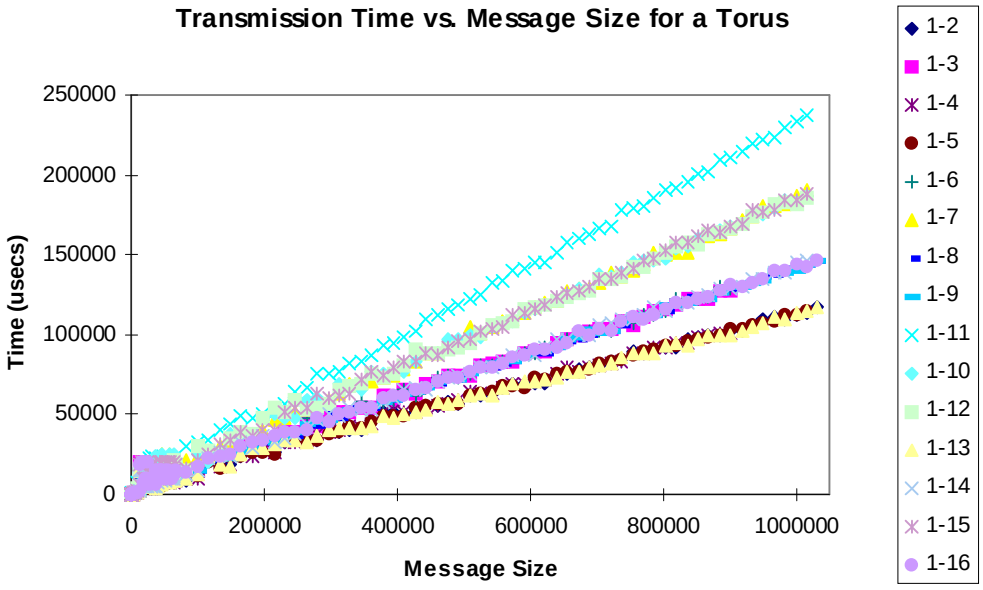


FIGURE 3. ***Point to point torus message passing time from node 1 to N***

illustrates the point to point message send/receive time from node 1 to each of the other nodes in the torus configuration. As can be seen from this graph, the message passing performance delineates itself in to 4 categories. These 4 categories represent the number of hops each node is from node 1. Therefore, the lowest transmission time is from node 1 to its adjacent neighbors, 2, 4, 5, and 13. The second category is nodes that must be communicated to through node 1’s neighbors (they are 2 hops away), i.e., 3, 6, 8, 9, 14, and 16. The third category are nodes 3 hops away, i.e., 7, 10, 12, and 15, and the final category is nodes 4 hops away for which there is only one, node 11. Similar performance curves can be generated for any other node pair, with similar results based on the nodes distance.

Figure 4 depicts the message passing time for the hub and switch configurations. In this case, the message passing time is only shown for one pair of nodes because the performance is roughly equal between different pairs. From this

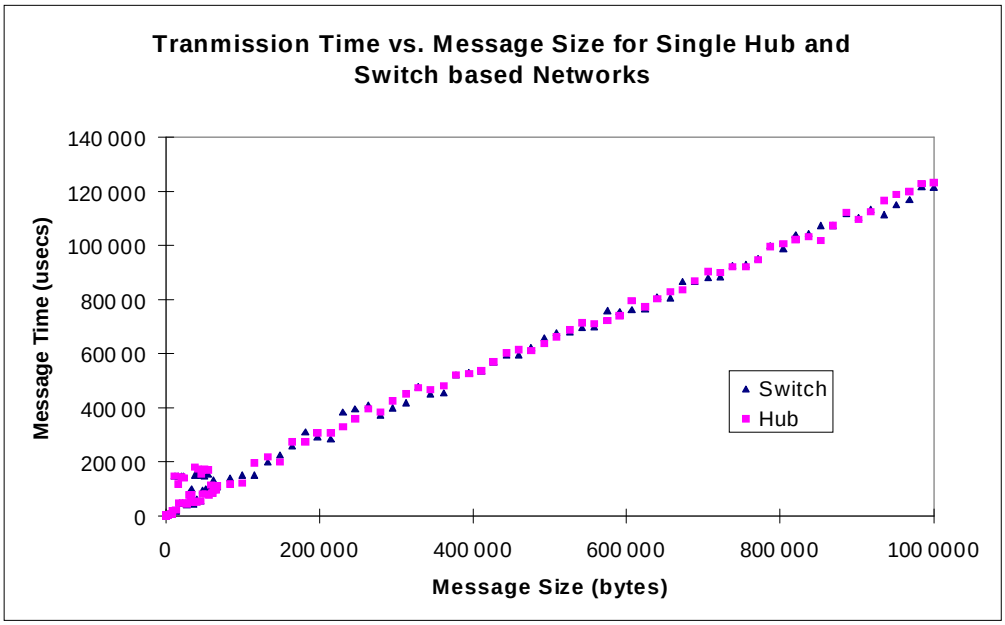


FIGURE 4. ***Point to point switch and hub message passing time***

graph the point to point performance of both a hub and switch appear to be the same. However, the next section will show that there are some slight differences. The point to point performance of the hub and switch are also similar to the nearest neighbors in a torus (see Figure 3).

4.1.1 Latency

To determine the latency of message passing Figures 5 and 6 depict the message passing time for small messages. As can be seen from these graphs, latency for a single hop on the torus or for the hub are about 175 µsecs. Then, each hop on the torus adds about 40 µsecs, so the latency for 2 hops is 215 µsecs, 3 hops is 255 µsecs, and 4 hops is 295 µsecs.

For the ethernet switch, the latency was virtually identical between different pairs of nodes, i.e., it varied within the margin of error for measurement. Figure 6 only shows the latency between a single node pair. The latency was slightly higher than the hub (190 µsec). This is likely due to the routing required within the ethernet switch fabric.

4.1.2 Bandwidth

MPI Bandwidth vs. message size is shown in Figures 7 and 8. As can be seen from these graphs, Ethernet bandwidth is quite erratic. However, some patterns can be seen. As expected, bandwidth for small message sizes is low, building to a sustained bandwidth of approximately 8-8.5 MB/sec for one hop on the torus or on the hub. For nodes more than one hop away on the torus, the bandwidth drops about 1.5 MB/sec per hop (8.5 MB/sec, 7 MB/sec, 5.5 MB/sec, 4 MB/sec). Also,

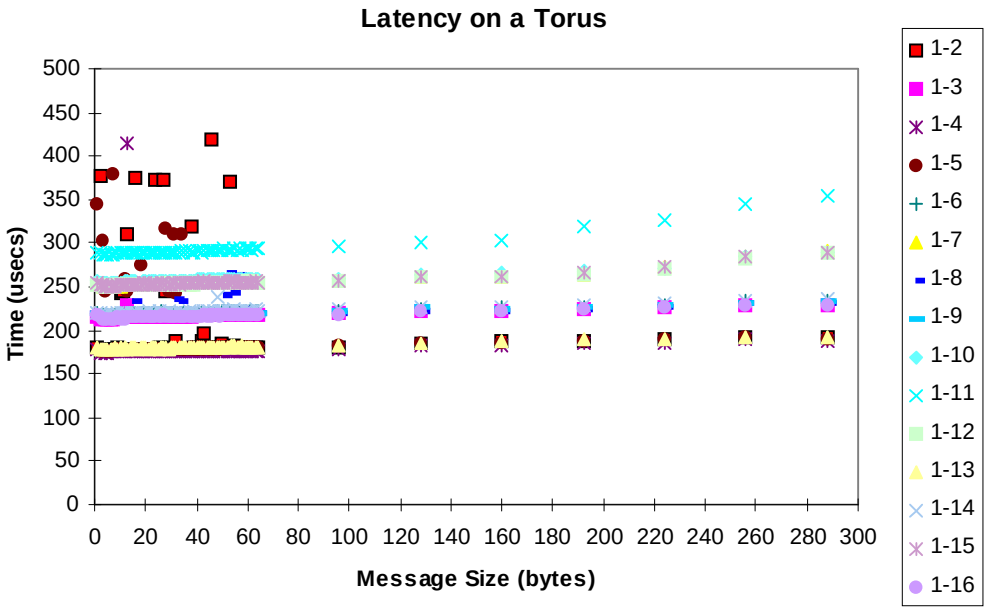


FIGURE 5. ***Transmission time for small messages on a torus***

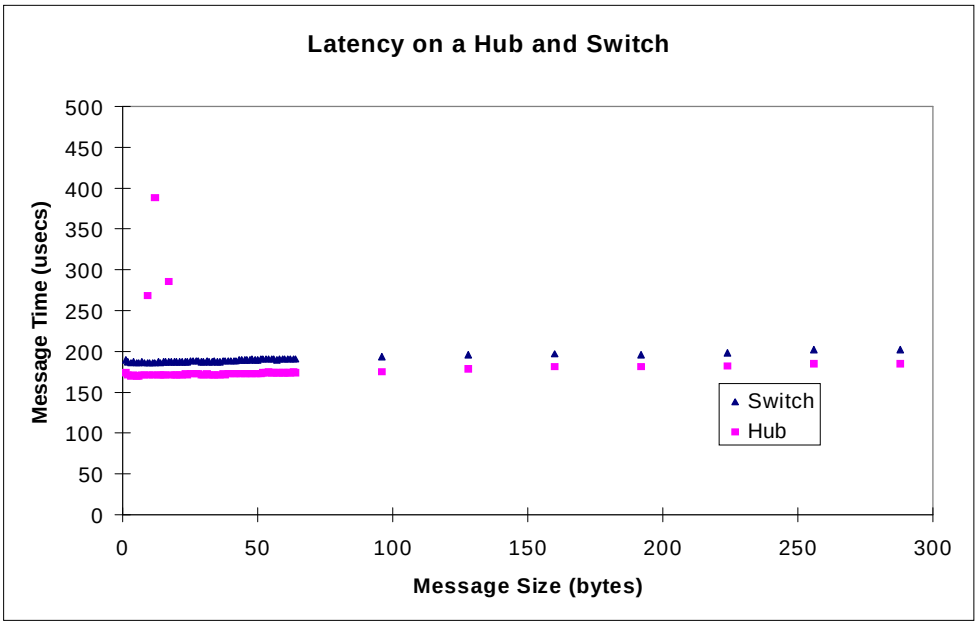


FIGURE 6. ***Transmission time for small messages on a hub***

note that the bandwidth reaches peak performance at an 8K message size, then it drops down and starts to build to peak slowly as message size approaches 1 MB. This anomaly is likely due to either the Ethernet or TCP packet size.

The performance of the ethernet switch was similar to the hub. There was some variation between ethernet switch ports, i.e., it varied from 7.8 MB/sec to 8.4

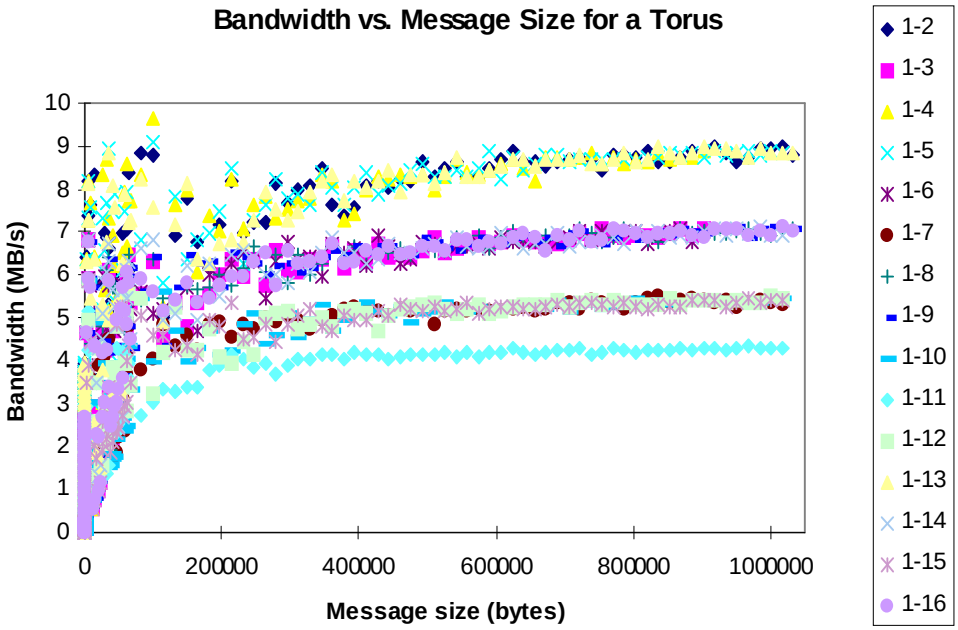


FIGURE 7. ***Bandwidth performance of the torus topology***

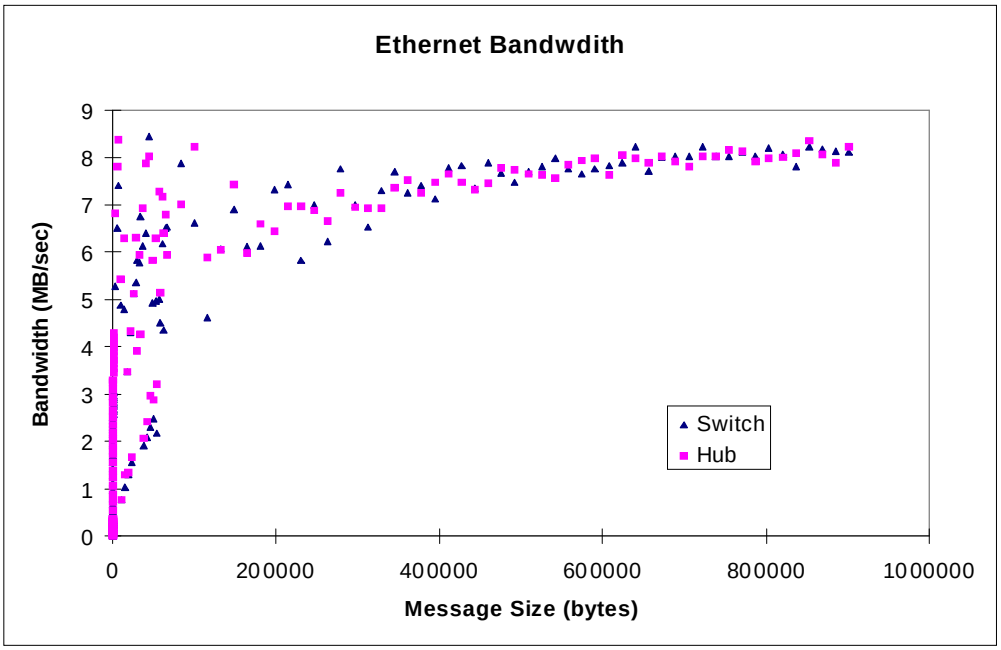


FIGURE 8. ***Bandwidth performance on a Hub***

MB/sec. Figure 8 only shows the bandwidth between a single pair of nodes. We did not attempt to find a pattern in the bandwidth differences because they were very small in comparison to the measurement error.

4.2 Collective Communication

To measure the performance of collective communication, a MPI broadcast benchmark was utilized. The benchmark measured the time required to broadcast a message to a given set of nodes and perform a MPI barrier synchronization. Message sizes used for the broadcast were varied between 1 and 32768 bytes in 2^n steps. Each message size was broadcast 20 times.

4.2.1 Bandwidth

The aggregate bandwidth of the Torus for collective communication is depicted in Figure 9. Our experiments show that for message sizes below 1024 bytes the

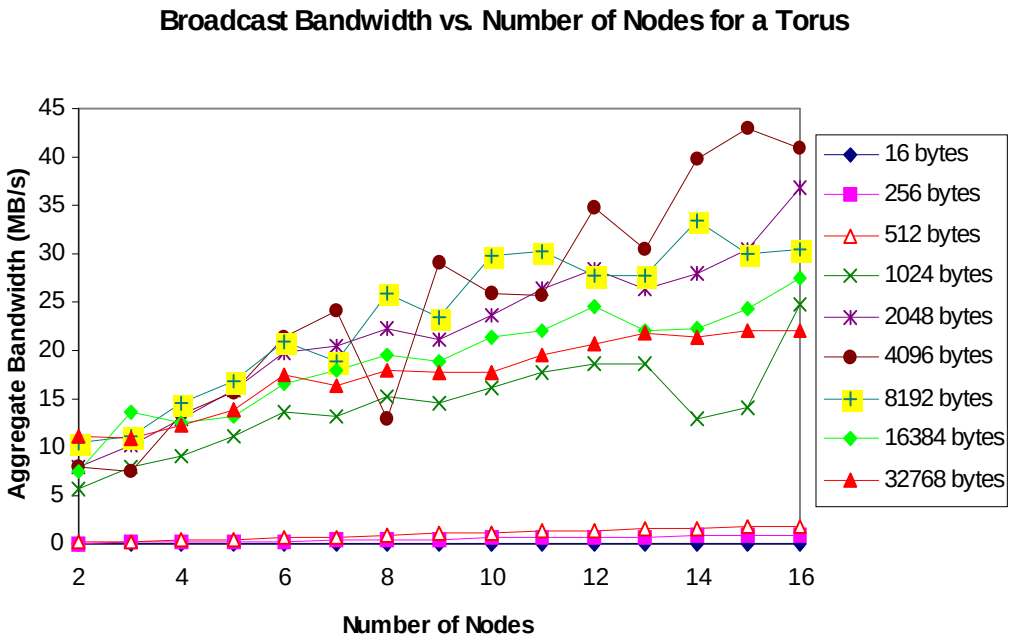


FIGURE 9. ***Collective communication performance on a torus***

aggregate bandwidth is very poor. Both the Ethernet frame size and the TCP packet size could be possible causes for this. Above the 1024 byte threshold, performance becomes much closer to expected levels. The maximum aggregate bandwidth was observed to be about 43 MB/s for the 4096 byte message size. While the theoretical maximum aggregate bandwidth of the torus should be 400 MB/s, this does almost reach the maximum bisection bandwidth (50MB/s). Further, it is quite good given the cost of software routing, TCP/IP overhead, etc. In general, the aggregate bandwidth increases as the number of nodes increases for a given message size. The exceptions are probably due to inconsistencies in routing latency and network contention.

The collective communication bandwidth for the hub is shown in Figure 10. The cut-off for good performance is still at about 1024 bytes, however performance isn't as poor below this size as was seen in the torus. Aggregate bandwidth

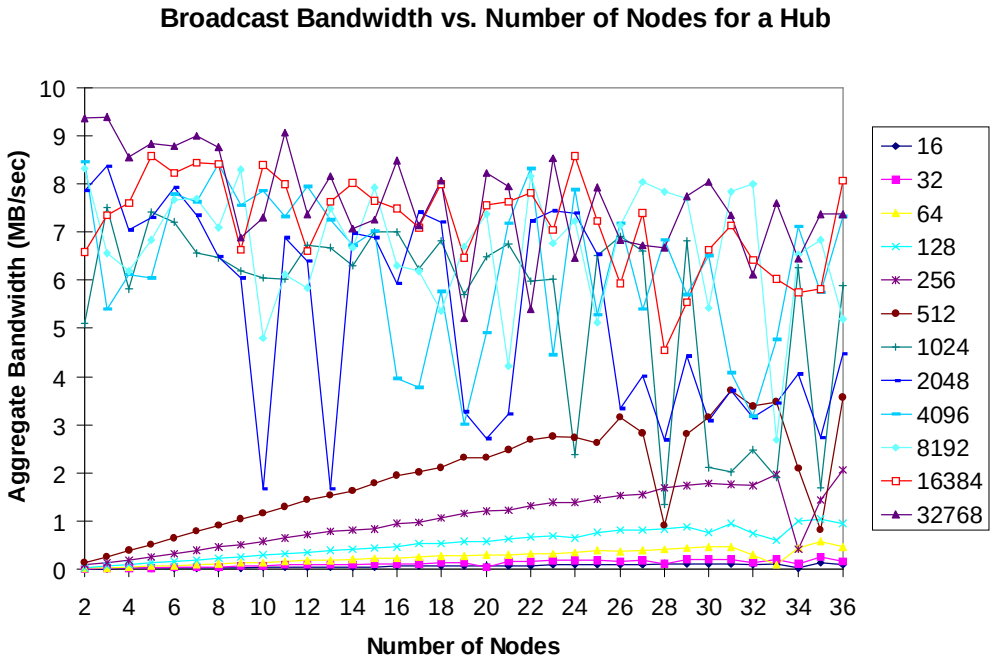


FIGURE 10. ***Collective communication performance on a hub***

increases regularly as the number of nodes increases for messages up to 512 bytes. As can be seen, performance is very irregular for larger message sizes and the maximum aggregate bandwidth is about 9MB/s. Both the irregular performance and the low maximum bandwidth are probably a result of collisions on the shared network and the network's low maximum bandwidth of 100Mb/sec (12.5MB/sec).

The collective communication bandwidth for the switch is shown in Figure 11.

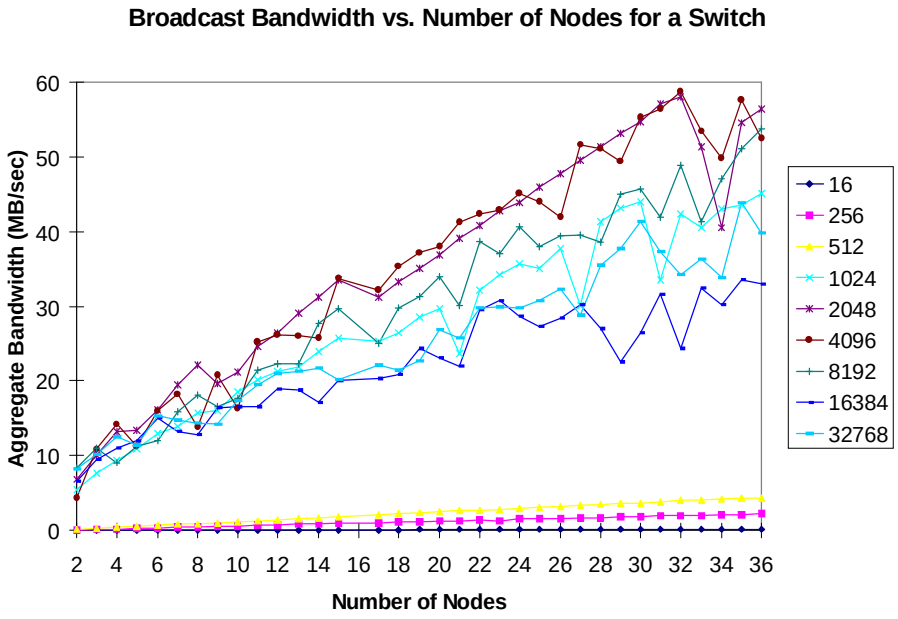


FIGURE 11. ***Collective communication performance on a switch***

As in Figure 9, the aggregate bandwidth scales as the number of nodes is increased. However, for a 16 node machine, the switch bandwidth is about 10MB/sec less than the torus (the switch bandwidth for 16 nodes is about 33 MB/sec). Of course, as the number of nodes scales beyond the number tested with the torus, the switch continues to scale to almost 60MB/sec. The peak bandwidth if the switch was really a non-blocking crossbar would be about 8MB/sec per port or 288MB/sec. However, as evidenced from these experiments, the actual peak bandwidth is less than 25% of peak.

4.2.2 Barrier Synchronization Time

A comparison between the hub, switch, and torus barrier synchronization time is shown in Figure 12. Clearly, the torus and switch provide significantly faster barrier synchronization than the hub as the number of nodes increases. Also, the hub performance is much more inconsistent. The barrier synchronization time for the torus is slightly greater than the switch time, although the differences are minor for the sizes we could test. As the number of nodes increase beyond 16, the switch network scales well, so the barrier time increases only slightly up to 36 nodes.

4.3 The NAS Parallel Benchmarks

Computational Fluid Dynamics (CFD) is one of the primary fields of research that has driven modern supercomputers. This technique is used for aerodynamic simulation, weather modeling, as well as other applications where it is necessary to model fluid flows. CFD applications involve the numerical solution of non-

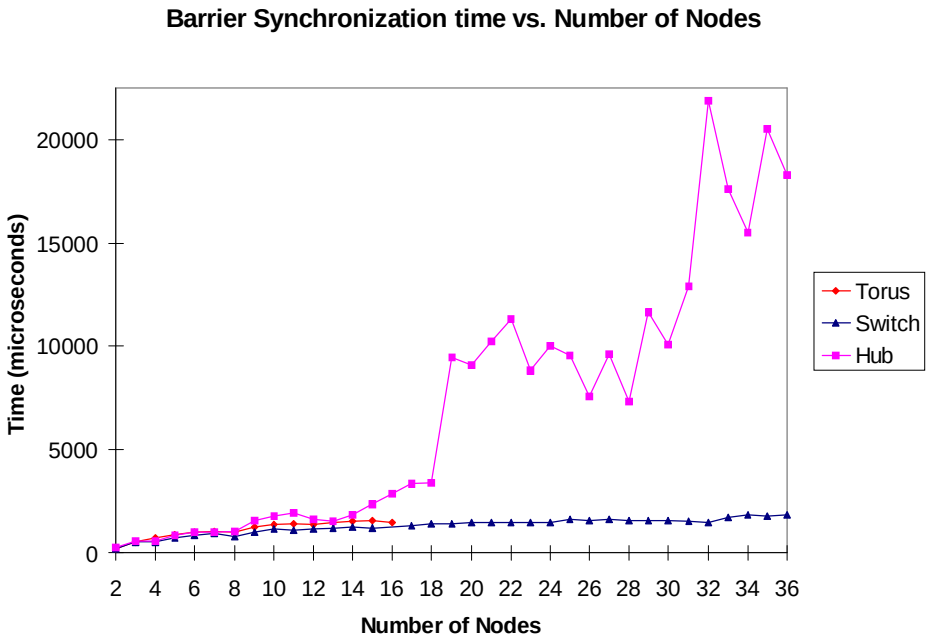


FIGURE 12. ***Comparison of barrier synchronization time for the hub, switch, and torus***

linear partial differential equations in two or three spacial dimensions. The governing differential equations representing the physical laws governing fluids in motion are referred to as the Navier-Stokes equations. The NAS Parallel Benchmarks [BaB91] consist of a set of five kernels, less complex problems intended to highlight specific areas of machine performance, and three application benchmarks. The application benchmarks are iterative partial differential equation solvers that are typical of CFD codes.

In this section, we show results for the NPB 2.2 codes [BaH95] which are MPI implementations of the NAS Parallel Benchmarks. The NPB 2.2 benchmark set includes codes for the three application benchmarks, BT, SP, and LU. It also includes code for 4 of the five original kernel benchmarks, EP, FT, MG, and IS (version 2.2 does not include code for CG). In this paper we present results for the three application benchmarks, we chose not to present results for the kernel benchmarks because they do not add substantially to our understanding of the networks tested. These benchmarks are designed for four different problem sizes, called classes, S, A, B, and C. For this paper we present results for the Class A, B, and C sizes, Class S is a “sample” size and is not interesting on systems big enough to run the larger sizes. The matrix size and iteration count for

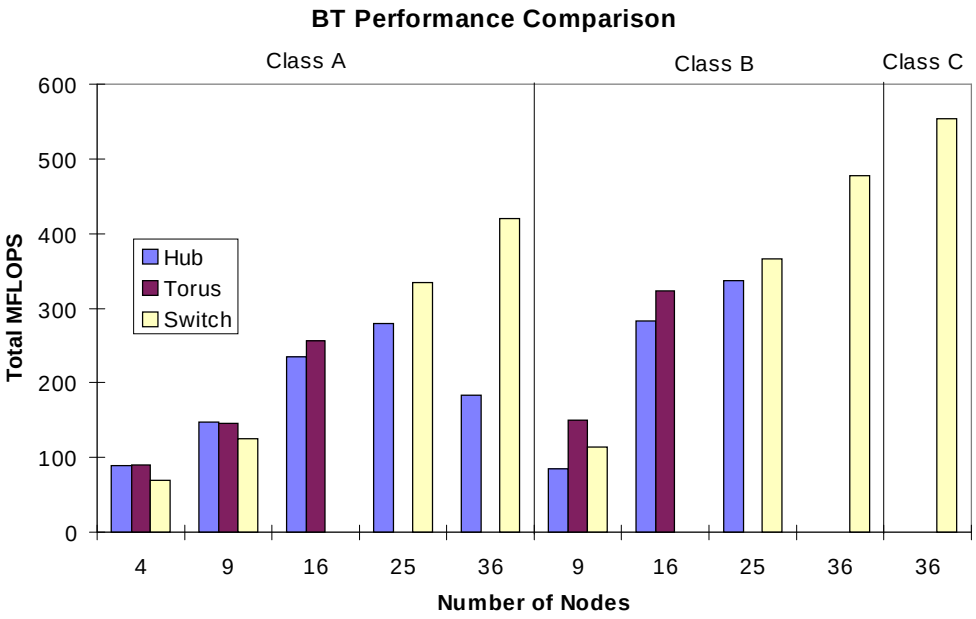


FIGURE 13. ***Comparison between hub, switch, and torus topologies for BT benchmark***

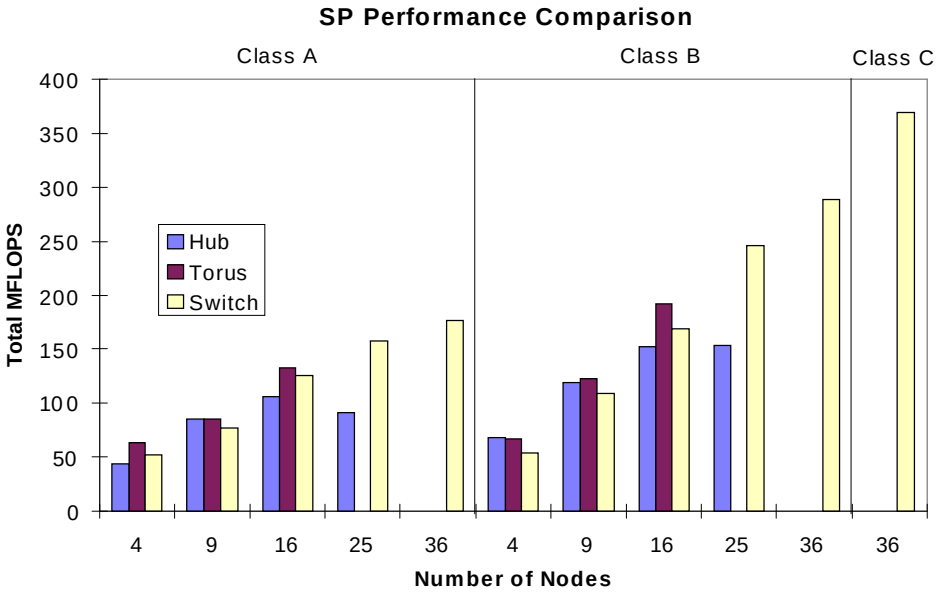


FIGURE 14. ***Comparison between hub, switch, and torus topologies for SP benchmark***

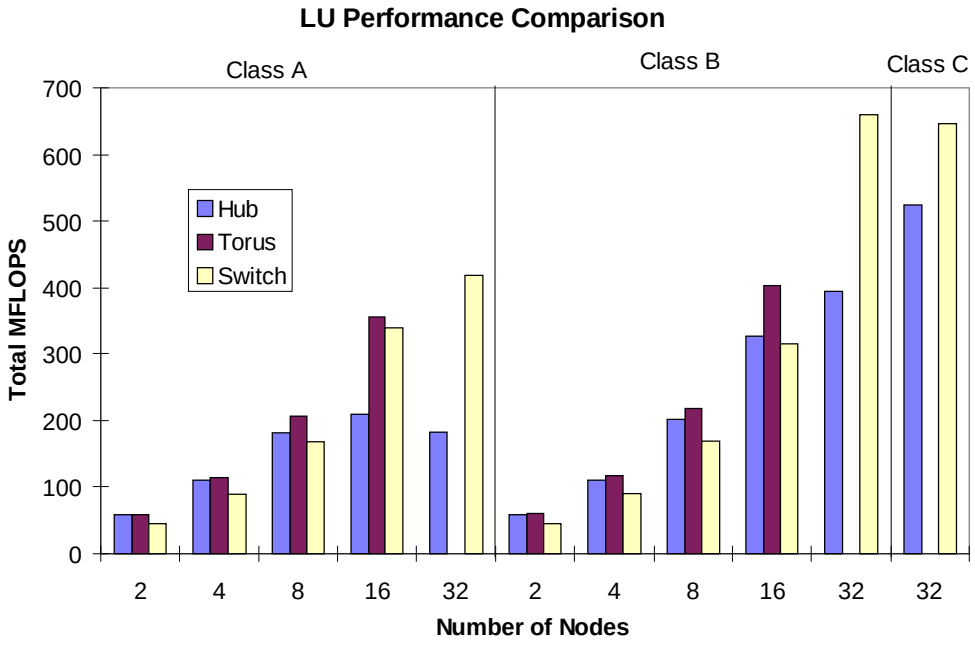


FIGURE 15. ***Comparison between hub, switch, and torus topologies for LU benchmark***

these sizes are shown in Table 1. Tables containing the results of these

Table 1:NAS Parallel Benchmark Sizes

Benchmark	Matrix Size for Class			Iterations
	A	B	C	
BT	64x64x64	102x102x102	162x162x162	200
SP	64x64x64	102x102x102	162x162x162	400
LU	64x64x64	102x102x102	162x162x162	250

experiments are presented in the appendix of this paper.

The NAS parallel benchmarks were compiled with the Portland Group's Fortran 77 compiler, pgf77, using the options: **`-O -Knoieee -Munroll -Mdalign -tp p6`**. These benchmarks were run for all valid sizes that would fit on the available nodes, This included 1, 2, 4, 8, 16, and 32 nodes for LU, FT, MG, and IS because they required node counts that were a power of two. BT and SP, however, required sizes that were perfect squares, so they were run for 1, 4, 9, 16. 25, and 36 nodes. Note that in the appendix single node times are only shown for the hub, though they should be the same for the torus since the network is not used. In addition, we measured performance of the torus for both for 4 nodes in a row (nodes 1, 2, 3, and 4 of Figure 1) and for a 2x2 layout (i.e., nodes 1, 2, 5, and 6 on Figure 1). This made a minor difference in performance, however it may be important on larger systems. Finally, torus results are only shown with up to 16 nodes since we did not construct a larger torus.

In Figures 12, 13, and 14 the performance of the three NAS application benchmarks on a hub, switch, and torus are compared. The first thing to notice from the graphs is that in many cases the hub does not perform as poorly as one might expect, particularly for the Class A benchmarks. In most cases the performance of the torus was better than the hub. In the few cases where the hub was better, through the difference was negligible. Also, as expected, differences between the hub and torus increase as the number of nodes increases, due to contention on the hub.

The switch, however, in all cases performs worse than the torus. In addition, for smaller numbers of nodes, the hub outperforms the switch. However, the switch does dramatically outperform the hub for larger numbers of nodes. The performance of the torus indicates that the higher aggregate bandwidth is more important than the increase of latency. It may also indicate a lot of node N to N +/- 1 communication, which is generally a nearest neighbor on the torus. It is harder to explain why the hub outperforms the switch for small numbers of nodes. However, it is clear that the hub is not a good choice for more than 16 nodes.

Of the application benchmarks, LU has the highest performance, 659 MFLOPS (Class B) for a 32 node switch, and 524 MFLOPS (Class C) for a 32 node Hub, and 402 MFLOPS (Class B) for a 16 node torus. This result is typical of the measurements we have made on Ethernet networks, i.e., LU's network characteristics seem to match nicely with Ethernet. BT also performs well, 554 MFLOPS (Class C), for the switch and 336 MFLOPS (Class B) for a 25 node hub, and 323 MFLOPS (Class B) for a 16 node torus, though it is significantly slower than LU. SP performs the worst, with less than half of the performance of LU. This would indicate that while some algorithms do match well to the performance characteristics of Ethernet, others perform significantly worse.

5.0 Hybrid Hub/Switch Based Networks

One way of reducing the cost of a switch is to use a hybrid hub/switch based network as we described in Section 2. The question, however, is what effect the hub density (i.e., the number of hub ports used per switch port) has on performance. To measure this, we ran the same benchmarks we utilized in section 4 on a system with a hub density varying from 1 (i.e., no hubs) to 7 (7 nodes per hub, each hub attached to the switch).

The latency and bandwidth were for the most part the same as the switch (approx. 190µsec latency, 8MB/sec). What was interesting was that for a hub density of 1-6, the latency and bandwidth were relatively insensitive to node placement. However, when we went to a hub density of 7, the latency within a hub went to 200µsec and across hubs (through the switch) to 225µsec. There seems to be no good reason for this, except that there may be some constraint in the switch software.

The collective communication measures were less surprising. Figure 16 shows

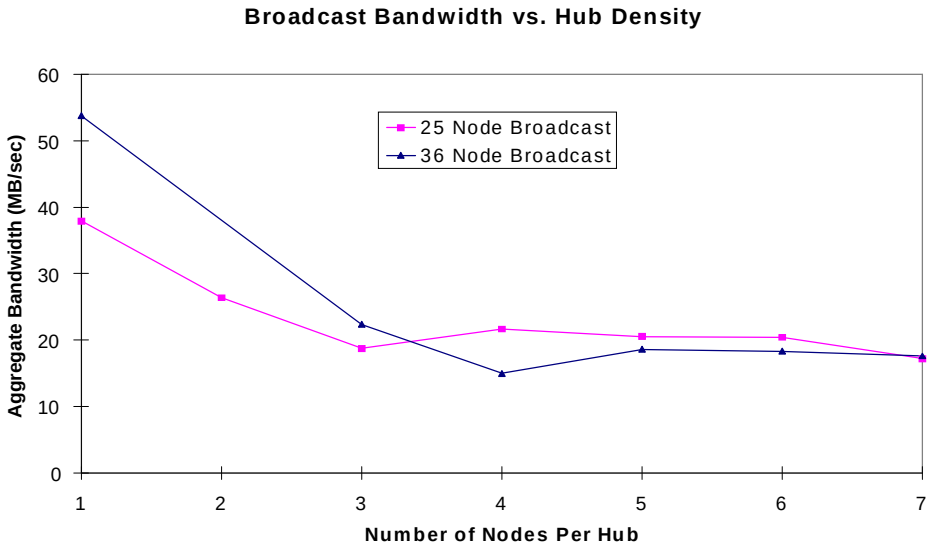


FIGURE 16. ***The effect of hub density on 8192 byte broadcast bandwidth***

the broadcast bandwidth vs. hub density. The graph indicates that the aggregate bandwidth drops off quickly after the first level of hubs is added. However, the drop levels off to about 20MB/sec aggregate bandwidth for 3-7 nodes. This is about 1/2 of the 25 node aggregate bandwidth for a switch and 1/3 of the 36 node aggregate bandwidth. What is interesting is that the bandwidth does not drop off any more with 7 nodes per hub, even though the previous experiments show that the aggregate bandwidth across a totally hub based network is only 9MB/sec.

Barrier synchronization performance is shown in Figure 17. This graph indicates that the barrier synchronization time is virtually unaffected by hub densities from 1-6 with only a slight degradation at a hub density of 7. This degradation at 7 nodes per hub may be related to the latency effects mentioned previously. This is not particularly surprising because Figure 12 shows the barrier synchronization time for a 7 node barrier on a hub is similar to that on a switch.

Probably the most important measurement we can make on a hybrid network is its effect on the performance of the NAS parallel benchmarks. Therefore, we measured the performance of the three application benchmarks and plotted MFLOPS vs. hub density in Figure 18, 19, and 20. As these graphs show, the performance of the NAS parallel benchmarks is relatively insensitive to hub densities of 7 or less. Of these benchmarks, SP is the most sensitive to hub density, but the difference is still less than 20% for a hub density of 5 using 36 processors. These measurements indicate that in most cases it will be reasonable to use a hub to expand switch capacity. In particular hub densities of 2 or 3 (which are possible with a single 4-port linksys hubs we utilized) cost about 10% in performance loss for SP and LU, and had virtually no cost for BT.

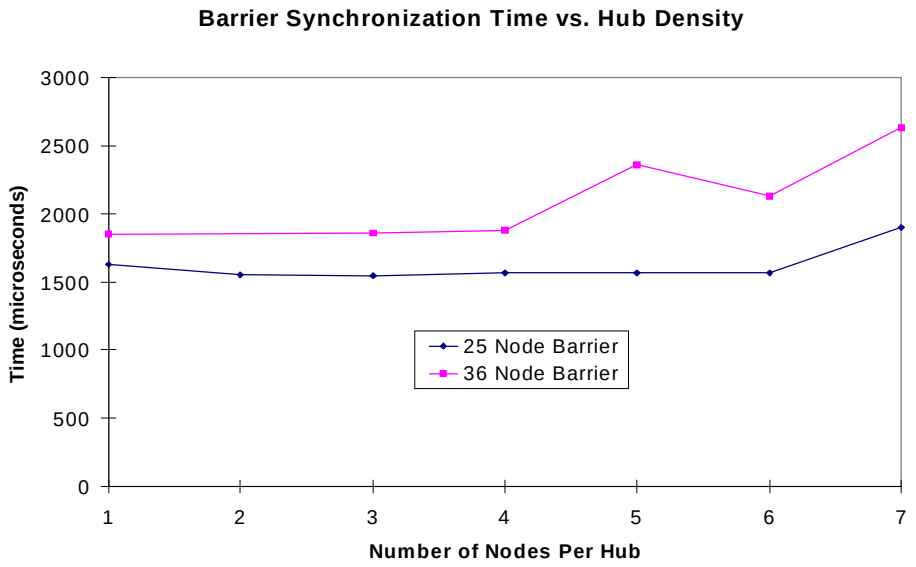


FIGURE 17. ***The effect of hub density on barrier synchronization time***

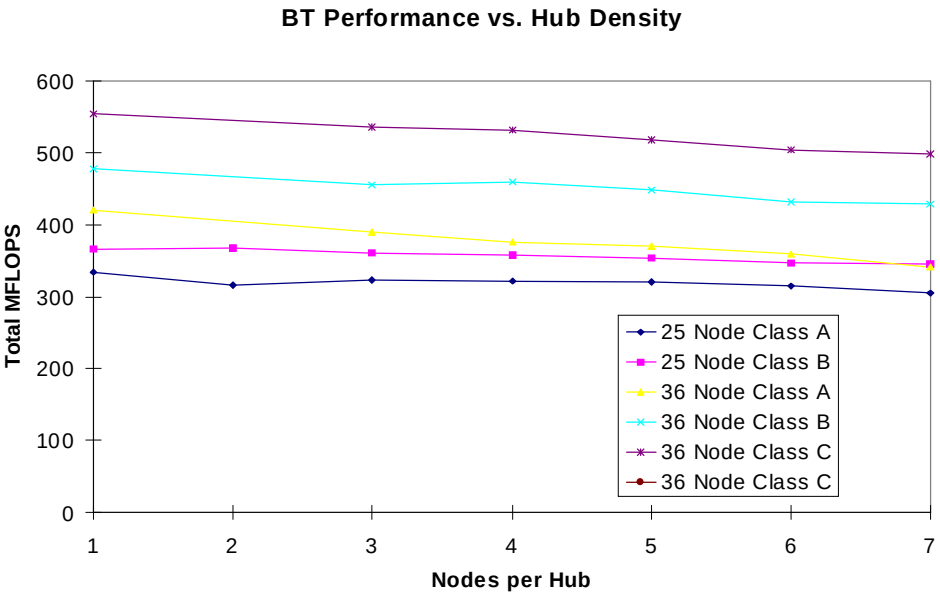


FIGURE 18. ***The effect of hub density on the BT benchmark***

6.0 Conclusion

Our experiments show that for the sizes measured (less than or equal to 16 processors) a torus outperforms both an ethernet switch and hub based network. In addition, for small systems the hub outperforms an ethernet switch, though for systems of more than 16 processors an Ethernet switch based network continues to scale while a hub based network loses performance. We also have shown that

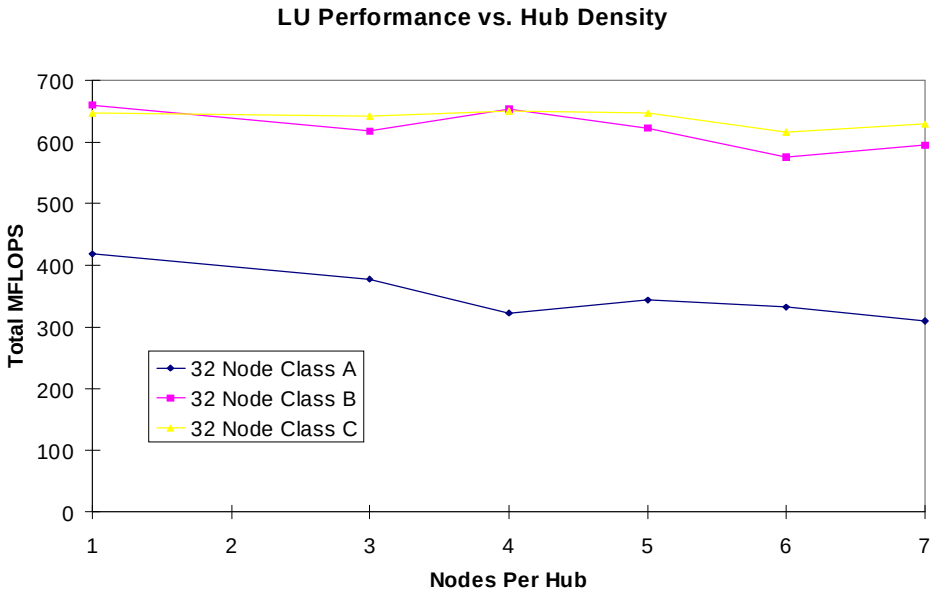


FIGURE 19. ***The effect of hub density on the LU benchmark***

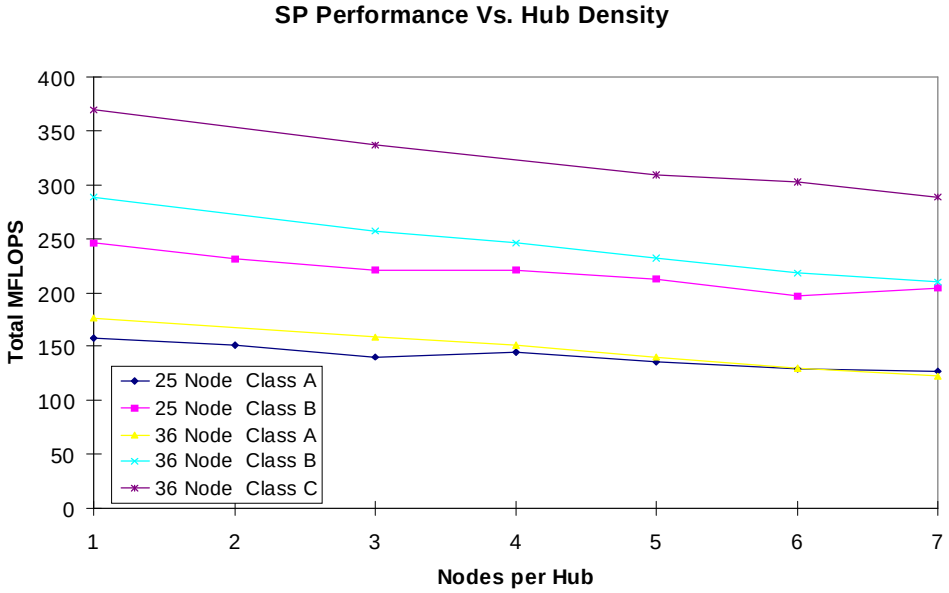


FIGURE 20. ***The effect of hub density on the SP benchmark***

in a mixed hub/switch based network, the NAS parallel benchmarks continue to perform well with hub densities up to 7 nodes per hub.

In conclusion, the final network utilized for Whitney can not be a single hub based system, but some combination of ethernet switching or TCP/IP routing with hubs should perform well. The final decision for a Whitney network, however, must not only be based on performance constraints, but also on system manageability and cost. In terms of manageability the switch/hub hybrid

network is the winner since it requires no special node routing configuration. In terms of cost, however, the torus is the winner. What is clear, however, is that it is possible to build a scalable network for Whitney, even with the limitations of current ethernet switches that only scale to about 100 nodes, and virtually any scalable network will perform well with the NAS parallel benchmarks.

7.0 References[5]

[BaB91] D.H. Bailey, J. Barton, T.A. Lasinski, and H. Simon, *The NAS Parallel Benchmarks*, Tech. Report RNR-91-002, NASA Ames Research Center, 1991.

[BaH95] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow, *The NAS Parallel Benchmarks 2.0*, Tech. Report NAS-95-020, NASA Ames Research Center, 1995.

[Fin95] S.A. Fineberg, *Implementing Multidisciplinary and Multi-zonal Applications Using MPI*, Tech. Report NAS-95-003, NASA Ames Research Center, 1995.

[GrL96] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, *A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard*, http://www.mcs.anl.gov/mpi/mpicharticle/paper.html, Argonne National Laboratory and Mississippi State Univ., 1996.

[Iee95] *802.3u-1995 Supplement to CSMA/CD: MAC Parameters, Physical Layer, and MAUs*, Institute of Electrical and Electronic Engineers, 1995.

[Mes94] Message Passing Interface Forum, *MPI: A Message Passing Interface Standard*, Computer Science Dept. Technical Report CS-94-230, University of Tennessee, 1994.

5. NAS technical reports are available via the WWW at URL: **`http://www.nas.nasa.gov`**

Appendix: NAS Parallel Benchmark Results

			Total MFLOPs		
Benchmark	Class	Nodes	Hub	Torus	Switch
BT	S	1	29.4		
BT	S	4	47.77	70.53	60.69
BT	S	9	40.22	103.89	105.2
BT	S	16	28.73	139.8	128.25
BT	A	1	23.67		
BT	A	4	88.84	91.02	68.99
BT	A	9	148.09	146.07	124.82
BT	A	16	235.28	255.79	226.06
BT	A	25	280.11		333.7
BT	A	36	184.35		420.29
BT	B	9	85.07	149.9	114.48
BT	B	16	282.25	323.31	257.07
BT	B	25	336.21		365.66
BT	B	36			477.56
BT	C	36			554.31
SP	S	1	30.63		
SP	S	4	18.86	52.57	49.91
SP	S	9	11.19	41.67	59.82
SP	S	16	9.99	41.35	52.91
SP	A	1	18.97		
SP	A	4	43.73	63.07	51.65
SP	A	9	85.77	85.23	76.88
SP	A	16	105.54	133.11	124.96
SP	A	25	91.11		157.41
SP	A	36			176.44
SP	B	1	18.57		
SP	B	4	67.34	67.21	53.88
SP	B	9	118.61	122.87	108.87
SP	B	16	152.36	191.82	168.73
SP	B	25	153.34		246.23
SP	B	36			288.99
SP	C	36			369.27
LU	S	1	55.39		
LU	S	2	25.56	25.65	18.42
LU	S	4	39.71	40.05	16.2
LU	A	1	30.96		
LU	A	2	57.96	59.55	45.49
LU	A	4	110.3	114.84	89.2
LU	A	8	181.82	206.61	168.55
LU	A	16	210.53	355.37	339.16
LU	A	32	182.57		418.77
LU	B	1	30.82		
LU	B	2	59.47	60.47	46.09
LU	B	4	111.11	118.12	90.76
LU	B	8	201.79	218.23	168.81
LU	B	16	328.1	402.22	315.55
LU	B	32	394.43		659.52
LU	C	32	523.62		646.59



NAS TECHNICAL REPORT

Title:

Analysis of 100Mb/s Ethernet for the Whitney Commodity Computing Testbed

Author(s):

Samuel A. Fineberg and Kevin T. Pedretti

Reviewers:

"I have carefully and thoroughly reviewed this technical report. I have worked with the author(s) to ensure clarity of presentation and technical accuracy. I take personal responsibility for the quality of this document."

Two reviewers must sign.

Signed: _____

Name: _Tom Faulkner_____

Signed: _____

Name: _Chris Kuszmaul_____

After approval, assign NAS Report number.

Branch Chief:

Approved: _____

Date:

NAS Report Number: