Performance Evaluation of Communication Networks for Distributed Computing

Rod Fatoohi*

Report NAS-95-009, September 1995

NASA Ames Research Center
Moffett Field, California

Abstract

Several new high-speed network technologies have emerged in the past few years. Some of these networks offer high speed links and switch topology with up to a 100 Mbytes/s transfer rate per link. This is a study to evaluate the performance of these networks in distributed computing environments. Six networks are evaluated here: HiPPI, ATM, Fibre Channel, IBM Allnode switch, FDDI, and Ethernet. These networks are parts of two testbeds: DaVinci - a cluster of 16 SGI R8000 workstations at NASA Ames - and LACE - a cluster of 32 IBM RS6000 workstations at NASA Lewis. The performance results at three programming levels are presented and compared. These levels are: BSD socket programming interface, PVM message passing library which uses BSD socket interface, and three simulated applications of the NAS Parallel Benchmarks using PVM. The results show that the emerging network technologies can achieve reasonable performance under certain conditions. However, at the application level the network performance is still far behind the theoretical peak rates.

A condensed version of this report appeared in the Proceedings of the Fourth International Conference on Computer Communications and Networks (Las Vegas, 1995), IEEE Computer Society Press.

*The author is an employee of Computer Sciences Corp. Mailing address: Rod Fatoohi, Mail Stop T27A-1, NASA Ames Research Center, Moffett Field, CA 94035, Ph. (415) 604-3486, Fax. (415) 604-3957, E-mail: fatoohi@nas.nasa.gov.

1 Introduction

In recent years several new network technologies have emerged in local and wide area networks. At the same time there has been increased interest in distributed computing as a viable option for many large-scale applications. These applications, such as scientific computing and visualization, require high computing power and fast communication networks. The rapid increase in the computing performance of microprocessors as well as the easy availability of programming tools, such as the Parallel Virtual Machine (PVM), have made distributed computing more appealing for scientists and engineers. However, the conventional network technologies may not be quite adequate for such applications due to their high latency and low throughput [6]. This might change with the emerging technologies that promise higher bandwidth, lower latency and richer connectivity.

The emerging high-speed networks can theoretically transmit data per link at a hundred megabytes per second. Combined with high connectivity (switched network), the aggregate network bandwidth can reach gigabytes per second. However, the maximum achievable bandwidth at the application level is still very far behind the maximum network bandwidth; see [12, 13] for details. The major cause for this poor performance is the interaction between the host and the network interface, which is called the network subsystem. Lin et al. [13] identified three major areas of the network subsystem: the hardware architecture of the host (memory, system bus, I/O adapter, and I/O bus), the host software system (operating system, application programming interface, higher level protocol process, and device driver), and the network interface. All these components can contribute to the performance of the communication networks. Lin et al. [13] also studied the network subsystem for the Fibre Channel network and showed that by modifying the device driver, the maximum achievable bandwidth was improved by 75%.

This is an attempt to evaluate the performance of emerging network technologies as well as existing technologies used in distributed computing. Here we study the performance of HiPPI, ATM, Fibre Channel, and an IBM switch called Allnode as examples of the emerging technologies. We also study Ethernet and FDDI as examples of the traditional networks. Two testbeds are considered: DaVinci at the Numerical Aerodynamic Simulation (NAS) Systems division at NASA Ames Research Center and LACE at NASA Lewis Research Center. DaVinci is a cluster of 16 SGI R8000 workstations connected by Ethernet, FDDI, and HiPPI. LACE (Lewis Advanced Cluster Environment) is a cluster of 32 IBM RS6000 workstations connected by Ethernet, FDDI, ATM, Fibre Channel, and IBM Allnode switch. Also, an IBM SP2 machine at NAS is considered for comparison.

This study was performed at three levels: BSD socket interface using two programs: ttcp and bench [10], PVM message passing library which uses BSD sockets, and application level using PVM. Several communication tests were used for the first two levels while the three simulated applications of the NAS Parallel Benchmarks [2] were used for the third level. The emphasis of this work is to examine the performance of these networks under different conditions and identify the factors that influence their performance. The communication tests were chosen to study such factors as: message size, socket buffer size, and simultaneous communication. The algorithms

chosen for the application tests have different communication requirements such as communication pattern, number of messages and message sizes. Here we are interested in the interplay between these algorithms and communication networks which have different characteristics such as topology, latency, and throughput.

In the first few sections we give a brief description of the networks and systems used for this work. Then we present and discuss our results in performing three sets of tests. The first two sets are communication tests while the third one is application tests. Finally we offer some concluding remarks about the efficacy of the communication software of public networks to obtain satisfactory performance.

2 Communication Networks

Communication networks can be divided into two basic types: circuit-switched and packet-switched. In circuit-switching, a dedicated path is established between two hosts whenever they want to communicate. During the communication time, no other traffic can travel over the established path. The telephone system uses circuit-switching technology. In packet-switching, the sender divides the message into segments, called packets, and sends them over the network. Each packet carriers identification of its destination. Ethernet and FDDI are examples of packet-switched networks.

Another important feature of a network is its topology; i.e., whether it is a shared medium or a switch. In a shared medium network, only one host can send message(s) to the other hosts on the network. This means that if more than one host want to transmit messages simultaneously only one host can transmit while the others have to wait. This is also sometimes called "serialized communication." On the other hand, in a switched network two and more hosts can communicate through the switch simultaneously. Depending on the architecture of the switch, there could be a delay (blocking) when many hosts try to send messages at the same time. Ethernet and FDDI are examples of shared medium networks while HiPPI and Allnode are switched networks. However, both Ethernet and FDDI have switch technology available.

2.1 Ethernet

The Ethernet is a 1.25 Mbytes/s broadcast bus technology with distributed access control [5]. It is a bus because all connected hosts share a single communication channel; it is broadcast because all transceivers, which connect host interfaces to the network, receive every transmission; its access control is distributed because there is no central authority granting access. The host interface chooses packets the host should receive and filters out all others. Ethernet uses Carrier Sense Multiple Access with Collision Detect (CSMA/CD) technology. In CSMA, each host that wants to transmit a message listens in to check if the network is idle. When no transmission is sensed, the host starts transmitting. If a host detects a collision, which can occur when two hosts sense the channel is idle at the same time, it aborts transmission, and tries again after a delay. The maximum packet size for Ethernet is 1500 bytes.

2.2 FDDI

Fiber Distributed Data Interface (FDDI) is a 12.5 Mbytes/s fiber optic token ring network [15]. It has two fiber rings. If either one breaks, the other can be used as a backup. Hosts use a token to control transmission of data. In FDDI, a token circulates around the ring whenever all hosts are idle. If a host needs to transmit data, it must capture the token first. Then it can transmit its frames for a predefined period of time. Finally, the host reissue a new free token. The maximum frame size for FDDI is 4500 bytes.

2.3 HiPPI

High Performance Parallel Interface (HiPPI) is a point to point link that uses twisted-pair copper cables with a maximum length of 25 meters [4]. The HiPPI standard gives specifications for transmitting data at two speeds: either 100 Mbytes/s or 200 Mbytes/s. Here we discuss only the 100 Mbytes/s HiPPI since that what is available in the testbed platforms. HiPPI cables are sets of 50 lines; 32 lines for data and the rest for other signals. The data transmission is done in parallel; one bit per line. The HiPPI network uses crossbar switches to connect hosts, which provide non-blocked dedicated connections. Once a connection is established a packet can be sent from one host to another. Each packet contains one or more bursts, and each burst contains up to 256 32-bit words. There is also serial HiPPI which extends HiPPI links to a maximum distance of 10 kilometers by using fiber-optic cables.

2.4 ATM

Asynchronous Transfer Mode (ATM) is a packet oriented transfer mode based on fixed length cells (53 bytes) [11]. Each cell consists of a 48-byte information field and 5-byte header. The header contains mostly routing information for moving the cells from one node to the next. ATM is connection-oriented; i.e., every cell travels over the same route. All services (voice, video, data) can be transported via ATM, including connectionless services. The ATM standard describes only the cell format, without specifying rates, framing, or physical medium. For this work, we used two Fore Systems ASX-200 switches with TAXI ports. The maximum transfer rate for a TAXI line is 12.5 Mbytes/s.

2.5 Fibre Channel

Fibre Channel (FC) is a versatile technology, defined by the Fibre Channel Standard (FCS), that offers both circuit switching and packet switching at multiple data rates [14]. Fibre Channel topologies include point-to-point, loop, or switch (which is called a fabric). Information can flow between two ports in both directions simultaneously. The data is sent in frames that are maximum 2148 bytes long (2048 bytes data, 64 bytes optional header, and 36 bytes for addresses and link control information). For this work, we used the Ancor MCA CIM 250 Interface Adapter and Ancor CXT 250 Fibre Channel Switch which has a 25 Mbytes/s data rate per port [1]. The CXT switch is a two-dimensional switching architecture that uses both space-division

and time-division interconnection techniques. Space division switching allows direct connections between nodes on the network while time division switching allows time-multiplexed connections among all nodes on the network.

2.6 Allnode

The Allnode is an IBM proprietary multistage switch that uses four 8×8 crossbar blocks to create a 16×16 switch [9]. There are 32 1-byte wide unidirectional paths between the adapters and the switch (16 send and 16 receive). The switch has a peak rate of 40 Mbytes/s, but the adapter can only drive it at a peak of 25 Mbytes/s. The adapter further reduces performance at the Micro Channel because it can only send at a peak of 12 Mbytes/s and receive at a peak of 8.5 Mbytes/s. The switch uses circuit switching with no buffering in it. It supports three forms of communication: TCP/IP, File system Application Programming Interface (API), and Low Latency Programming Interface (LLPI).

3 Distributed Computing Clusters

Two clusters of workstations were used for this study: DaVinci and LACE. Both of them are considered as tightly coupled or dedicated clusters. Tightly coupled workstations are usually constructed by gathering several workstations into a controlled environment, with no "primary users", and are often connected by high speed networks. Unlike loosely coupled workstations which usually have primary users and are connected by conventional networks.

3.1 DaVinci

DaVinci is a cluster of 16 SGI Power Challenge L machines with 75 MHz MIPS R8000 processor at NAS. Fourteen of them are single processor machines with 64 Mbytes of memory while the other two are dual processor machines with 256 Mbytes of memory. All machines are connected by Ethernet, FDDI, and HiPPI; and ATM in the near future. The operating system used was IRIX 6.0.

3.2 LACE

Lewis Advanced Cluster Environment (LACE) is a cluster of 32 IBM RS6000 machines, each with at least 64 Mbytes of memory, at NASA Lewis Research Center. The upper 16 machines are 66 MHz RS6000/590 while the lower machines are 50 MHz RS6000/560. All machines are connected by two Ethernet networks: one is public (for file server and other activities) and the other is private (for multiprocessing). Also, the upper 16 machines are connected by ATM, Fibre Channel, and IBM Allnode switch (see Section 2.6). In addition, the lower 16 machines are connected by an older (prototypic) version of the Allnode switch and machines 9 through 24 are connected by an FDDI ring. The operating system used was AIX 3.2.5.

4 Parallel Computer: IBM SP2

The IBM SP2 at NAS has 160 nodes connected by a bidirectional multistage interconnection network. Each node is an IBM RS6000/590 processor with at least 128 Mbytes of memory. The building block of the network is a bidirectional 4 $\times$ 4 (physically 8 input, 8 output) crossbar switching elements [16]. Each switching element has 8 receiver modules, 8 transmitter modules, an unbuffered crossbar, and a central queue. A 16 node bidirectional multistage interconnection, called a frame, is constructed from 8 switching elements. These frames are cascaded to form a larger system. Nodes send messages to other nodes by breaking messages into packets and send them using buffered wormhole routing. Packets vary in length up to 255 1-byte flits (A flit is the smallest unit on which flow control is performed). The switch has a peak rate of 40 Mbytes/s in each direction per link.

The SP2 could be viewed as a dedicated cluster of workstations with a network that has relatively low latency, high bandwidth and can support 160 nodes.

5 Parallel Programming System: PVM

The Parallel Virtual Machine (PVM) is a collection of public-domain system software routines that enables parallel processing on a network of heterogeneous computers as well as parallel computers [7]. It is composed of two parts: a run time system (daemon) that resides on all of the computers participating and a set of user interface primitives that can be incorporated into a Fortran (or C) code. This includes primitives for process control, message passing, and synchronization between processes running on different machines. The version that was used for this work is PVM 3.3.

PVM daemons communicate with one another through UDP sockets while a PVM task communicates with its daemon over a TCP connection. For PVM tasks to communicate with each others, there are two routing modes: normal routing and direct routing. Normal routing involves three steps: a TCP connection between a task and its daemon, a UDP connection between the two daemons, and another TCP connection between the other task and its daemon. Direct routing, on the other hand, establishes a direct TCP link between tasks.

Sending a message is normally composed of three steps in PVM: buffer initialization, message packing into the buffer, and sending the message. Receiving a message normally involves two steps: receiving and unpacking. PVM also provides another mechanism for communication called pack send (psend) and pack receive (precv). Pack send combines the three send steps into one while pack receive combines the two receive steps into one.

In the present work, all our PVM implementations are based on direct routing and psend/precv.

IBM has developed an optimized version of PVM for the SP2 and the Allnode switch called PVMe [8]. Unlike PVM, PVMe does not interface directly with TCP/IP to perform data communication between nodes. Instead, it interfaces with the communication software that runs on the adapters. Also, PVMe does not support direct routing or psend/precv.

6 Communication Tests: BSD Socket Interface

Several communication tests were performed on the two clusters using the two programs: *ttcp* and *bench*. The ttcp program measures point-to-point performance using either TCP or UDP protocols. It has many options including: message size, socket buffer size, number of messages, and setting TCP_NODELAY (which controls buffering in sending data). In this work, ttcp was chosen to measure throughput using TCP since it has flow control and is more reliable than UDP. Also, from tests run with TCP_NODELAY enabled, we found it did not have an impact so TCP_NODELAY was disabled (by default).

The bench program [10] implements two types of tests: bulk transfer and round-trip. In a bulk transfer, a number of messages are transferred back to back through the network. When the transfer completes, the receiver sends a single message back to the sender for acknowledgement. In a round-trip test, messages are sent (one at a time) from one machine to another, then echoed back. In this work, the round-trip test was chosen to measure latency with UDP because of the simple nature of the protocol.

Several experiments were conducted to measure the throughput and latency of the networks under different conditions. The program ttcp was used in EXP_0 through EXP_4 while bench was used in EXP_5. These experiments are:

- EXP_0 (ttcp): Using one transmitter and one receiver, the message size and the socket buffer size were varied randomly to get the message size that can give optimal or near optimal throughput for all networks considered. It was observed that a message of size 32 Kbytes (1 Kbytes = 2^{10} bytes) is large enough to be chosen for the next experiment.

- EXP_1 (ttcp): Using one transmitter and one receiver and a fixed message size (32 Kbytes), the socket buffer size was varied (through doubling) from 1 Kbytes to 1 Mbytes (1 Mbytes = 2^{20} bytes) on DaVinci and up to 64 Kbytes (which is the limit on IBM RS6000 machines) on LACE. The impact of changing the socket buffer size as well as the optimal buffer size were recorded for every network.

- EXP_2 (ttcp): Using one transmitter and one receiver and the optimal buffer size (from EXP_1) for each network, the message size was varied (through doubling) from 128 bytes to 1 Mbytes to study the impact of changing the message size.

- EXP_3 (ttcp): Using two machines and optimal message and buffer sizes (from EXP_1 and EXP_2), two messages were transmitted simultaneously, one from each machine, to measure the bidirectional throughput of a link.

- EXP_4 (ttcp): Using two transmitters and two receivers (four machines) and optimal message and buffer sizes (from EXP_1 and EXP_2), two messages were transmitted simultaneously to measure performance degradation due to network contention.

- EXP_5 (bench): Using one transmitter and one receiver and the optimal buffer size, an eight byte message was transmitted between two machines to measure the latency of the network.

These experiments were performed and the results are reported for the following networks: HiPPI, FDDI, and Ethernet on DaVinci and ATM and Fibre Channel (FC) on LACE. These experiments were also performed on FDDI and Ethernet on LACE but the results are not reported because they are similar to the results of these networks on DaVinci. Each test was run for at least 20 seconds to produce reliable data. All measurements were obtained under conditions of light network traffic. However, we noticed some fluctuations in the timing results.

Figure 1 shows the results of EXP_1; i.e., the achievable throughput of the five networks for a fixed message size and variable socket buffer size. One interesting observation is the dependency of the achievable throughput of HiPPI on the socket buffer size. HiPPI needs a one Mbyte buffer to achieve its best rate (77 Mbytes/s) while it achieves only 9% and 45% of its best rate using buffers of sizes 64 and 256 Kbytes, respectively. Other networks require smaller socket buffers to approach their highest achievable rates. However, because of the size limit on socket buffer for IBM RS6000 machines, it is hard to predict the performance of ATM and FC for larger buffers.

The results of EXP_2 are plotted in Figure 2 using the following socket buffer sizes: 1 Mbytes for HiPPI, 128 Kbytes for FDDI, 8 Kbytes for Ethernet, and 64 Kbytes for ATM and FC. This figure shows that HiPPI can achieve a reasonable percentage of its peak rate (about 77%) only with long messages (16 Kbytes and longer). Other networks can approach their highest achievable rates with smaller messages: 4 Kbytes for ATM and FC, 1 Kbytes for FDDI, and about 64 bytes for Ethernet. The results also show that ATM can achieve about 83% of its peak rate, which is quite remarkable given that the bandwidth available after protocol overhead (ATM and TAXI line) is only 87% [3]. Another observation is that the FC rate for small messages is very low.

Several performance metrics are considered here, including [12]:

r_{max} (maximum achievable throughput): the highest achieved rate. It can be obtained from EXP_1 and EXP_2.

$n_{1/2}$ (half performance length): the message size needed to achieve half of r_{max}. It can be obtained from EXP_2 with some approximation since data were not collected for all message sizes.

t_0 (startup latency): the time required to send a message of minimum size. This can be obtained from EXP_5.

Table 1 lists the obtained values of these metrics for the five networks. This table shows that HiPPI has achieved the highest rate among the five networks but it is only for long messages and large socket buffer sizes. Also, the table shows that latency rangs between 455 μsec for ATM and 850 μsec for Fibre Channel. The $n_{1/2}$ measure shows that Ethernet is very efficient even with very small messages due to the fact that Ethernet is an old technology and its software has been well optimized.

Table 1: Network parameters using BSD socket interface

System	r_{max}	$n_{1/2}$	t_0
	(MB/s)	(Bytes)	(μsec)
DaV/Ethernet	1.0	29	550
DaV/FDDI	8.4	407	561
DaV/HiPPI	77.6	6791	593
LACE/ATM	10.4	1248	455
LACE/FC	6.0	3294	850

Table 2: Network performance under different conditions using ttcp

System	unidirectional	bidirectional	4 machines	msg size	buf size
	(MB/s)	(MB/s)	(MB/s)	(Kbytes)	(Kbytes)
DaV/Ethernet	1.0	0.5	0.5	32	8
DaV/FDDI	8.4	4.6	5.1	32	128
DaV/HiPPI	77.6	42.0	76.7	32	1024
LACE/ATM	10.4	6.3	10.4	32	64
LACE/FC	6.0	4.9	6.0	32	64

Table 2 lists the results of the bidirectional test (EXP_3) and the four machines test (EXP_4) using the optimal message and socket buffer sizes. Also, the best results of the unidirectional test (r_{max}) are given for comparison. Here the network performance represents the bandwidth per link; i.e., the minimum transfer rate achieved for all messages submitted. These results show the performance limitation of Ethernet and FDDI when more than one machine attempts to send a message since both networks use shared medium. On the other hand, the switched networks such as HiPPI, ATM, and FC showed no performance degradation when two machines send messages to two other machines simultaneously. The results are not conclusive for the bidirectional test where the switched networks showed some performance degradations, but by less than 50%.

7 Communication tests: PVM

Two Fortran programs were used to measure the throughput and latency of these networks under the PVM message passing library; see [6] for more details. In the first program, called ring, the processors form a ring where each processor receives a message of prescribed length from a previous processor and sends the same message to the next processor. Only one message goes around the ring at any given time. This program measures point-to-point performance and latency of a network.

The second program, called complete exchange, involves the transposition of a distributed $(m \times p)$ rectangular matrix, where p is the column dimension and is

Table 3: Network parameters using PVM

System	Software	r_{max} (MB/s)	$n_{1/2}$ (Bytes)	t_0 (μsec)
DaV/Ethernet	PVM	1.0	848	833
DaV/FDDI	PVM	3.1	5585	1000
DaV/HiPPI	PVM	3.1	6237	1667
LACE/ATM	PVM	3.2	3550	780
LACE/FC	PVM	2.5	33369	1231
LACE/Allnode	PVMe	6.2	3204	97
SP2	PVMe	27.6	3546	63

same as the number of processors and m is the prescribed row dimension (or column size). At the beginning, the matrix is distributed across the cluster of processors through column-wise partitioning, with each processor holding one complete column of the matrix. At the end of complete exchange operation, the matrix is distributed row-wise across the cluster with each processor holding one or more complete rows. This program is designed to measure the network performance under simultaneous all-to-all communications.

Several message (or column) sizes, ranging between zero and 1 Mbytes, were used for each program. For each message size, the time to perform many iterations, so that the test will last at least five seconds, was measured on each processor and the average over these iterations of the maximum across all processors was reported. Based on the collected data, the achievable throughput for each message size was computed. One difference between the two programs is that ring measures point-to-point performance of a link while complete exchange measures the throughput of a network in a mult-scatter mode, which is a function of the link bandwidth and the network connectivity. All measurements were obtained under conditions of light network traffic.

Figure 3 shows network throughput using the ring program for different systems. PVM was used for HiPPI, FDDI, and Ethernet on DaVinci and ATM and Fibre Channel on LACE while PVMe was used for the Allnode switch on LACE and on the SP2. These results clearly show the superiority of the SP2 network and the Allnode switch over the public networks under PVM even though some of these public networks have comparable transfer rates to the SP2. Most of these networks (HiPPI, ATM, FCS, and FDDI) achieved only about 3 Mbytes/s while Allnode achieved twice that rate, and the SP2 achieved about 28 Mbytes/s (70% of its peak rate).

Table 3 lists the obtained values for some performance metrics (defined in Section 6) using PVM. Latency (t_0) was measured by sending a zero byte (or one byte) message. From this table, t_0 ranges between 0.8 and 1.7 milliseconds for public networks (Ethernet, FDDI, HiPPI, ATM, and FC) under PVM while it is less than 100 μseconds for the SP2 and Allnode. A comparison between Table 3 and Table 1 shows that the performance of PVM lags behind BSD socket interface. Both of them measure point-to-point performance but BSD socket interface does not incur the

overheads of buffer managements, connection management, and state maintenance that PVM has. These overheads are more apparent in high-speed networks (such as HiPPI, ATM, and FC) than in the traditional networks.

Figure 4 shows the aggregate network throughput for four machines using the complete exchange program under PVM. This figure also shows the superiority of the SP2 network and the Allnode switch over the public networks under PVM. Similar results were observed for eight machines.

8 Application Tests: NAS Parallel Benchmarks

The NAS Parallel Benchmarks (NPB) are a set of eight benchmark problems that are designed to measure the performance of parallel computer systems for a subset of algorithms that characterize various computationally intensive aerophysics applications [2]. The eight problems consist of five kernels and three simulated Computational Fluid Dynamics (CFD) applications. Here we are interested in the three simulated applications using the class A size ($64 \times 64 \times 64$) problems. These applications were written in Fortran.

The three simulated applications of NPB contain algorithms which involve a significant amount of inter-processor data movement that typifies many state of the art CFD applications. Here we are interested in the solution of a coupled system of PDEs on logically structured grids using three different time implicit relaxation techniques (LU, SP, BT) [2]. The LU benchmark employs a Symmetric Successive Over-Relaxation (SSOR) scheme resulting in the solution of regular-sparse, block (5×5) lower and upper triangular systems. Both the BT and SP benchmarks use variants of the three-factor, approximate factorization schemes similar to the classical Alternating Direction Implicit (ADI) method. In SP benchmark this results in the solution of a sequence of multiple, independent scalar pentadiagonal systems, each oriented along the three mutually orthogonal directions of the computational space. The BT benchmark is a close relative of the SP benchmark, with the primary difference being the solution of block (5×5) tridiagonal systems, instead of scalar pentadiagonal systems. In summary, these three simulated application codes represent a variety with regard to: a) degree of exploitable concurrency, b) computation to communication ratio and c) frequency and size of inter-processor communications.

The parallel implementation of SP is based on a two-way-pipelined-Gaussian-elimination (SP-twpge) algorithm which uses a 3-D uni-partitioning scheme. The parallel implementation of BT is based on a transpose algorithm (BT-trans) using a 1-D partitioning scheme. The LU parallel implementation is based on the skew hyperplane mapping approach and a 2-D partitioning scheme. More details and references about these algorithms are given in [6]. The total number of messages and total communication volumes per iteration, i.e., relaxation step, for the three algorithms are given in Figures 5 and 6. These numbers are summations across all processors. The BT-trans algorithm requires sending long messages while the LU algorithm requires sending many short messages. The number and volume of messages are moderate for the SP-twpge algorithm. In terms of communication patterns, while BT-trans requires all-to-all communications, SP-twpge and LU require communication only with

the either the nearest or the next to nearest neighbors of a logical processor array.

The execution time per iteration (in seconds) of the three algorithms on DaVinci (using Ethernet, FDDI, and HiPPI), LACE (using ATM, FC, and Allnode), and the SP2 using 1, 4, and 8 nodes are plotted in Figures 7 through 9. (Results for LU on Allnode are not given because of some software problems). PVM was used for all these implementations, except on the SP2 and Allnode where PVMe was used. The same code was run on a single processor (machine) as well as on multiple processors, with no specific single processor optimization performed for any given architecture. The first observation is that the IBM RS6000/590 processor is faster (by a factor of more than two) than the SGI R8000 processor for this class of applications. The other observation is that there is little or no speedup achieved with Ethernet on DaVinci and ATM and FC on LACE in most cases. The problem with Ethernet is that it suffers from low throughput and high latency which makes Ethernet inadequate for this class of applications. The problem with ATM and FC is that these are new technologies and software are not well optimized for message passing libraries such as PVM. The best speedup was achieved with PVMe on the SP2 and the Allnode switch since PVMe is a customized version of PVM for these architectures.

9 Concluding Remarks

This study shows that the emerging network technologies (such as HiPPI, ATM, and Fibre Channel) have the potential to achieve satisfactory performance under certain conditions. However, that level of performance is not currently available at the application level due to communication software overheads. The performance of the SP2 and Allnode switch under PVMe shows that when the communication software is optimized, reasonable performance can be achieved at the application level. Similar efforts for new networks should be pursued. With the increase in the processing power of the workstations, a proportional increase in the achievable network throughput is needed to perform a class of large-scale applications, which requires significant amount of communication, efficiently in a distributed computing environment.

Acknowledgment

I would like to thank Sisira Weeratunga of CSC for providing the NAS Parallel Benchmark codes. Also, I am thankful to Steve Quan of CSC and Jim Feeney of IBM for many technical discussions. Finally, I would like to express my appreciation to the LACE team at NASA Lewis for allowing me access and their assistance in using the LACE cluster. This work was funded through NASA contract NAS 2-12961.

References

[1] Ancor Communications, Inc., *MCA CIM 250 Installer's/User's Manual*. Publication No. PUB 009 A Rev. C. Also in URL: "http://ancor.com/".

[2] Bailey, D., Barton, J., Lasinski, T., and Simon, H., eds., *The NAS Parallel Benchmarks*, Report RNR-91-02, NASA Ames, Moffett Field, CA, January 1991.

[3] Cavanaugh, J., *Protocol Overhead in IP/ATM Networks*, Minnesota Supercomputer Center, Inc., Minneapolis, MN, August 1994.

[4] CERN, *General Introduction to HIPPI.* URL: "http://www.cern.ch/HSI/hippi/introduc.htm".

[5] Comer, D., *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, Prentice Hall, Inc., 1988.

[6] Fatoohi, R. and Weeratunga, S., *Performance Evaluation of Three Distributed Computing Environments for Scientific Applications.* In Proceedings of Supercomputing'94, (Washington, November 1994), IEEE Computer Society Press, pp. 400 – 409.

[7] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., *PVM 3 User's Guide and Reference Manual.* Report ORNL/TM-12187, Oak Ridge National Lab., Oak Ridge, TN, September 1994.

[8] IBM, *IBM AIX PVMe User's Guide and Subroutine Reference.* Release 3.0, Order Number SH23-0019-02, July 1994.

[9] IBM, *IBM Allnode Interconnect System: Overview.* Part # 57G2977, April 1994.

[10] Keeton, K., Anderson, T., and Patterson, D., *LogP Quantified: The Case for Low-Overhead Local Area Networks.* In Proceedings of HOT Interconnects III, Stanford, August 1995.

[11] Lane, J., *ATM Knits Voice and Data on any Net.* IEEE Spectrum,Vol. 31, No. 2, February 1994, pp. 42 – 45.

[12] Lin, M., Hsieh, J., Du, D., Thomas, J., and MacDonald, J., *Distributed Network Computing over Local ATM Networks.* In Proceedings of Supercomputing'94, (Washington, November 1994), IEEE Computer Society Press, pp. 154 – 163.

[13] Lin, M., Hsieh, J., Du, D., and MacDonald, J., *Performance of High-Speed Network I/O Subsystem: Case Study of a Fibre Channel Network*, In Proceedings of Supercomputing'94, (Washington, November 1994), IEEE Computer Society Press, pp. 174 – 183.

[14] Meggyesi, Z., *Fibre Channel Overview.* URL: "http://www.cern.ch/HSI/fcs/spec/overview.htm".

[15] Mirchandani, S., and Khanna, R., eds., *FDDI Technology and Applications*, John Wiley & Sons, Inc., 1993.

[16] Stunkel, C., Shea, D., Grice, D., Hochschild, P., and Tsao, M., *The SP1 High-Performance Switch.* In Proceedings of the Scalable High Performance Computing Conference, (Knoxville, May 1994), pp. 150 – 157.

Figure 1. Throughput for fixed msg size (32 KB) using ttcp

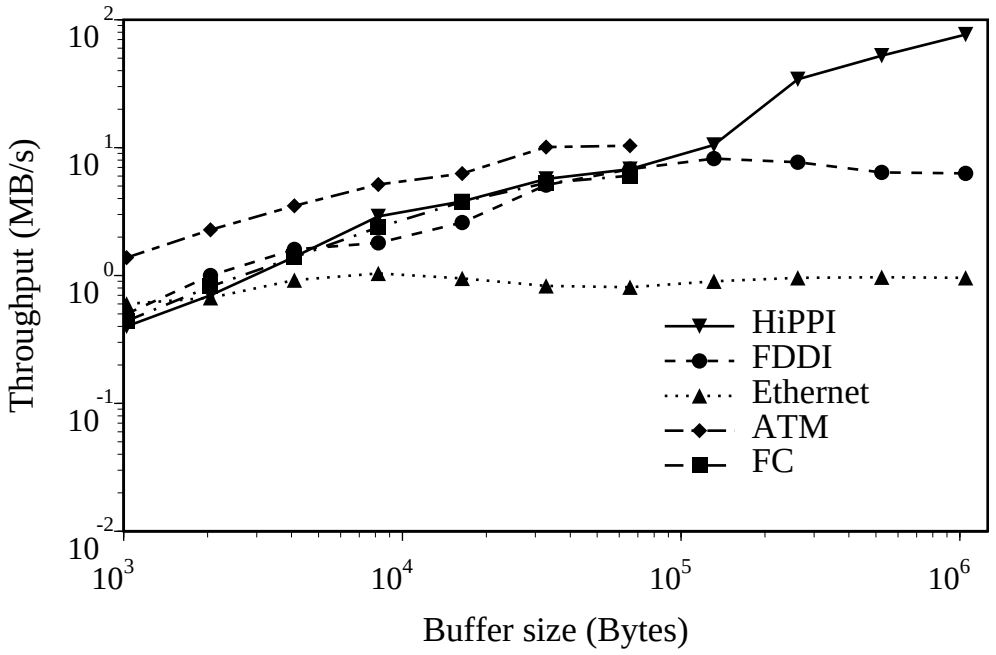


Figure 2. Throughput for fixed buffer using ttcp.

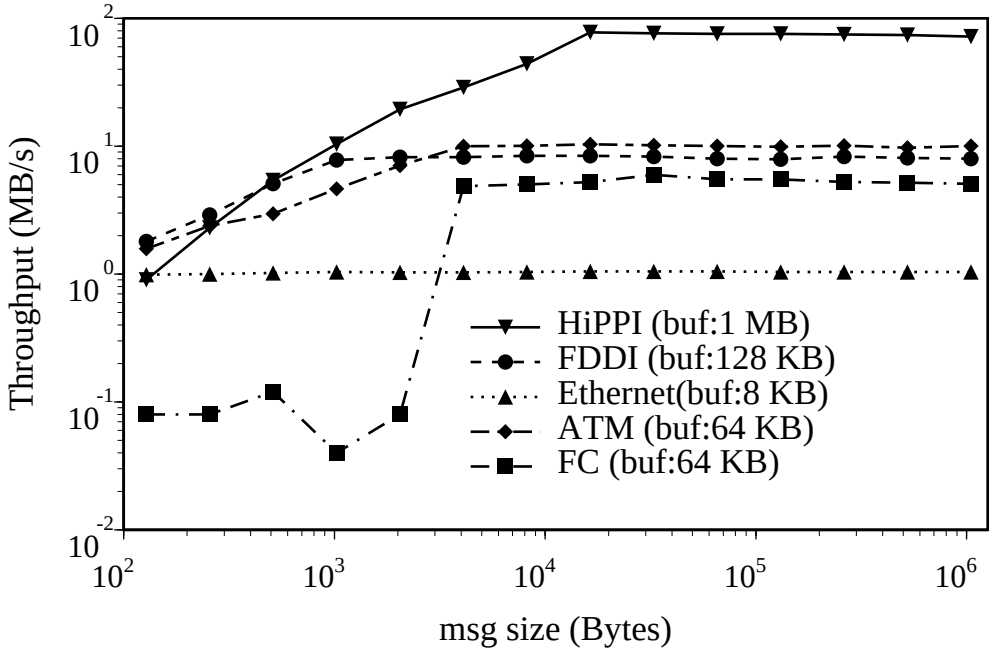


Figure 3. Network Bandwidth using ring under PVM.

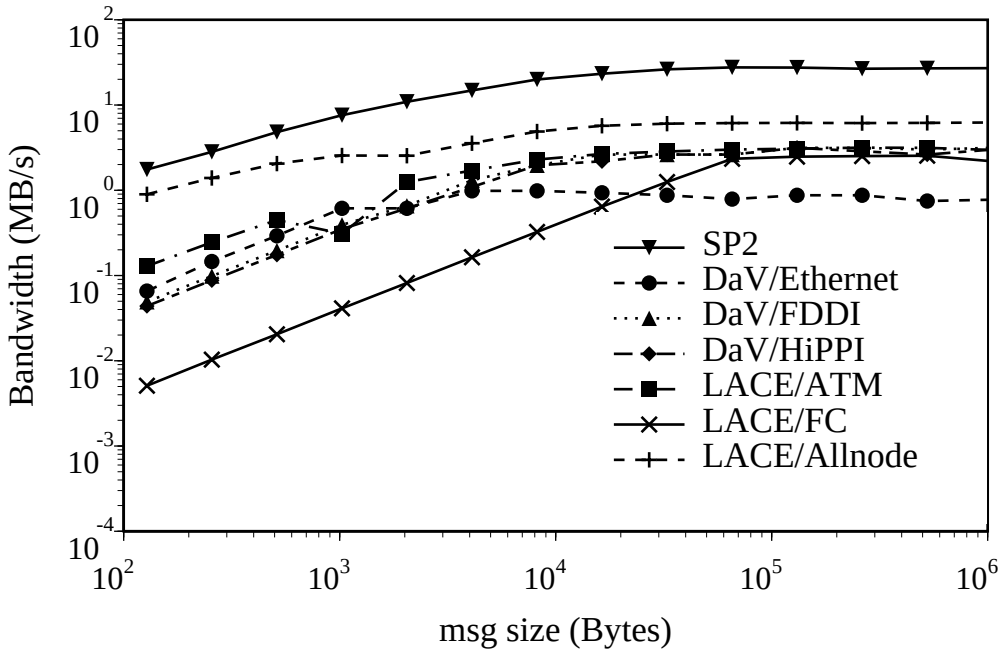


Figure 4. Aggregate Bandwidth using 4 Proc with PVM.

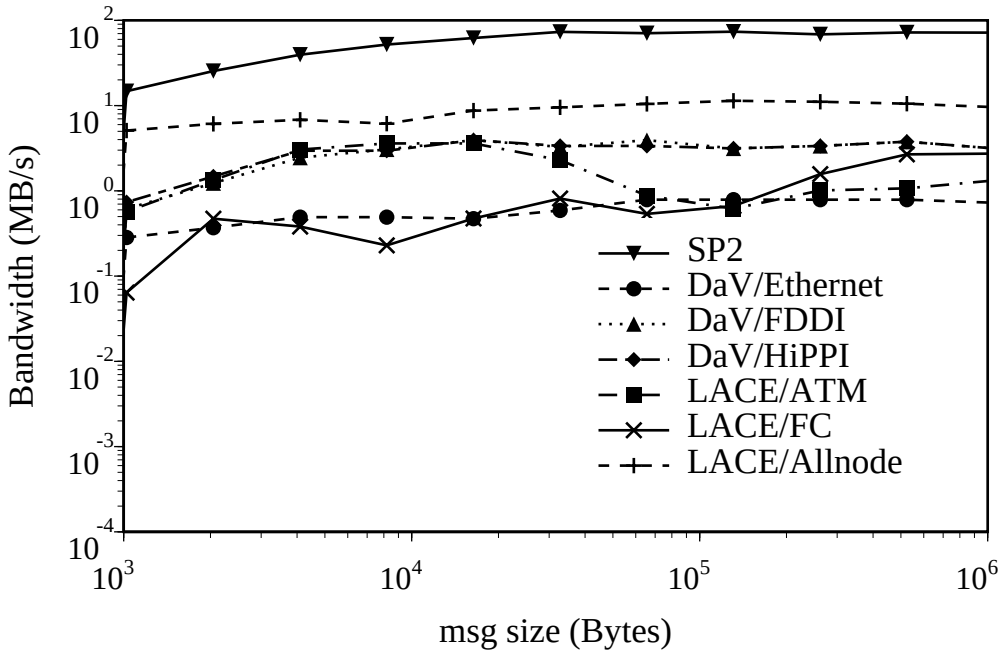


Figure 5. Total Number of Messages for NPB.

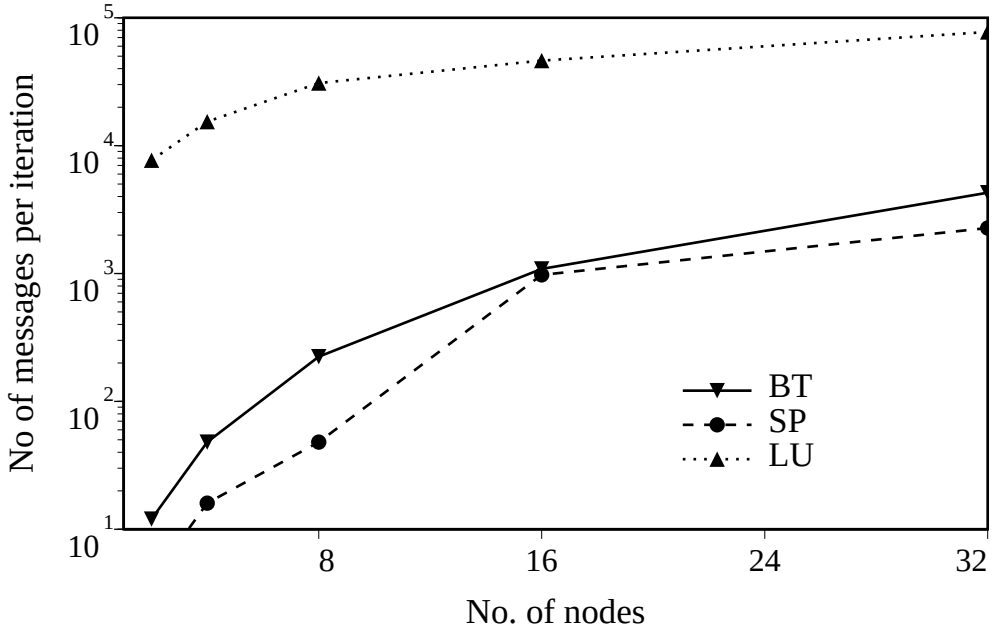


Figure 6. Total Communication Volume for NPB.

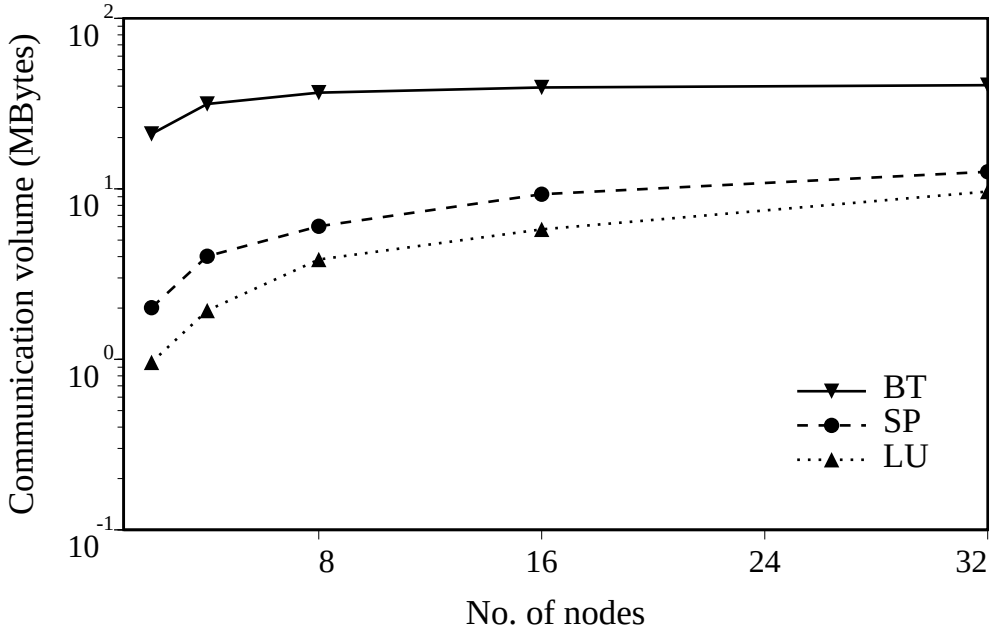


Figure 7. SP Performance.

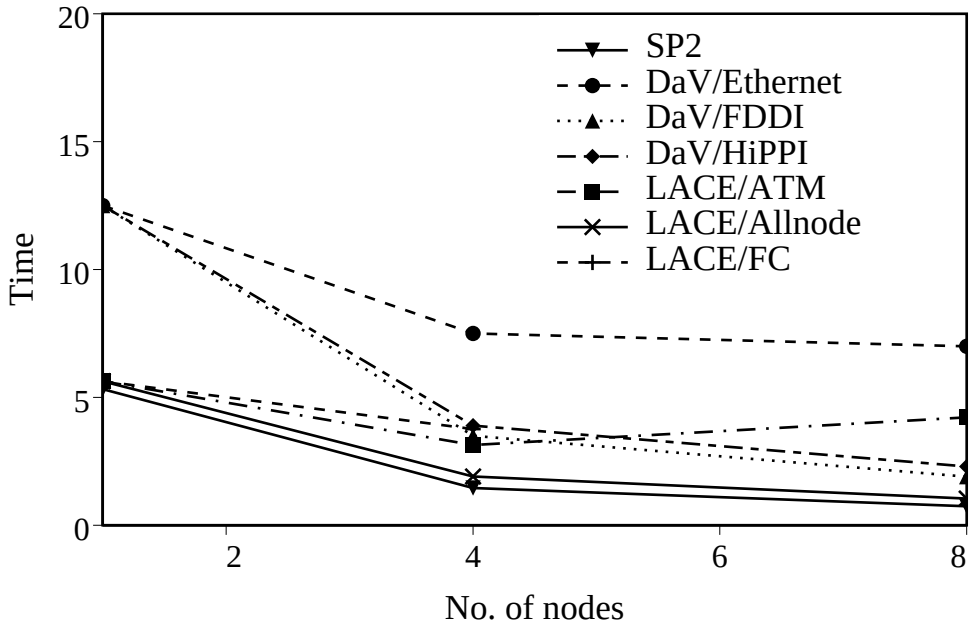


Figure 8. BT Performance.

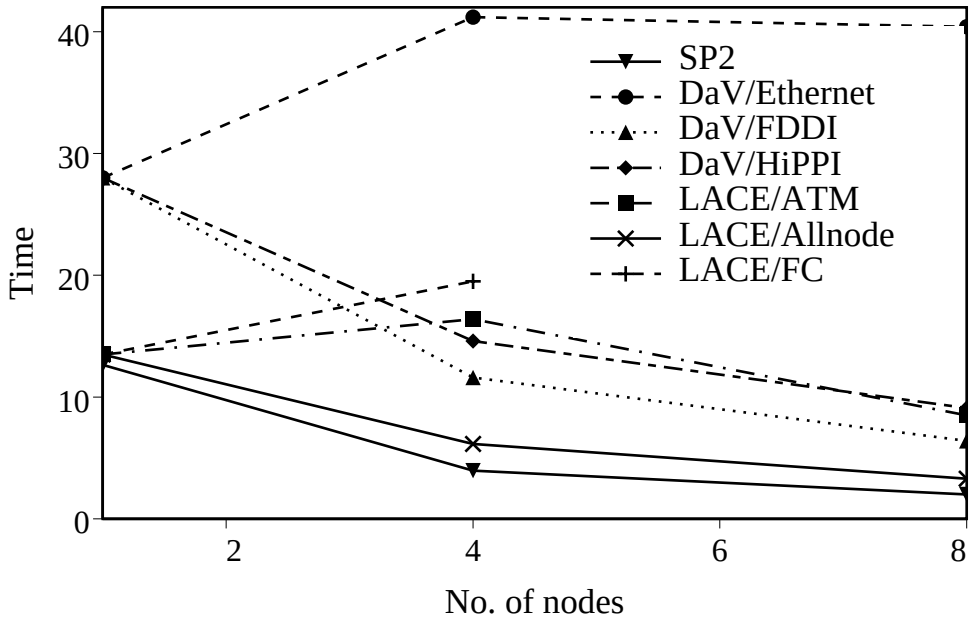


Figure 9. LU Performance

