ORNL/TM-13239

Computer Science and Mathematics Division

Mathematical Sciences Section

PERFORMANCE OF ATM/OC-12 ON THE INTEL PARAGON

Thomas H. Dunigan

Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367
thd@ornl.gov

Date Published: May 1996

Research was supported by the Office of Scientific Computing of the Office of Energy Research, U.S. Department of Energy.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Lockheed Martin Energy Research Corp.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-96OR22464

Contents

PERFORMANCE OF ATM/OC-12 ON THE INTEL PARAGON

Thomas H. Dunigan

Abstract

This report summarizes communication performance of GigaNet's OC12 ATM interface for the Intel Paragon. One-way latency of 41 μs and bandwidth of 68 MB/s (full OC12) are measured using GigaNet's AAL5 API between two Paragons. Performance is compared with Ethernet, HiPPI, and the Paragon's native message-passing facility.

1. Introduction

The Department of Energy selected Oak Ridge National Laboratory (ORNL) as one of its high performance computing centers as part of the government's High Performance Computing and Communications (HPCC) initiative. The initiative provided ORNL with funds to procure a massively parallel computer and to support various Grand Challenge applications. ORNL's Center for Computational Science operates three massively parallel Intel Paragons as part of the HPCC program. In addition to supporting research in Grand Challenge applications, the HPCC program also provides funds for improving the computational environment. This research seeks ways to better integrate parallel processors into computational science. Research areas include software tools, high-speed terabyte storage systems, computational steering and visualization, and high-speed networks. As part of the high-speed network research, ORNL participated in a joint research project to evaluate ATM interfaces for the Intel Paragon.

DOE's Sandia National Laboratory was the lead laboratory in this research project to investigate the use of ATM for interconnecting massively parallel processors. The short-term objective was to use high speed ATM interfaces to interconnect Paragons working on a single application on the exhibition floor of Supercomputing 1995. The longer term objective is to use ATM over a wide-area network to interconnect Paragons at ORNL and Sandia.

ATM is of interest for several reasons. The ATM technology can be used on a local area network, and the common carriers are endorsing it for wide-area services as well. External network services (Ethernet, HiPPI, FDDI) on the Paragons have performed poorly (high latency, low bandwidth). The proposed ATM interface promised much higher performance. Participating with Sandia were ORNL, Intel, and GigaNet, Inc. Sandia procured four OC-12 ATM interfaces from GigaNet. Initial acceptance tests were performed at Intel in September, 1995. Performance testing and PVM development was performed in the fall of 1995 at ORNL. The results of those performance tests are described in this report.

In the following section, the Paragon architecture and its communication subsystems are reviewed, then the architecture of the GigaNet ATM interface and software are summarized. Section 3 describes communication performance of the Paragon ATM system, comparing it with the Paragon's internal message-passing system and with HiPPI and Ethernet. Section 4 reviews preliminary results of two applications using multiple Paragons coupled by ATM and PVM.

2. Architecture

The Intel Paragon system [6] is a mesh-connected parallel processor. In the first member of the Paragon family, the GP system, each node on the mesh consists of two 50 MHz i860XP processors, memory, and communication hardware. One processor is used for computation, and the second processor is for communication. A Paragon MP node consists of three 50 MHz i860XP processors, memory, and communication hardware Each i860XP has its own 16 KB data and instruction cache, and each node has at least 64 MB of memory. The bus interconnecting the processors, mesh-interface, and memory operates at 400 MB/second. The 50 MHz i860XP is a super-scalar architecture capable of a peak 75 Mflops (double precision). (Figure 2.1). The GP nodes are interconnected by a 2-D mesh using the NIC-A interface with 175 MB/second communication channels and a per-hop latency of only 40 ns. The MP nodes are interconnected with the NIC-B interface and provide up to 200 MB/second (peak) communication channels. The nodes are logically subdivided into service nodes, compute nodes, and I/O nodes (Figure 2.1). The service nodes appear as a single host and support time-sharing through the OSF operating system. The compute nodes run OSF or SUNMOS. The I/O nodes are connected to local networks (Ethernet, HiPPI, and ATM) and arrays of disks (RAID) and provide a UNIX file system, swap/paging space, and a Parallel File System (PFS).

The GigaNet ATM OC-12c Protocol Engine is an MP3 I/O node (2 slots) that is attached to the Paragon mesh. The GigaNet interface supports all appropriate full duplex STS-12c/OC-12c CCITT physical layer requirements. The hardware interface has receive and transmit buffers, SAR logic, TCP/IP acceleration logic, and logic for direct access to the Paragon mesh. The direct access to the Paragon mesh permits moving data from compute nodes to/from the ATM interface without copying the data to/from the I/O node memory. For further details on the GigaNet interface see [5].

The GigaNet ATM API provides a low level, but high performance applications interface to the ATM/AAL5 layer. ATM/AAL5 is a connection-oriented best effort packet transfer service. Packets can be up to 65,536 bytes in length. The programmer is responsible for providing a queue of receive buffers, for polling to see if data is available, and for managing time-outs and re-transmissions. No operating system services are used by the API, in particular, the ATM message passing (like NX) is not part of the UNIX I/O paradigm. The API presently supports only 255 circuits per ATM interface, though a Paragon may have more than one GigaNet ATM interface. GigaNet is developing an IP over ATM service based on RFC 1577 ([9]).

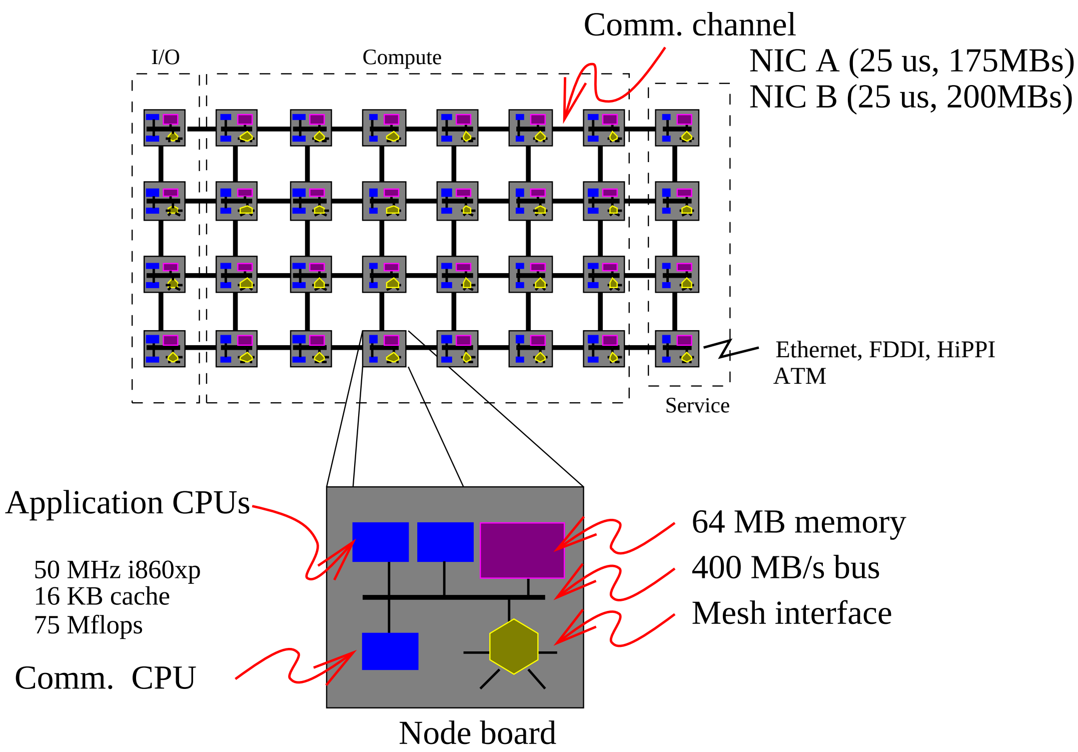

Figure 2.1: *MP node board.*

3. Performance

We ran a number of communication tests in order to establish the base-line performance of the ATM interface and to identify any bottlenecks or "bugs." We compared the ATM communication performance to the native (NX) message passing of the Paragon and to the Paragon's external network services (Ethernet and HiPPI). A more detailed analysis of the Paragon communication systems can be found in [1] and [2].

Test configurations

Our communication test suite consisted of an echo test, exchange test, and sink test. In the echo test, node A sends to node B, node B sends the message back to A. A measures the roundtrip time for N repetitions and for different size messages. In the exchange test, node A sends to B at the same time node B sends to A, both nodes receive the respective messages and repeat the sequence N times. In the sink test, node A just sends N consecutive messages to B. B just receives them as fast as it can. Both A and B time the sequence. B starts its timer when it receives the first message. The transmitter often overruns the receiver.

The transmitter's time often cannot be trusted. (The OS may be buffering the transmissions.) Unfortunately, if packets are lost, the receiver may not complete and report its times, unless there is out-of-band signaling to mark the end of the transmission. For NX, messages are never lost, but ATM can lose packets.

Message buffers were locked down to eliminate virtual memory effects. Buffer alignment can effect performance as can message length, not just the total length, but an odd number of bytes may be handled less efficiently than a multiple of two or four. We aligned buffers on 8K boundaries, this gives optimum performance on the Paragon. We tested performance with messages of varying lengths.

The tests were run at ORNL during the fall of 1995 using OSF v1.3.3 and version 1 of GigaNet's API. We tested the ATM performance using two ATM interfaces in one Paragon and between two Paragons. We tested with both NIC-A and NIC-B configurations, using both service nodes and compute nodes. Service nodes support timesharing and network services and so tests performed on those nodes show more variation and slightly lower performance. It was necessary to test ATM performance on the service nodes, because we are also interested in the performance of PVM with ATM, and the current implementation of PVM uses the service nodes for inter-host communication.

Latency/Bandwidth tests

Two key metrics of a message-passing system are bandwidth and latency. The time for a small, or zero length, message is usually bounded by the speed of the signal through the media (latency) and any software overhead in sending/receiving the message. Small message times are important in synchronization and determining optimal granularity of parallelism. If cross-country links are involved (as planned in our target ATM application), latency will be dominated by the transmission delay. So even though our computer-room tests may shows small transfer times, the applications being developed for the cross-country tests will need to account for the signal delay in crossing the country. For large messages, bandwidth is the bounding metric, usually approaching the maximum bandwidth of the media. For ATM over OC12 (622 Mbs), the maximum user-data bandwidth is 593 Mbs, or roughly 68 MBs.

Figure 3.1 shows the bandwidth between two Paragon NIC-B compute nodes using ATM and NX. Message lengths range from 4 bytes to 64K, the maximum AAL5 packet. For the sink test, the GigaNet ATM reaches full OC12 bandwidth. For 64K messages, the native message passing, NX, runs at 120 MBs. Under OSF, NX peaks at 154 MBs for 1 MB messages. The ATM echo test consumes only half the available bandwidth in one direction. The aggregate data rate for the exchange test approaches the peak ATM bandwidth. If the application has to copy data into/from message buffers, that copy will further degrade throughput as shown by the data for an echo test plus a *bcopy()*. For the Paragon, the maximum bcopy rate is 63 MBs.

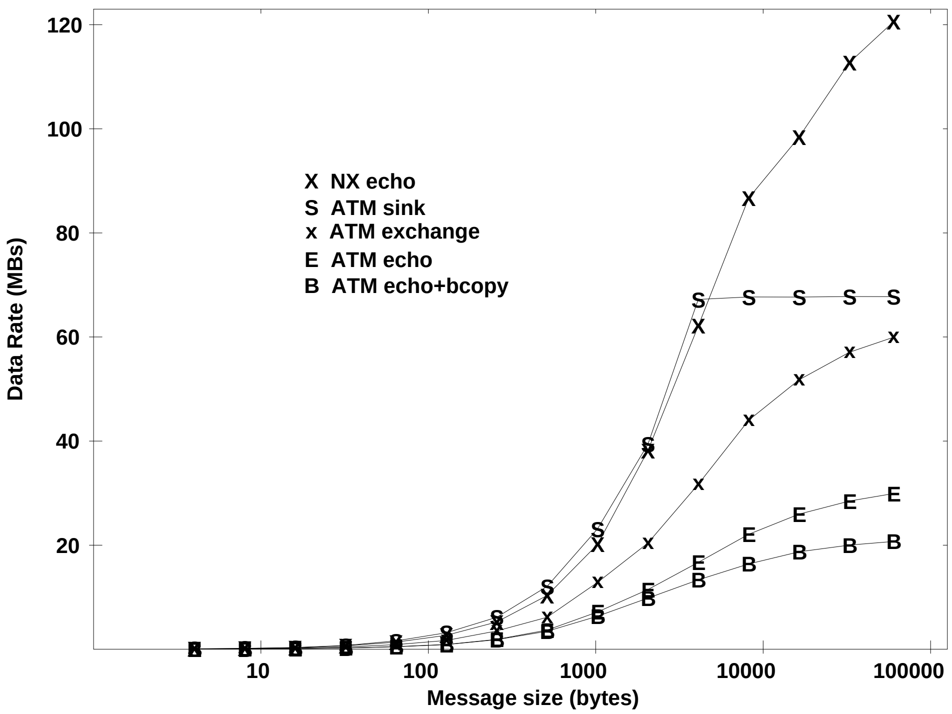

Figure 3.1: *Bandwidth for various message sizes (NIC-B).*

Figure 3.2 shows the message transfer time for various message sizes. For the back-to-back messages of the sink test, the ATM is performing at about the speed of the NX message passing. The ATM echo transfer times are about 125 μs for small messages. As noted above, the transfer time will be dominated by transmission delays for cross-country applications.

Figure 3.3 illustrates the effect of the NIC chip on communication performance. The newer Paragon MP machines have the NIC-B chip. The GP Paragons used the older NIC-A chip.

The latency or transfer times are actually averages over hundreds or thousands of iterations. Some applications may wish to attempt tight synchronization through message events. Figure 3.4 illustrates the variations in echo round-trip times between two NIC-B compute nodes over ATM when sending an 8-byte message. This same pattern is exhibited by NX message-passing too [1], and so the ATM hardware/software is not introducing any unexpected delays. The 370 μs spike occur every 10 ms as a result of timer interrupts by OSF. Most operating

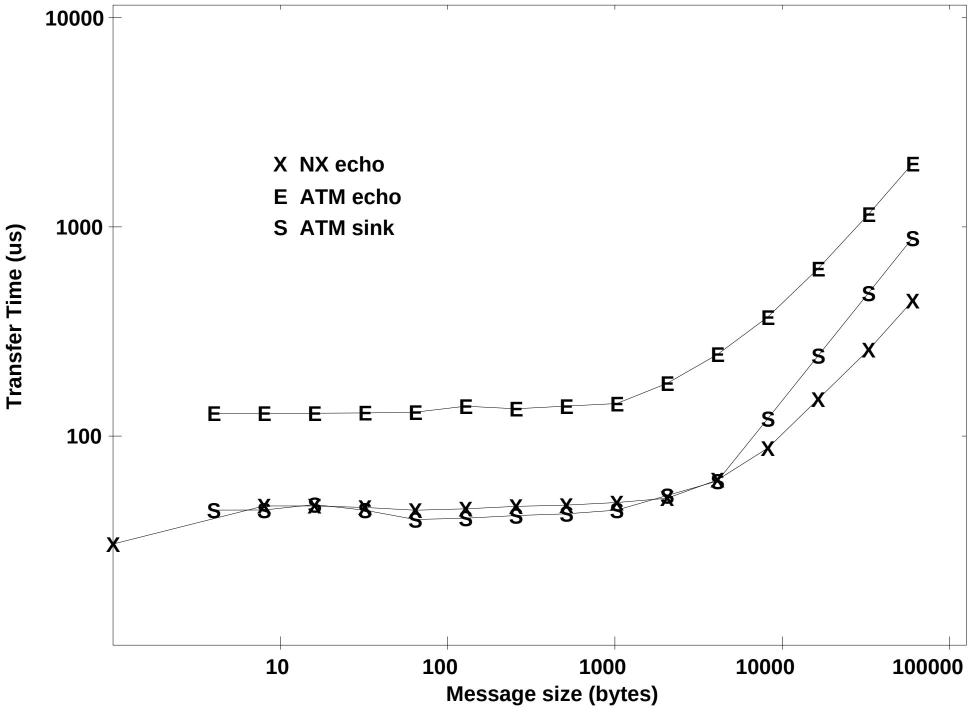

Figure 3.2: *One-way transfer time for various message sizes (NIC-B).*

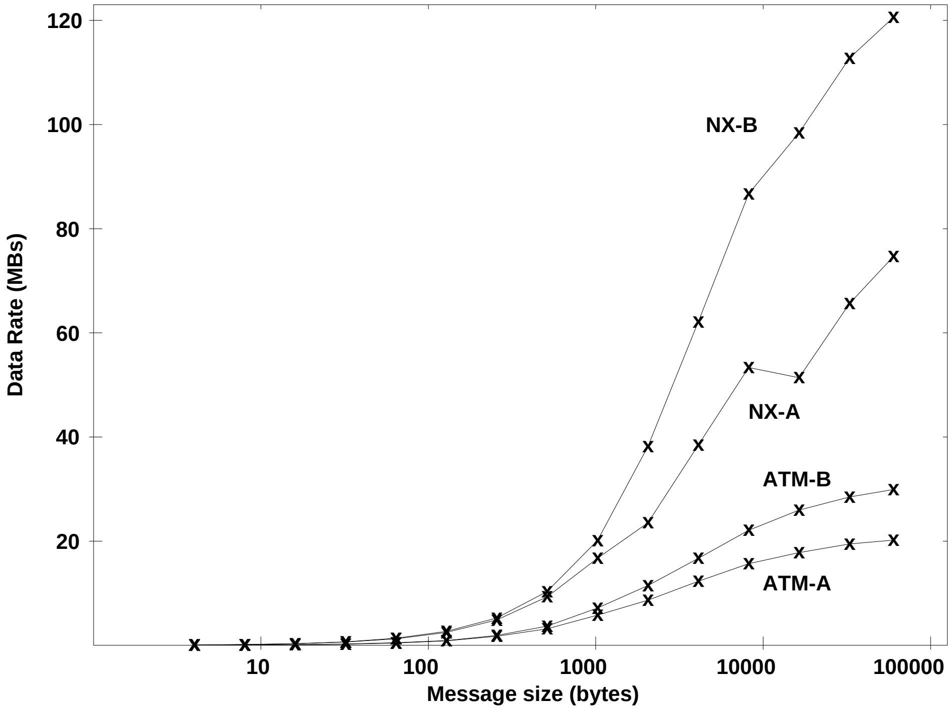

Figure 3.3: *Comparison of echo bandwidths for NX and ATM with NIC-A and NIC-B.*

systems exhibit this behavior, and make tight synchronization by message passing probablistic. Also note that for cross-country links or when using an network or operating system shared by others, jitter can be more variable.

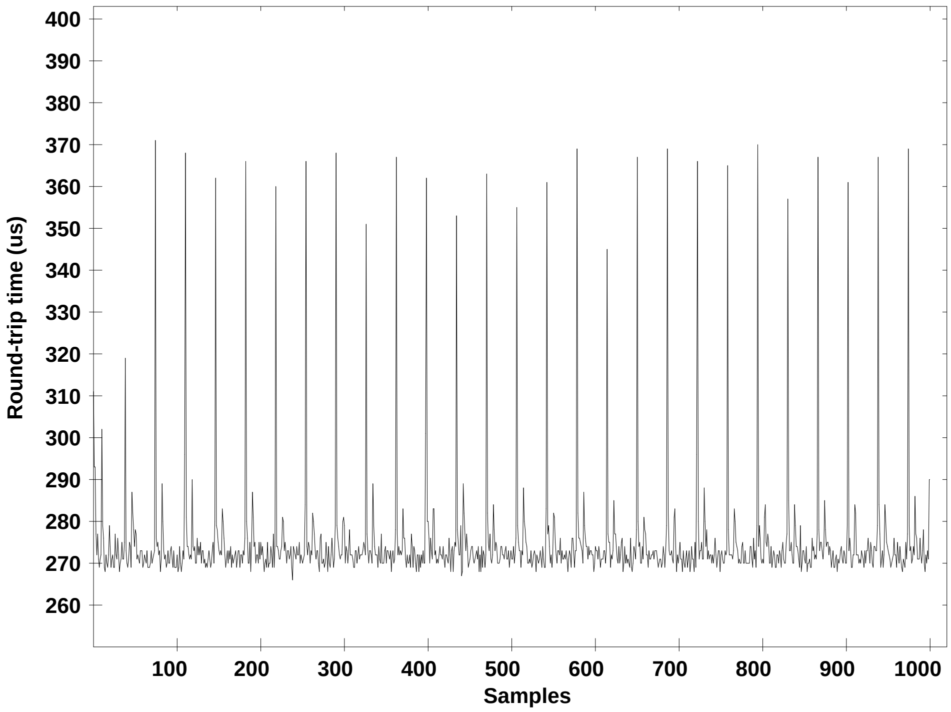

Figure 3.4: *ATM round-trip jitter for an 8-byte message.*

As noted earlier, the length of a message can have an effect on performance too. Figure 3.5 shows the variation in data rate from a portion of the ATM echo test data, where the message length is varied by 4-byte increments. Three harmonics can be observed. Performance improves when the message length is a multiple of the cache-line size (32 bytes) or the ATM cell data length (48) bytes. Performance is optimum when the message length is a multiple of both of those quantities (96 bytes).

We performed many other tests looking for anomalies, but found nothing significant. Message-passing performance would degrade by a few percent if buffers were not aligned on 8K boundaries for both NX and ATM. We tested performance between service nodes on two Paragons. As expected, message passing times showed more variations on the service nodes as the benchmarks competed with other services on the time-shared service nodes. We also tried bi-directional sink tests between two pair of nodes to see if we could achive full-duplex communication as supported by the ATM hardware. We were only able to achieve an aggregate data rate of 82 MBs out of the potential 136 MBs.

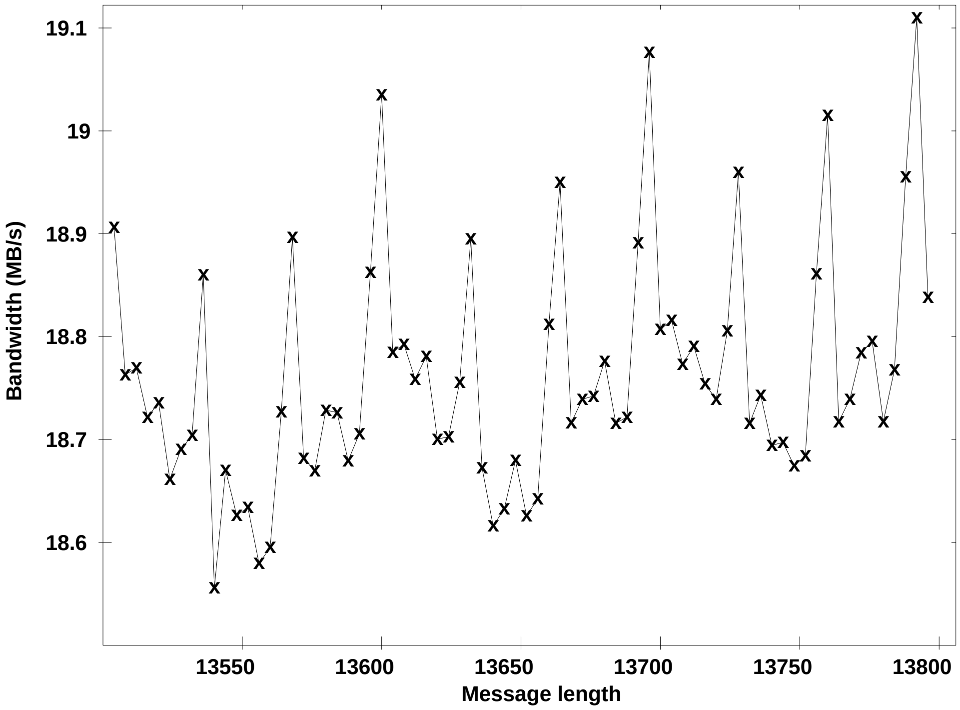

Figure 3.5: *Performance sensitivity to message length. Message size is stepped by 4 bytes.*

We also performed tests with OC12 circuits going through a FORE ATM switch. The switch added 3 μs to the latency and had no effect on bandwidth. The switch was also used to go from OC12 to OC3 and back. Going through an OC3 between the OC12 links to the Paragons added 11 μs to the latency and, of course, reduced the bandwidth. Sandia's Pratt and Jackson conducted OC3 delay and error tests using their BBN Line-line emulator. Results of these tests will be reported in another document. Preliminary results are available from *http://www.epm.ornl.gov/~dunigan/atm.*

Comparison with FDDI/Ethernet

One objective of the ATM interfaces for the Paragon was to provide a higher performance network connection than the existing HiPPI and Ethernet interfaces on the Paragon. HiPPI is a 100 MB/second data service that is typically limited to computer-room services. Although longer distances are supported by fiber, most installations are copper based and are limited to 25 meters. HiPPI is a point-to-point service, but hardware switches are available. The Paragon uses HiPPI for accessing disk farms using IPI-3 and for host-to-host communication using IP over HiPPI. The IPI-3 services are quite effective on the Paragon, the

IP services are not. Kumar [7] has measured send data rates of 96 MB/second (for a 2 MB message) and receive data rates of 65 MB/second for an application running on the HiPPI I/O node, so the board and device driver are capable of running at nearly peak rate. The receive data rate suffers from the time it takes to map kernel pages to the application. Kumar shows that if the received data is just discarded in a loopback test, then the HiPPI node can send and receive at a 191 MB/second rate.

Unfortunately, the typical Paragon application accesses the HiPPI from a compute node and not from the HiPPI node. Using a HiPPI loopback test from a compute node, we measured only 12 MB/second for a 2 MB message. This loopback test required super-user privileges to access the lower-level OSF IPC facilities. The IP performance of HiPPI between compute nodes on two Paragons performs even more poorly, as more protocol layers are added. We measured only 4 MB/second for a 1 MB message. The test used TCP with a window size of 256 KB. IP over Ethernet between Paragons also performs poorly. Ethernet provides peak bandwidth of 1.2 MB/second, and most workstations provide at least 900 KB/second using TCP/IP. Our Paragon test of TCP/IP over Ethernet measured only 407 KB/second. Figure 3.6 compares the Paragon data rates for ATM, Ethernet, and HiPPI. Note that the ATM data rates are for AAL5, the IP over ATM implementation was still under development in the fall of 1995. From the figure, the ATM implementation provides a clear performance advantage for interconnecting Paragons.

The time to transmit a small message is also quite poor on the Paragons for UDP over Ethernet or HiPPI. Transfer times for typical workstations are on the order of 300 to 500 microseconds for UDP over Ethernet. For the Paragons, transfer time was 9 milliseconds for Ethernet and 12 milliseconds for HiPPI. Compare this with the 140 microsecond transfer time that we measured for ATM. The Paragon IP implementation requires transferring messages to and from the compute nodes with the network I/O node using OSF's NORMA IPC facility. The IPC facility, in turn, sits atop the NX message passing facilities. We measured the NORMA IPC performance between two compute nodes for both NIC-A and NIC-B implementations (Figure 3.7). Clearly, NORMA IPC will prevent a compute node from reaching peak HiPPI speeds.

4. Summary

Our initial performance measurements have shown that the GigaNet ATM hardware and software provides better performance than either Ethernet or HiPPI

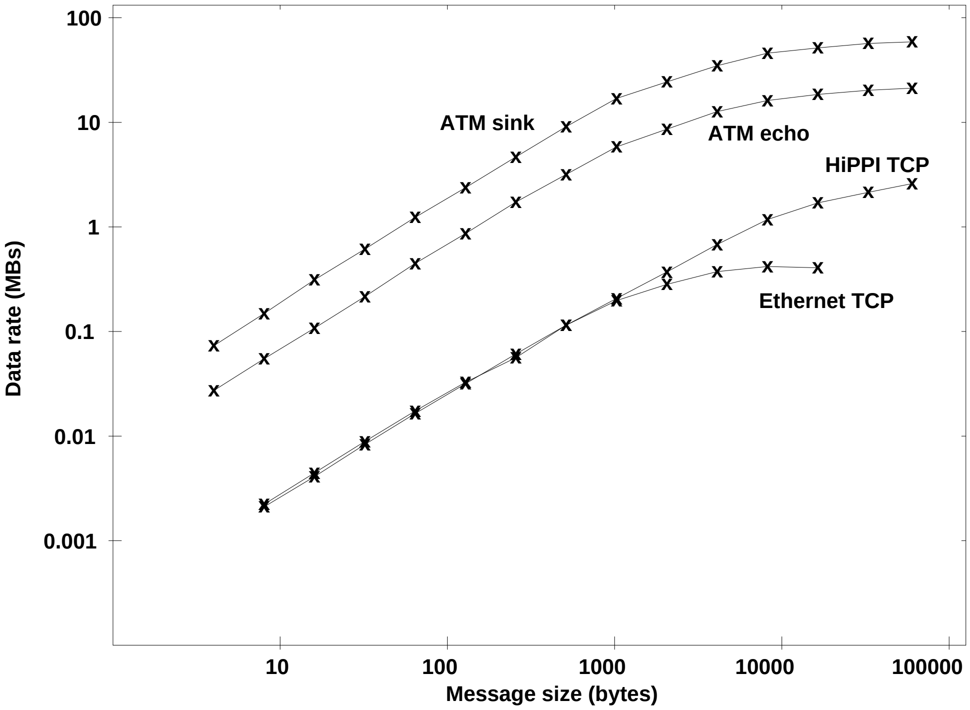

Figure 3.6: *Data rates for various messages sizes.*

on the Intel Paragons (see Table 4.1). Our ultimate objective is to use the ATM to interconnect Paragons to solve computing problems that require massive computing resources. A preliminary step toward that goal was to implement several parallel applications using Paragons coupled by PVM over ATM and demonstrate them at the December Supercomputing '95 conference.

PVM, Parallel Virtual Machine [4], was chosen because the applications could be ported and tested over different network technologies as the development and testing of the Paragon ATM software and hardware progressed. PVM is a general message-passing subsystem that uses IP to interconnect heterogeneous workstations and supercomputers. PVM's portability and flexibility make it a popular platform for implementing parallel applications. Since we did not have a working IP-over-ATM implementation, PVM had to be extended to include the GigaNet AAL5 API. (As noted earlier, IP implementations often suffer from poor performance, so an AAL5 API may be desirable even when an IP-over-ATM implementation becomes available.)

Two applications were implemented with PVM over ATM. The first code was from ORNL's first-principles simulation of materials properties Grand Challenge. The code was run using two Paragons over Ethernet and then over ATM. The code ran 18% faster over ATM, and measurements indicated that PVM/ATM

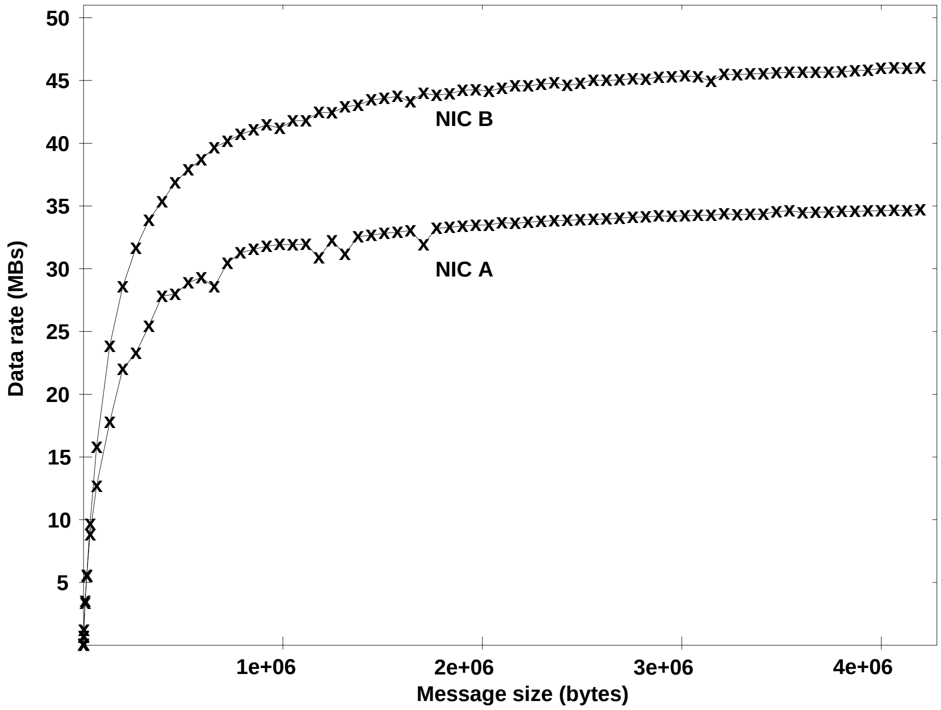

Figure 3.7: *NORMA IPC data rates for various messages sizes.*

had a latency of about 2 ms and a bandwidth of only 2 MBs. This modest performance indicated a number of optimizations were required in PVM. Sandia's solid dynamics code (PCTH) was also run between Paragons over ATM on the floor of Supercomputing 95. The PCTH performance was four times slower than a dedicated calculation on one Paragon, but proved that a run between Paragons across the country was feasible [8].

Finally, the Paragon ATM interface was also used in doing some preliminary video data processing from an AT&T EMMI. The EMMI provided an ATM/OC3 full motion video stream that the Paragon stored, played-back, and encrypted. Some software traffic shaping (flow control at the ATM cell level) was required when the Paragon was generating the video data stream so as not to overrun the EMII.

Acknowledgements

The principal investigators for this project are Steve Gossage and Tom Pratt from Sandia. Sandia provided most of the funding for the equipment. ORNL's Center for Computational Science provided a testbed of Paragon with technical support from Intel's Peter Molnar. GigaNet's Rich Prohaska was extremely helpful. ORNL's Phil Papadopoulos modified PVM to support ATM.

Transfer time and bandwidth		
	time (us)	MB/s
NIC A		
NX	36	90
ATM echo	140	40
ATM sink	56	58
NIC B		
NX	25	171
ATM echo	125	30
ATM sink	41	68
HiPPI TCP	12000	4
Ethernet TCP	9000	0.4

Table 4.1: *Small-message transfer time and large-message bandwidth.*

Many others participated and are listed on the Web site that was used as the "living" lab journal for the testing. Please visit *http://www.epm.ornl.gov/~dunigan/atm.* Also, GigaNet has a report summarizing these tests [3].

5. References

[1] T. H. Dunigan. Early experiences and performance of the Intel Paragon. Technical report, Oak Ridge National Laboratory, 1993. ORNL/TM-12194.

[2] T. H. Dunigan. Beta testing the intel paragon mp. Technical report, Oak Ridge National Laboratory, 1995. ORNL/TM-12830.

[3] D. Follett, M. Gutierrez, and R. Prohaska. A high performance ATM protocol engine for the Intel Paragon, 1996. http://www.tiac.net/users/giganet/white-paper.html.

[4] G. A. Geist and V. S. Sunderam. Network Based Concurrent Computing on the PVM System. Technical report, Oak Ridge National Laboratory, June 1991. ORNL/TM-11760.

[5] GigaNet Inc. GigaNet ATM OC-12c Protocol Engine, 1995. http://www.tiac.net/users/giganet/product.html.

[6] Intel. *Paragon XP/S Product Overview.* Intel, Beaverton, Oregon, 1991.

[7] V. Kumar. A host interface architecture for hippi. In *SHPCC 1994*, pages 142–149, 1994.

[8] Sandia National Lab. PCTH - a massively parallel Eulerian solid mechanics code, 1996. http://www.sandia.gov/1431/PCTHwdoc.

[9] M. Laubach. Classical IP and ARP over ATM, 1994. http://www.cis.ohio-state.edu/htbin/rfc/rfc1577.html.