Operating System/ Workstation Platform	Network	TCP/IP Round-trip Time (μs)	Measured Round-trip Time for 8 B Message (μs)	Peak UDP/ IP Receiver Throughput (Mbits/sec)	UDP/IP Half-Power Point (Bytes)
Solaris 2.4 on SS-20	Ethernet: AMD Lance Ethernet NI and Bay Networks Switched Ethernet	1857 + 2.02n (R^2=0.974)	1031	9.6	126
Solaris 2.4 on SS-20	ATM: SunATM-155 NI and Bay Networks LattisCell Switch	1317 + 0.26n (R^2=0.998)	1284	83.5	2031
Solaris 2.4 on SS-20	ATM: Fore SBA-200 NI and ASX-200 Switch	1634 + 0.26n (R^2=0.992)	1517	79.6	2421
Solaris 2.4 on SS-20	Myricom Myrinet	1614 + 0.31n (R^2=0.993)	1502	78.7	2922
HP-UX 9.01 on HP 9000-735	Intel Ethernet	1340 + 2.08n (R^2=0.981)	633	8.9	87
HP-UX 9.01 on HP 9000-735	HP Medusa NI and FDDI	462 + 0.27n (R^2=0.983)	396	82.8	3450

TABLE 9. Round-trip delay and peak delivered throughput for a variety of commercial local area network environments.

9256-byte Myrinet MTU. Figure 4b shows the switched Ethernet, with a change in slope occurring at roughly the 1500-byte MTU size for Ethernet.

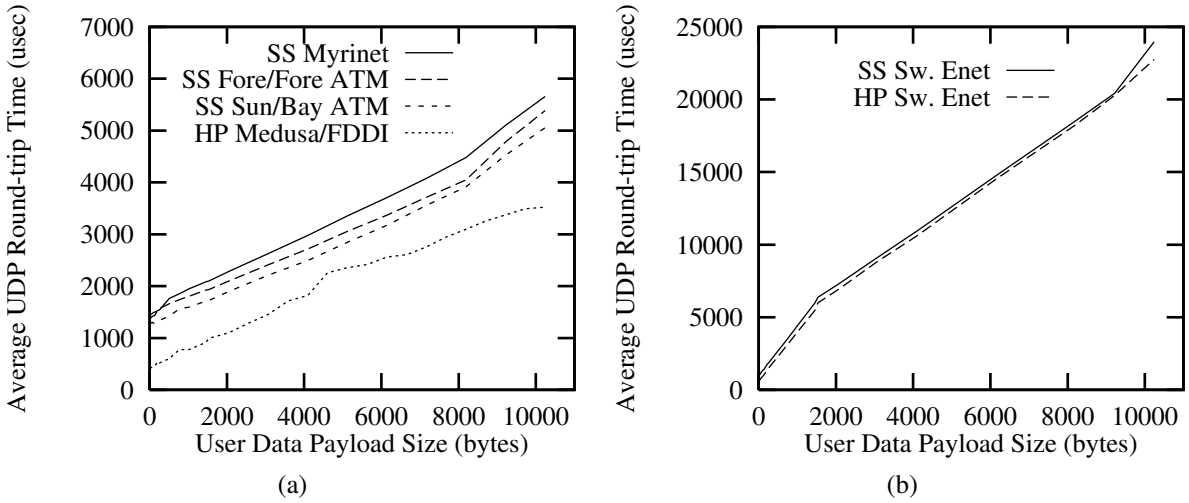


FIGURE 4. Round-trip overheads for actual networks. (a) High-speed networks. (b) Switched Ethernet networks.

Figure 5 presents the "out-of-the-box" throughput for the Solaris-based network platforms. For all three of the high speed (i.e., ATM or Myrinet) networks, throughput is limited to 40 Mbits/sec or lower. These throughputs are lower than expected, given that previous measurements of the same hardware running Solaris 2.3 produced throughputs in the 70 to 80 Mbits/sec range. We have received confirmation from Sun that this is a known performance problem in the networking subsystem of the Solaris 2.4 kernel. A fix is anticipated in the next release of Solaris.

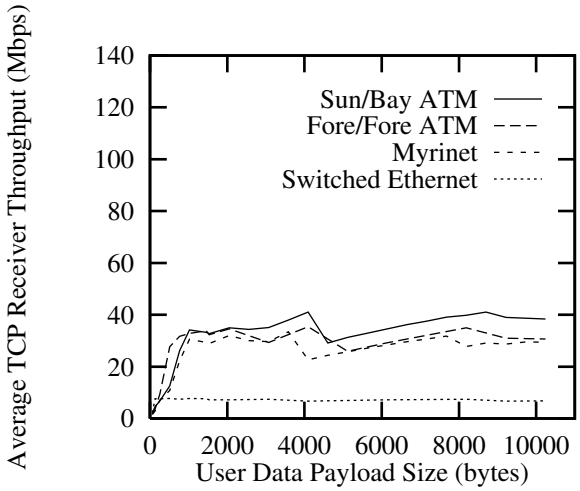


FIGURE 5. TCP/IP received throughputs for Solaris 2.4-based networks with default (8 KB) TCP/IP window sizes.

[Wolman94] Alec Wolman, Geoff Voelker, and Chandramohan A. Thekkath. “Latency Analysis of TCP on an ATM Network,” *Proceedings of the 1994 Winter USENIX Conference*, San Francisco, CA, January 17-21, 1994, pp. 167 - 179.

9 Appendix

Operating System/ Workstation Platform	**SPECInt92 Rating**[a]	**TCP/IP Round-Trip Overheads** (µs)	**Measured Round-Trip Time for 8 B Message** (µs)
Solaris 2.4 on SparcStation 20	73.7	615 + 0.13n (R^2= 0.996)	626
HPUX-9.01 on HP 9000-735[b]	109.1	249 + 0.12n (R^2=0.990)	258

TABLE 8. Protocol processing overheads for loopback interface.

a. These SPECInt92 ratings are for uniprocessor machines[DiMarco95]. Our Solaris 2.4 experiments were performed on a dual processor machine.

b. This experiment uses the standard Internet protocol stack, while the results listed in Table 5 use a special single-copy Internet protocol stack.

Figure 4a shows the round-trip times for Myrinet, Fore Systems ATM and Sun/Bay Networks ATM on the Solaris 2.4 SS-20 platform, as well as the Medusa/FDDI network on the HP-UX 9.01 HP 9000-735 platform. (It should be noted that the HP Medusa/FDDI exchanges utilize a single-copy protocol stack. This stack differ from the traditional protocol stack in that it takes advantage of the Medusa NI’s IP checksum unit and onboard memory to make only one copy of the data [Banks93].) Figure 4b shows the performance of Switched Ethernet for both Solaris and the traditional HP-UX protocol stack.

Both graphs show nearly linear round-trip times, with a handful of exceptions due to the network device driver’s maximum transfer unit (MTU), a value chosen to specify segmentation and reassembly between the IP layer and the device driver. In Figure 4a, we can see the effects of the 4200-byte Medusa MTU, the 9180-byte ATM MTU, and the

[Boden95] Nanette J. Boden, Danny Cohen, Robert E. Felderman, Alan E. Kulawik, Charles L. Seitz, Jakov N. Seizovic, and Wen-King Su. “Myrinet: A Gigabit-per-Second Local Area Network,” *IEEE Micro*, 15:1, February 1995, pp. 29 - 36.

[Clark89] David D. Clark, Van Jacobson, John Romkey, and Howard Salwen. “An Analysis of TCP Processing Overhead,” *IEEE Communications Magazine*, June 1989, pp. 23 - 29.

[Culler93] David Culler, Richard Karp, David Patterson, Abhijit Sahay, Klaus Erik Schauser, Eunice Santos, Ramesh Subramonian, and Thorsten von Eicken. “LogP: Towards a Realistic Model of Parallel Computation,” *Proceedings of PPOPP*, 1993.

[dePrycker93] Martin de Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Second Edition, Ellis Horwood Publishers, 1993.

[DiMarco95] John DiMarco. “Spectable,” file://ftp.cdf.toronto.edu/pub/spectable, February 1995.

[Gray93] Jim Gray, editor. *The Benchmark Handbook for Database and Transaction Processing Systems*, Second Edition, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1993.

[Gusella90] Riccardo Gusella. “A Measurement Study of Diskless Workstation Traffic on an Ethernet, “*IEEE Transactions on Communications*, 38:9, September 1990, pp. 1557 - 1568.

[Jacobson92] Van Jacobson. “Design Changes to the Kernel Network Architecture for 4.4 BSD,” 4.4 BSD Class, Berkeley, CA, May 1992.

[Kay93] Jonathan Kay and Joseph Pasquale. “The Importance of Non-Data Touching Processing Overheads in TCP/IP,” *Proceedings of SIGCOMM ‘93*, San Francisco, CA, September 13 - 17, 1993, pp. 259 - 268.

[Martin94] Richard Martin. “HPAM: An Active Message Layer for a Network of HP Workstations,” *Proceedings of Hot Interconnects II*, Stanford, CA, August 11 - 13, 1994, pp. 40 - 58.

[Martin95] Richard Martin. Personal communication, August 1995.

[Partridge93] Craig Partridge and Stephen Pink. “A Faster UDP,” *IEEE/ACM Transactions on Networking,* 1:4, August 1993, pp. 429 - 440.

[Patterson90] David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, 1990, pp. 40 - 42.

[Reilly94] Jeff Reilly and Reinhold Weicker. “A Summary of the SPEC Benchmark Suites,” *SPEC Newsletter*, 6:1, March 1994, pp. 3 - 5.

[Rodrigues95] Steven Rodrigues, Kimberly Keeton and Thomas Anderson. “Fast Sockets: Putting the ‘Fast’ Back into Fast Networks,” submitted for publication.

[Steenkiste94] Peter A. Steenkiste. “A Systematic Approach to Host Interface Design for High-Speed Networks,” *IEEE Computer*, 27:3, March 1994, pp. 47 - 57.

[Stevens94] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

and Myrinet have higher per-message overheads than Ethernet, it is to be expected that performance on these "high-speed" networks should be worse than performance on Ethernet.

5 Conclusions

Recent years have seen a transition from shared media local area networks to switch-based local area networks, which provide much higher aggregate bandwidth and better behavior under load. The industry has focused on link bandwidth as the most important figure of merit, while other factors affecting communication performance, such as overhead and latency, have been mostly ignored. These factors play an important role, however, in determining network performance, especially for request-response style communication with small messages.

We have developed a portable, user-level tool to measure various components of communication cost. Our measurements of overhead, latency and throughput indicate that, while link and aggregate network bandwidth are increasing, software overheads are also increasing. This increase in overhead has a significant impact on end-user performance for applications such as NFS file service and online transaction processing, which commonly perform request-response communication using small messages. In particular, we saw that application performance did not improve when the bandwidth was increased.

What are the implications of these findings for network designers? An important class of applications will not be able to take advantage of increases in network bandwidth unless host protocol software overheads are lowered. Several techniques for providing lower overheads have already been successfully explored. Further significant performance increases are also possible if we are willing to investigate alternate protocol stacks, such as the active message-based Fast Sockets or U-Net, for use in the local area. To see improvements in end-user application performance, however, these experimental techniques must be incorporated into commercial networking products.

6 Future Work

Our analysis has focused on point-to-point performance on unloaded networks, without exploring the effects of contention. As stated earlier, this is an important factor in determining communication performance in networks that support many users. Our initial measurements of behavior under contention, using several simultaneous point-to-point conversations, have been unable to produce a degradation in performance (i.e., either delivered throughput or round-trip time) with respect to point-to-point performance. We believe that this is due to the high overheads to send and receive messages, which prevent us from generating high enough load to cause contention. Initial experiments using many-to-one and one-to-many communication patterns to explore endpoint contention do show a degradation in performance, as expected. However, high overheads again limit our ability to generate load. These studies are the subject of ongoing work.

7 Acknowledgments

The authors would like to thank David Culler, Jim Gray, Rich Martin and Steve Rodrigues for many insightful discussions about the analysis of communication performance. Eric Anderson, Mike Dahlin, Jim Gray Bruce Mah, Alan Mainwaring and Rich Martin provided useful comments on drafts of this paper. Bruce Mah co-authored the network benchmark program described in Section 3.2. Rich Martin provided valuable information regarding the operation of the Myrinet and Medusa/FDDI networks.

8 References

[Banks93] David Banks and Michael Prudence. "A High-Performance Network Architecture for a PA-RISC Workstation," *IEEE Journal on Selected Areas in Communications*, 11:2, February 1993, pp. 191 - 202.

[Basu95] Anindya Basu, Vineet Buch, Werner Vogels, and Thorsten von Eicken. "U-Net: A User-Level Network Interface for Paralel and Distributed Computing," Computer Science Technical Report "to appear", May 1995.

Processor/ Operating System	Network	Average NFS Response Time (ms)	Ratio of Link Bandwidths wrt Ethernet	Ratio of UDP 8-byte Round-Trip Delays wrt Ethernet
Solaris 2.4 on SS-20	AMD Lance Ethernet and Bay Networks EtherCell Switch	14.49	1	1.00
Solaris 2.4 on SS-20	Fore Systems SBA-200 NI and ASX-200 ATM Switch	11.83	15	1.36
Solaris 2.4 on SS-20	Myricom Myrinet	13.34	64	1.43

TABLE 6. Average NFS server response time.

sec Myrinet. These meager response time improvements are due to several factors. Most of the messages sent by NFS are part of request-response exchanges, and they are very short: 68% of the messages in the experiment are less than 210 bytes long. Additionally, 23% of the messages are roughly 8 KB long, and 8% of the messages are roughly 1 KB long. The combination of request-response communication and large percentage of small messages makes overhead the limiting factor for NFS communication performance. Because both Myrinet and Fore ATM have higher per-message round-trip overheads than Ethernet, it is not surprising that they see only a limited response time improvement. This limited improvement results from the additional bandwidth Myrinet and ATM provide to transfer large and medium-sized messages, which comprise roughly 30% of the overall messages transferred.

4.2 Transaction Processing Environment

Online transaction processing database systems are another important class of applications. In these environments, clients communicate in a request-response fashion to perform transactions on the database stored at the server The transaction processing workload is a DebitCredit benchmark, similar to TPC-B, where transactions are performed in a banking environment [Gray93]. Each transaction must update bank account, branch, and teller records, as well as update a history file. The performance metric is the number of committed transactions, subject to a residence time constraint that 90% of the transactions take less than two seconds. We ran this workload with a commercial shared memory database. The database server machine was loaded with 256 Mbytes of memory to allow the entire database to reside in main memory: none of the transactions should access disk. As with the NFS experiments, we used a single client load generating process to avoid queueing delays at the server. Each transaction generates four round-trip exchanges with TCP/IP messages of less than 200 bytes between the client and server machines. Our results, which are the average of four independent repetitions, are shown in Table 7.

Processor/ Operating System	Network	Average Transactions per Second (tps)	Ratio of Link Bandwidths wrt Ethernet	Ratio of TCP 8-Byte Round-Trip Delays wrt Ethernet
Solaris 2.4 on SS-20	AMD Lance Ethernet and Bay Networks EtherCell Switch	76.95	1	1.00
Solaris 2.4 on SS-20	Fore Systems SBA-200 NI and ASX-200 ATM Switch	67.10	15	1.47
Solaris 2.4 on SS-20	Myricom Myrinet	65.68	64	1.46

TABLE 7. Average DebitCredit transactions per second.

We see that the transaction rates for both ATM and Myrinet are worse than that for Ethernet. The reasons for this performance degradation are that the DebitCredit benchmark uses only request-response communication with very small (i.e., less than 200 bytes) messages; hence, this benchmark is also overhead limited. Again, because both Fore ATM

Ethernet. One explanation for this increase in overhead is that the Ethernet hardware and software are highly tuned due to their maturity, while the ATM and Myrinet hardware and software are first (or second) generation products, and have not yet reached maturity.

From Figure 3 and Table 5, we also note that even with the communication-intensive one-way transfer experiment, the hosts are able to drive only roughly half of the 155 Mbits/sec link bandwidth for ATM. They don't come close to driving the 640 Mbits/sec Myrinet bandwidth. Throughput is limited by the rate at which the sender can send messages into the network, due to the large start-up overheads of sending a message.

Because overheads to send and receive messages are high, these costs must be amortized over many bytes before reasonable throughput can be obtained. ATM networks must send relatively large (e.g., 1KB to 2KB) messages in order to obtain half of the peak delivered throughput. Myrinet must send even larger (e.g., nearly 3KB) messages to see reasonable throughput. These half-power points are much higher than the typical message sizes observed by Kay and Pasquale on an FDDI LAN (i.e., less than 200 bytes) [Kay93].

The operating system implementation of the protocol stack plays an important role in determining the overhead incurred by the sending and receiving hosts. There has been considerable research in reducing both per-message and per-byte costs associated with the Internet protocols [Steenkiste94, Partridge93, Jacobson92]. Some of these optimizations are being utilized in new commercial operating systems and network interfaces. For instance, Table 5 shows results for the HP-UX Medusa single-copy protocol stack, which takes advantage of the buffer memory on the Medusa card to eliminate several copy operations. The socket layer copies data directly from the user-level application buffers to the NI buffers, checksumming the data on the fly. If we look at the round-trip overheads for sending an 8-byte message on the Medusa/FDDI network in Figure 4b, we see that these optimizations result in a 250-microsecond, or 38% decrease in the round-trip overhead, when compared with that of Switched Ethernet. In addition, Figure 3 and illustrates that the Medusa single-copy stack is able to drive nearly all of the 100 Mbits/sec FDDI aggregate bandwidth.

To see further improvements in overheads, more aggressive approaches have been explored. As one example, U-Net implements a user-level version of TCP/IP on top of an ATM workstation cluster [Basu95]. They achieve round-trip times of about 200 µsec, and throughputs of 120 Mbits/sec. Another approach is Fast Sockets [Rodrigues 95], which provides the traditional stream and datagram socket interfaces, but implemented on top of a light-weight active message layer built for a network of HP workstations connected via Myrinet [Martin94]. Fast Sockets achieve a 65µsec round-trip time for small messages, and a high throughput: 92 Mbits/sec.

4 Effects of Network Performance on Application Performance

Section 3.2 demonstrated that the protocol overheads comprise the majority of communication costs, especially for request-response style communication and communication using small messages. Furthermore, the overheads associated with the network interface software are increasing for higher bandwidth networks, compared to the more highly tuned Ethernet overheads. Both of these factors can have a serious impact on the performance of applications, as we show in this section, by presenting the performance of NFS and a transaction processing database environment over Ethernet, ATM, and Myrinet.

4.1 NFS-based File Systems

The workload for our NFS performance measurements was an NFS workload based on an abstraction of actual NFS operation mixes and operation request rates. This workload was modeled after the laddis benchmark from the SPEC System-level File Server suite [Reilly94]. The benchmark measures NFS server and network performance in terms of NFS response time, as a function of the load offered by one or more clients. Because client caching is disabled for the benchmark, each NFS request must be forwarded over the network to the server. To avoid queueing delays at the server, we used a single client load generating machine; the client offered a relatively low load of ten requests per second. The performance metric, NFS response time, includes the client NFS and network protocol overheads, the network latency, the server NFS and network protocol processing, and the server disk access time. Our results, which are the average of four independent repetitions, are shown in Table 6.

From Table 6, we see that the increase in bandwidth from the 10 Mbits/sec offered by Ethernet to the 155 Mbits/sec offered by ATM results in a 18% improvement in NFS response. We see a smaller improvement (8%) for 640 Mbits/

3.3.4 Local Area Network Overheads

If we examine the UDP/IP loopback round-trip times (in Figure 1 and Table 2) and round-trip times for actual networks (in Figure 2 and Table 4), we see that the loopback round-trip times account for only about half of the round-trip times seen for actual networks. These differences are due to two factors. First, the device drivers were different. For the results presented in Table 2, the loopback device driver is measured, whereas in Table 4 the device driver for the corresponding network is measured (i.e., the Ethernet driver or the ATM driver). Second, the round-trip latency through the switch is included in the data in Table 4. To see the overheads for actual networks, we can subtract the round-trip latency through the switches from the overall round-trip latencies seen. These overhead results are shown in Table 5.

Operating System/ Workstation Platform	**Network**	**UDP/IP Round-trip Overheads (μs)**	**Per-Message Integer Instruction Issue Opportunities**	**Per-Byte Integer Instruction Issue Opportunities**
Solaris 2.4 on SS-20	Ethernet: AMD Lance Ethernet NI and Bay Networks EtherCell	1534 + 2.18n (R^2=0.979)	153,400	218
Solaris 2.4 on SS-20	ATM: SunATM-155 NI and Bay Networks LattisCell Switch	1053 + 0.32n (R^2=0.999)	105,300	32
Solaris 2.4 on SS-20	ATM: Fore SBA-200 NI and ASX-200 Switch	1406 + 0.32n (R^2=0.998)	140,600	32
Solaris 2.4 on SS-20	Myricom Myrinet	1490 + 0.36n (R^2=0.997)	148,900	36
HP-UX 9.01 on HP 9000-735	Intel Ethernet	1226 + 2.18n (R^2=0.981)	121,400	216
HP-UX 9.01 on HP 9000-735	HP Medusa NI and FDDI	446 + 0.34n (R^2=0.992)	44,100	33

TABLE 5. Round-trip overheads for a variety of commercial local area network environments.

Table 5 also calculates the number of integer instruction issue opportunities that are lost during the round-trip overheads for each network. These figures were calculated by multiplying the appropriate linear regression model parameter, the processor clock rate (e.g., 50 MHz for the SS-20 and 99 MHz for the HP 9000/735), and the number of integer instruction issue opportunities per cycle (e.g., two for the SS-20 and 1 for the HP 9000/735).

3.4 Discussion of Results

The results in Section 3.3 show us that although network vendors have been successful at increasing the bandwidth delivered to the user beyond the 10 Mbits/sec offered by Ethernet, they have paid little attention to end-host overhead. We see that protocol software overheads remain very high: during the time spent in overheads required for a request-response exchange, Table 5 shows that 100,000 to 150,000 integer instructions could have been issued!

Table 2, Table 3, and Table 5 indicate that the round-trip overheads associated with UDP/IP clearly dominate the switch latencies (e.g., by as much as nearly three orders of magnitude for the Myrinet environment). UDP/IP per-message round-trip times are on the order of 500 to 1400 μsec, while switch latencies range from only a few μsec to a few hundred μsec.

Furthermore, if we examine the UDP/IP round-trip time for Ethernet, ATM, and Myrinet on a SparcStation 20 in Figure 2a, we note that the per-message component of the round-trip overhead increases as the bandwidth of the network increases. This is due predominantly to increases in device driver software and network interface hardware overheads. If we more closely examine the round-trip times for sending an 8-byte message on Myrinet, for example, we note that there is a 536-microsecond, or 59%, increase in the round-trip overhead, when compared to that of

than 40 Mbits/sec. (See Figure 5 in Section 9 for more detailed measurements.) For the remainder of the experiments described in this paper, we kept the Solaris 2.4 TCP window sizes at 64 KB. Figure 3b shows receiver throughput for HP-UX networks.

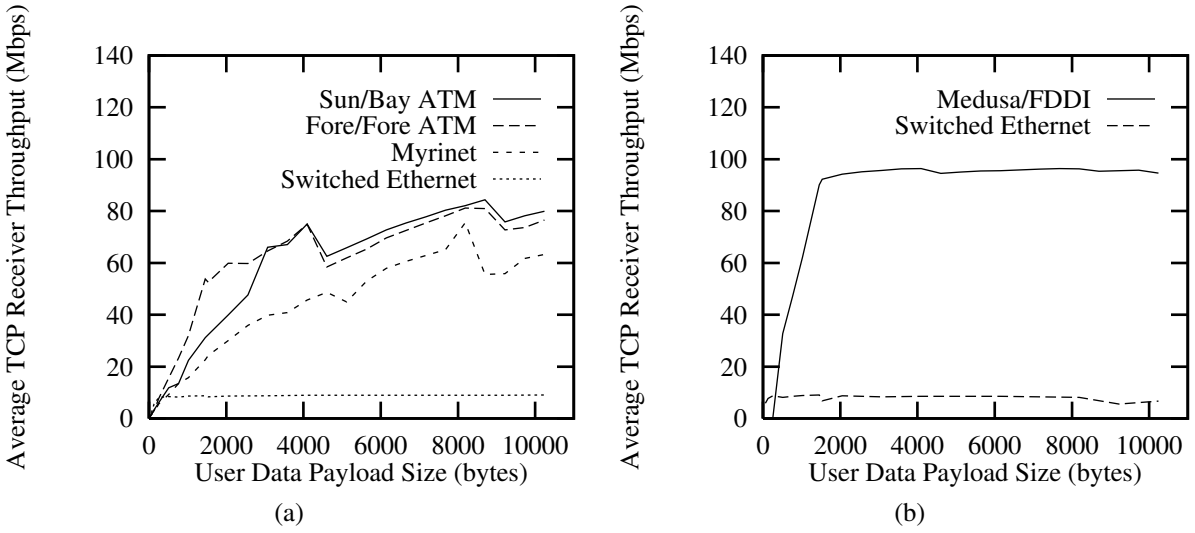


FIGURE 3. TCP/IP received throughputs for several local area network environments. (a) Solaris 2.4-based networks with 64 KB TCP/IP window sizes. (b) HP-UX 9.01-based networks.

Per-message and per-byte components of the round-trip time were obtained by performing simple linear regression on the results of round-trip tests for packet sizes less than 8 KB. These results, as well as the peak bandwidth and the half-power point, are shown in Table 4.

Operating System/ Workstation Platform	**Network**	**UDP/IP Round-trip Time (μs)**	**Measured UDP Round-trip Time for 8 B Message (μs)**	**Peak TCP/ IP Receiver Throughput (Mbits/sec)**	**TCP/IP Half-Power Point (Bytes)**
Solaris 2.4 on SS-20	Ethernet: AMD Lance Ethernet NI and Bay Networks EtherCell	1638 + 2.18n (R^2=0.979)	1009	9.0	108
Solaris 2.4 on SS-20	ATM: SunATM-155 NI and Bay Networks LattisCell Switch	1261 + 0.32n (R^2=0.999)	1285	84.3	2194
Solaris 2.4 on SS-20	ATM: Fore SBA-200 NI and ASX-200 Switch	1432 + 0.32n (R^2=0.998)	1368	81.2	1196
Solaris 2.4 on SS-20	Myricom Myrinet	1492 + 0.36n (R^2=0.997)	1443	75.3	2847
HP-UX 9.01 on HP 9000-735	Intel Ethernet	1330 + 2.18n (R^2=0.981)	657	9.0	13
HP-UX 9.01 on HP 9000-735	HP Medusa NI and FDDI	446 + 0.34n (R^2=0.992)	408	96.4	786

TABLE 4. Round-trip delay and peak delivered throughput for a variety of commercial local area network environments.

sured the round-trip time between two hosts connected by routes that passed through the switch two times. By subtracting the first measurement from the second, we derived the round-trip latency through the switch. (The only exception to this methodology is the method used to determine the Ethernet switch latency For first part of this experiment, we measured round-trips across an idle shared Ethernet. For the second part of the experiment, we measured round-trips across one hop through the switch.) Table 3 illustrates the round-trip latency of several commercially available switches. (Note that we did not measure the latency of the network interface cards.)

Switch	**Round-Trip Latency** (µs)
Bay Networks EtherCell Ethernet	104
Bay Networks LattisCell ATM	208
Fore ASX-200 ATM	26
Myricom Myrinet 8-Port Switch	1 [Martin95]

TABLE 3. Switch latencies.

3.3.3 Local Area Network Performance

We measured round-trip delays and delivered throughput for several commercial local area network systems. Figure 2a and Figure 2b show the round-trip times for messages less than 250 bytes, which comprise nearly all local area network traffic. (Figure 4a and Figure 4b in Section 9 show UDP/IP round-trip times for messages from 8 bytes to 10 K bytes.) Figure 2a shows the round-trip times for Myrinet, Fore Systems ATM and Sun/Bay Networks ATM and switched Ethernet on the Solaris 2.4 SS-20 platform, while Figure 2b shows performance for switched Ethernet and the Medusa/FDDI network on the HP-UX 9.01 HP 9000-735 platform. (It should be noted that the HP Medusa/FDDI exchanges utilize a single-copy protocol stack. This stack differs from the traditional protocol stack in that it takes advantage of the Medusa NI's IP checksum unit and onboard memory to make only one copy of the data [Banks93]. The switched Ethernet measurements utilize the traditional HP-UX protocol stack.)

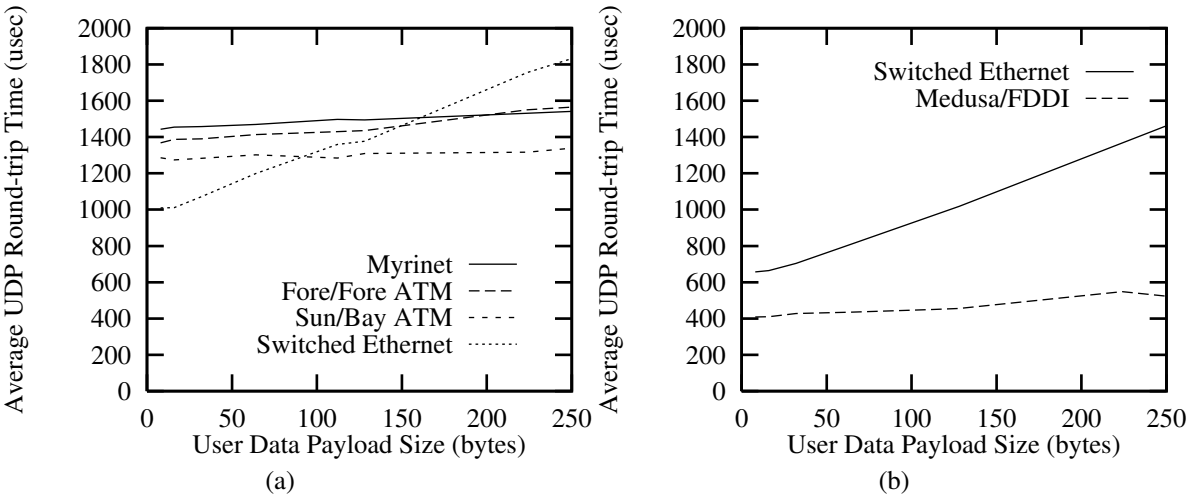


FIGURE 2. UDP/IP Round-trip times for several local area network environments. (a) Solaris 2.4 networks (small messages). (b) HP-UX 9.01 networks (small messages). Data for message sizes up to 10 KB can be found in Section9.

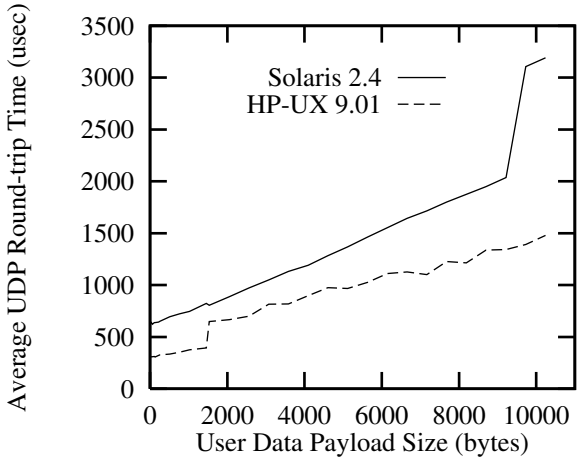
TCP/IP delivered throughput is shown in Figure 3a and Figure 3b. Figure 3a presents throughput for the Solaris-based networks with the TCP/IP window size increased from its default size of 8 KB to 64 KB. This increase was suggested as a workaround to a known performance problem in the Solaris 2.4 kernel that limited throughput to less

4. A user data size of eight bytes was chosen to ensure that the user data plus the UDP, IP, ATM AAL 5, and ATM protocol headers and trailers fit into a single 53-byte ATM cell.

3.3 Results

3.3.1 Loopback Protocol Software Overheads

We measured end-host overheads by performing UDP/IP round-trip timings for communication between a host and itself. This style of experiment, called *loopback*, gives the round-trip time through the Internet protocol software and a loopback device driver, and includes twice the send overhead plus twice the receive overhead. While it does not measure the exact time to inject a message into the network, it provides a reasonable estimate of the time spent in the UDP/IP protocol stack. Figure 1 shows the UDP/IP round-trip times, and Table 2 displays the per-message and per-byte costs.



(a)

FIGURE 1. Loopback UDP/IP round-trip times for Solaris and HP-UX.

Operating System/ Workstation Platform	SPECInt92 Rating[a]	UDP/IP Round-Trip Overheads (μs)	Measured Round-Trip Time for 8 B Message (μs)
Solaris 2.4 on SparcStation 20	73.7	604 + 0.15n (R^2 = 0.998)	620
HPUX-9.01 on HP 9000-735[b]	109.1	320 + 0.12n (R^2=0.961)	308

TABLE 2. Protocol processing overheads.

a. These SPECInt92 ratings are for uniprocessor machines[DiMarco95]. Our Solaris 2.4 experiments were performed on a dual processor machine.

b. This experiment uses the standard Internet protocol stack, while the results shown in Figure 2, Figure 3, Table 4, and Table 5 use a single-copy Internet protocol stack.

3.3.2 Switch Latency

Switch latency was determined by a series of two round-trip measurements using UDP/IP and 8-byte packets.[4] First, we measured the round-trip time between two hosts connected by a single path through the switch. Second, we mea-

3.2 Measurement Methodology

Our analysis was performed with a simple user-level program that performs communication using TCP/IP and UDP/IP through the BSD-style socket programming interface. We chose these protocols because they comprise nearly all of local area network traffic [Kay93]. Furthermore, because sockets and TCP/IP and UDP/IP are used to perform communication, the benchmark program is very portable among platforms and network environments. The program implements two types of experiments that reflect commonly used communication styles: round-trip exchanges, and one-way data transmission. As we will see in the following section, this combination of user-level tests is very useful for deriving information about various components of the network. [1]

In the round-trip experiments, a client machine sends a single request message to a machine simulating a server. The server then echoes the message back to the client to simulate a reply. The client then repeats this exchange for a fixed number of packets. These measurements, using small message sizes, can help to determine the latency or overhead through various components of the network.

In the one-way transfer experiments, a sending machine transmits a fixed number of messages as fast as it can to a receiving machine. When the transfer completes, the receiver sends a single message back to the sender to acknowledge the transfer. This type of experiment is used to measure the effective point-to-point throughput of the network and attached hosts.

While we performed experiments using all combinations of protocol and communication pattern, we focus our presentation in this section on UDP/IP round-trip experiments and TCP/IP one-way transfer experiments.[2] (Results from the alternate combinations of experiments are presented in the appendix in Section 9.) Each experiment was run for message sizes from 8 bytes to 10 Kbytes; in each experiment, 5000 messages were exchanged. For round-trip experiments, the round-trip time was computed by dividing the elapsed time by the number of messages exchanged. For one-way transfer experiments, the delivered throughput was measured as the amount of user data received.[3] Throughout all experiments, both the send and receive socket buffer sizes were increased to 64 KB. Each data point reported is the average of six independent repetitions of the corresponding experiment.

In addition to presenting the raw round-trip time data for the round-trip experiments, we also present a linear regression model that breaks down the round-trip into per-message costs and per-byte costs. (Note that a round-trip is composed of two sending overheads plus two receiving overheads plus two latencies.) We performed simple linear regression for message sizes up to 8 KB, to obtain a linear model, $M + Dn$. Here M is the per-message cost (i.e., the y-intercept of the graphed data), and D is the per-byte cost (i.e., the slope of the graph), and n is the message size (i.e., the user data payload) in bytes. We also give the coefficient of determination, R^2, to indicate the fit of the regression. The higher the value of R^2, the better the regression: if all observed values are equal to those predicted by the model, R^2 is 1. For one-way transfer experiments, in addition to the raw data, we present the peak received throughput as well as the *half-power point*, which is the message size for which half of the peak throughput is obtained.

---

1. The software for both the round-trip and one-way transfer experiments is available via the WWW at the URL `http://http.cs.berkeley.edu/~kkeeton/Software/software.html`.

2. UDP was chosen for the round-trip test because of the simple nature of the protocol -- each message is sent as its own unreliable individual datagram, without any transport-level acknowledgment. TCP was chosen for the throughput test because it provides flow control between the sender and receiver; that is, it prevents the sender from sending data faster than the receiver can accept it.

3. The actual bit throughput, which includes various protocol headers, is somewhat higher

2.7 Summary of Factors Affecting Communication Performance

Media sharing, bandwidth, overhead, latency, communication patterns, message sizes, and contention all affect LAN communication performance. Because each host is connected to the network via a dedicated link, switched network topologies yield higher aggregate bandwidth than shared-media topologies; they also typically exhibit less performance degradation under contention. Host software protocol and network interface overheads affect communication performance for both one-way transfers and request-response communication patterns. One-way transfers are limited by send overheads, which limits how much of the link bandwidth can be used. Overhead is especially important for small messages, which comprise an overwhelming majority of all LAN traffic. Both overhead and latency are important for request-response communication performance. Finally, the presence of contention or congestion in the network may affect communication performance by increasing queueing delays and causing packets or cells to be dropped.

3 Measurements of Commercial Local Area Networks

We performed several experiments to characterize the LogP parameters for commercially available network hardware and software. This section describes the experimental methodology and presents measurements for switch latency, end-host overhead, and delivered bandwidth for both one-way transfers and request-response-style communication.

3.1 Experimental Environment

We performed experiments on the different hardware and software environments described in Table 1. We used two types of host workstations: 50 MHz dual-processor Sun SparcStation 20s running Solaris 2.4 and 99 MHz Hewlett-Packard 9000-735s running HPUX-9.01. We measured several different networks, including Ethernet, FDDI, ATM, and Myrinet. Our three ATM switches use OC-3C links operating at 155 Mbits/sec. The Fore ATM switch has eight ports, while both the ATM switch and the Ethernet switch from Bay Networks have sixteen ports.

Myrinet is a new type of LAN that employs techniques from massively parallel processor interconnects [Boden95]. It provides flow-controlled communication channels and uses cut-through routing to route arbitrary-length packets. The Myrinet switch we measured has eight ports, and uses 640 Mbit/sec links.

The HP Medusa FDDI interface is an experimental network interface designed for a high-performance single-copy TCP/IP stack; it resides on the graphics bus of the HP workstation, has one megabyte of onboard memory, and supports IP checksums in hardware [Banks93].

Operating System/ Processor	**Network Interface (NI)**	**Switch**	**Link Bandwidth (Mbits/sec)**
Solaris 2.4 on SS-20	AMD Lance Ethernet	Bay Networks EtherCell Model 28115 Ethernet	10
Solaris 2.4 on SS-20	Sun ATM-155 Sbus ATM	Bay Networks LattisCell 10114 ATM	155
Solaris 2.4 on SS-20	Fore SBA-200 ATM	Fore ASX-200 ATM	155
Solaris 2.4 on SS-20	Myricom Myrinet	Myricom Myrinet	640
HPUX-9.01 on HP 9000-735	Intel Ethernet	Bay Networks EtherCell Model 28115 Ethernet	10
HPUX-9.01 on HP 9000-735	HP Medusa (FDDI)	N/A	100

TABLE 1. Host and network configurations.

2.3 Latency

As in the LogP model, we define latency as the time for a bit to cross the network from the source network interface to the destination network interface. Latency includes various NI functions, such as network media access, as well as switch and propagation delays. Switch delays can be attributed to signalling, routing translation and buffering delays. Propagation delays, both across the switching fabric and across the links, comprise the final component of latency

2.4 Communication Pattern

Overhead plays an important role in determining communication performance for operations where messages are sent in a one-way exchange between sending and receiving hosts, such as a store to remote memory In this scheme the sending overhead provides a lower bound on the time between consecutive message sends. The sender can send another message as soon as it has completed the software protocol processing for the current message, because it need not wait for an acknowledgment back from the receiver. For this communication style, the latency does not contribute to the cost of communication, because it can be overlapped with the software protocol processing for the next message.

For request-response operations, such as a remote procedure call (RPC), computation and communication cannot be overlapped. In this case, the requestor cannot resume computation until a response has been received. In this case, both latency and send and receive overheads limit the communication performance.

2.5 Message Sizes

Just as different factors dominate one-way and request-response communication patterns, different factors dominate communication cost for large and small messages. For instance, large messages amortize per-message overhead over a large number of bytes, making per-byte costs and bandwidth the more important factors in determining performance. For smaller messages, however, per-message overheads dominate, because much more of the communication time is spent in protocol processing than in actually transmitting the data onto the link. This distinction is especially important when we consider what message sizes are typically seen on LANs. Network traces [Kay93, Gusella90] show that 90% of the packets are UDP packets, mostly generated by NFS; the rest are TCP segments. The study shows that over 99% of all the TCP packets are less than 200 bytes long. UDP packets, on the other hand, fit a bimodal distribution: 86% are less than 200 bytes long, while 9% are at least 8192 bytes long. Because most messages are small, we can predict that overhead will limit network performance.

2.6 Contention and Congestion

Thus far, we have concentrated on the performance of point-to-point communication. Network performance, as seen by the end-user, can also be greatly affected by other traffic present in the network. If the aggregate cross-traffic is great enough to partially consume the network resources, we say that there is *contention* in the network. We say the network is *congested* if network resources have been completely consumed. Contention may result in degraded performance due to higher queueing delays in the switch and host buffers, longer time to gain transmit access to the bus in bus-based switches, and increased time to access shared switch buffers.

Networks react to congestion in different ways, depending on the switch hardware and host protocol software. For instance, if buffers in switches or on the end-hosts overflow with packets, the switch or host may choose to drop the packets. (Reliable delivery of those packets must then be accomplished through retransmission at the protocol software level.) Some networks, such as ATM, employ hardware-based flow control to restrict each sending-receiving pair to an agreed-upon rate of transmission and network usage. By restricting network usage to what the network capacity allows, and by controlling the flow of information between all communicating hosts, these networks are able to minimize congestion.

In this paper we focus on point-to-point network performance on an unloaded network; we leave analysis of network performance under contention for future work.

2 Background

The LogP model, described in [Culler93], presents a model for describing and evaluating communication via point-to-point messages between processors in a distributed-memory multiprocessor. *Latency*, *L*, is defined as the time for a bit to cross the network from source to destination. It includes the time for the network interface to process the message to be sent or received, the time to acquire transmission access to the network and any switch and propagation delays. *Overhead*, *o*, is the time the processor spends sending or receiving a message. During this time, the processor cannot perform other operations. This includes software protocol processing, buffer management, context switching (e.g., thread or process switching and interrupt handling) and handshaking with the network interface. *Gap*, *g*, is the minimum time interval between consecutive message transmissions or consecutive message receptions at a processor. The reciprocal of g is the available per-processor communication bandwidth. In local area networks, the overheads to send and receive a message are typically so high that they mask any 'gap' limitation imposed by the bandwidth of the network. As a result, the gap is irrelevant to our study. The final component of the model is *P*, the number of *processors*. Again, this parameter is not relevant for our study of point-to-point communication in the local area network environments.

In this section we argue that overhead, and latency, as well as network topologies, communication patterns, message sizes, and contention are the important factors for consideration in the evaluation of local area network performance.

2.1 Bandwidth and Shared vs. Switched Media

The focus of local area network technology is shifting from shared-media networks to switch-based networks. In the former, all hosts share the same network medium, usually in a bus or ring topology. As a result, the aggregate network bandwidth is approximately the same as the bandwidth of the individual links. Performance will degrade (potentially severely) as load on the network increases. Two examples of shared-medium networks are Ethernet, which operates at 10 or 100 Mbits/sec, and FDDI (Fiber Distributed Data Interface), which operates at 100 Mbits/sec.

Switch-based networks use point-to-point links to connect switches in an arbitrary topology that allows aggregate network bandwidth much larger than the bandwidth of a single link. Because hosts are connected to the network via a dedicated link, performance does not degrade as sharply when load on the network increases. Throughput can be increased incrementally by adding extra switches and links to match workload requirements. Examples of switched networks are switched Ethernet, which employs 10 or 100 Mbits/sec link speeds; ATM (Asynchronous Transfer Mode) [DePrycker93], which uses (among other link speeds) 155, 622, or 1244 Mbits/sec links; and Myrinet [Boden95], which uses 640 Mbits/sec links. Because of the advantages of switch-based networks, we focus our attention on them for the remainder of this study.

2.2 Overhead

While much higher link bandwidth may be available in new switched networks, communicating hosts may be unable to take advantage of it due to the software processing needed to send or receive a message. There are two components to local area network overhead: software protocol overheads and network interface hardware overheads.

Overhead from protocol software, such as the ubiquitous TCP/IP and UDP/IP protocol suites, can be further broken down into per-message and per-byte overheads [Wolman94, Kay93, Clark89]. Examples of per-message overheads are protocol control block manipulation, software interrupt handling, context switch processing, and protocol-specific processing (e.g., address resolution or TCP flow control). Examples of per-byte overheads are data checksumming and data movement (e.g., user-to-kernel or kernel-to-network interface). Other overheads, such as BSD-style mbuf manipulation and fragmentation of messages, are dependent on the length of the message, even though they are not incurred on a per-byte basis.

Network interface (NI) hardware overheads comprise the remainder of send and receive overheads. These overheads include the cost of handshaking between the NI and the host CPU, such as the cost of transferring data using direct memory access (DMA) or programmed I/O (PIO). Other network interface features, such as buffering or protocol support, can decrease the protocol software overheads. The primary impact of including buffer memory on the network interface is that the number of copy operations can be reduced; if interface buffers are mapped into the user address space, copies can be eliminated altogether. Finally, processing power can be provided on the NI to off-load portions or all of the protocol processing.

This paper was presented at Hot Interconnects III: A Symposium on High Performance Interconnects, Stanford University, Stanford, CA, August 10 - 12, 1995.

LogP Quantified: The Case for Low-Overhead Local Area Networks

Kimberly K. Keeton, Thomas E. Anderson, and David A. Patterson
{kkeeton,tea,pattrsn}@CS.Berkeley.EDU

Computer Science Division
University of California at Berkeley
387 Soda Hall
Berkeley, CA 94720-1776
(510) 642-9669
(510) 642-5775 (fax)

Abstract

As we make an ongoing transition to switch-based networks, we find that the marketing focus is link bandwidth. This paper argues that many other factors, such as overhead, latency, communication pattern, message sizes, and contention contribute to overall communication performance. Our findings indicate that overhead is the dominant factor in determining communication performance, especially for request-response communication using small messages. Furthermore, an important class of applications, namely NFS file systems and transaction processing database systems, will not be able to take advantage of increases in network bandwidth unless host protocol software overheads are also lowered.

1 Introduction

In recent years there has been a technological evolution from low-bandwidth shared media local area networks, such as Ethernet and FDDI, to switched local area networks, such as ATM and Myrinet. These new networks have both higher link bandwidth and much higher aggregate bandwidth. Their marketing literature cites link bandwidth specifications but often omits other factors that can affect performance, such as overhead, latency, and contention. The overwhelming focus on bandwidth might lead people to believe that it is *the* single figure of merit for evaluating network performance.

Unfortunately, bandwidth has become the MIPS of the commercial networking community. As with MIPS, there are several pitfalls to using bandwidth alone to explain network performance [Patterson90]. Most importantly, higher bandwidth does not necessarily imply improved communication performance. Network performance differs among programs, depending on communication patterns and message sizes. While higher bandwidth may improve performance for some applications (especially those that use large messages to transfer data), there are other classes of applications that will not be able to take advantage of increased bandwidth unless network latencies and end-host overheads are also decreased. Although there is considerable active research to decrease overheads and latencies [Basu95, Rodrigues95, Martin94], many of these successes haven't yet appeared in the commercial marketplace.

We argue that network performance, as seen by end-user applications, must be characterized in terms of bandwidth, overhead, latency, communication patterns, message sizes and contention. We find that network overheads dominate communication costs, especially for request-response communication patterns and communication using small messages. Our results indicate overheads in new higher-bandwidth networks are actually increasing. Both of these results have important implications for end-user application performance. We show for two important classes of applications, namely NFS file service and online transaction processing, that increasing network bandwidth does little to increase user application performance. In fact, the high overheads imposed by these higher-bandwidth networks actually cause application performance to be worse than Ethernet performance for certain workloads.

The paper is organized as follows. Section 2 describes the importance of various components of communication costs. Section 3 quantifies several of these components for commercially available local area networks. In Section4 we show the effects of network performance on end-user application performance. Section 5 summarizes our results and conclusions, and Section 6 presents areas for future research.