Isolating the Performance Impacts of Network Interface Cards through Microbenchmarks

Vijay S. Pai Scott Rixner Hyong-youb Kim

Rice University
Houston, TX 77005
{vijaypai,rixner,hykim}@rice.edu

Categories and Subject Descriptors: C.4 [Performance of Systems]: Measurement Techniques

General Terms: Measurement, Performance

Keywords: Network server performance, Networking microbenchmarks

1. INTRODUCTION

Many factors can prevent a Gigabit Ethernet network interface card (NIC) from achieving line rate in a modern web server. In fact, the various commercially available NICs have different performance characteristics that lead to throughput differences for actual web servers. For example, Figure 1 shows the performance achieved by the thttpd web server for client traces extracted from the Rice University computer science department (CS), a NASA web site (NASA), and the 1998 soccer World Cup tournament (World Cup). The latter two traces are available from the Internet Traffic Archive (`http://ita.ee.lbl.gov/`). The server system tested includes an AMD Athlon 2600+ XP processor running the FreeBSD 4.7 operating system, 2 GB of DDR SDRAM, a 64-bit/66 MHz PCI bus, and a single 40 GB IDE disk (none of the workloads are disk intensive). The tested systems differ only in their NIC, with the Intel Pro-1000/MT Server and Desktop, Alteon AceNIC with parallelized firmware [2], Netgear GA-622T, 3Com 3C996B, and Alteon AceNIC with released firmware arranged from left to right. There are substantial performance differences across the NICs in the web environment, as the fastest NIC consistently achieves 40–60% more throughput than the slowest.

A web server interacts with the network in two primary ways: receiving client HTTP requests and sending HTTP responses. Requests are typically quite small, on the order of 200 bytes of ASCII text, while responses vary from empty files to several hundred megabytes. Since web clients and servers communicate using TCP, the server must acknowledge requests, leading to minimum-sized (64-byte) Ethernet frames. Response data must be segmented and encapsulated in Ethernet frames, which allow up to 1460 bytes of TCP content in a maximum-sized (1518-byte) frame. Then, those segments are sent out according to TCP flow control policies based on the receipt of acknowledgments. A high-performance server NIC must thus support data volumes dominated by sends of large frames while also efficiently receiving and sending small frames. Each NIC's ability to support this type of network traffic accounts for the performance differences in Figure 1.

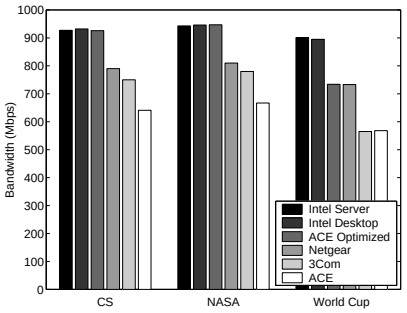

Figure 1: Throughput in Mbps achieved by the thttpd web server running on a PC-based server with various NICs.

2. NETWORKING MICROBENCHMARKS

Among the most popular microbenchmarks to measure the bandwidth of various network protocols and implementations are netperf and LMbench [1, 3]. However, there are three main problems with these tests. First, they use standard networking APIs (e.g., `read` and `write`) for portability rather than more advanced system calls such as `sendfile`, which uses zero-copy I/O to improve performance [4]. Second, the UDP tests do not throttle datagram production under overload conditions, so the operating system can do extra work to create datagrams only to have them dropped by the device driver or the network interface. Finally, the TCP tests use only a single connection, causing latency and window-size to limit achieved bandwidth. In contrast, server applications use techniques such as zero-copy I/O and fast event notification for connection management. These techniques allow servers to achieve higher network throughputs than these microbenchmarks, despite the extra computational work of servers.

This paper proposes and utilizes a new microbenchmark suite specifically aimed at isolating and characterizing the micro-level behaviors of network interfaces that impact system-level performance. The suite includes UDP and TCP unidirectional send and receive tests and UDP bidirectional tests; all tests are performed separately for maximum-sized and minimum-sized frames. Details of these tests and two others can be found in [5].

This suite isolates the performance of the NIC by overcoming the problems listed above. First, all transmissions of UDP and TCP data are performed by a new system call into FreeBSD that bypasses `read` and `write`, instead continually replaying pre-built packets of the appropriate size to the device driver. This system

This work is supported in part by a donation from AMD, by the DOE under Contract Nos. 03891-001-99-4G, 74837-001-0349 and/or 86192-001-0449 from LANL, and by the NSF under Grant Nos. CCR-0209174 and CCR-0238187.

SIGMETRICS/Performance'04, June 12–16,2004, New York, NY, USA
ACM 1-58113-664-1/04/0006.

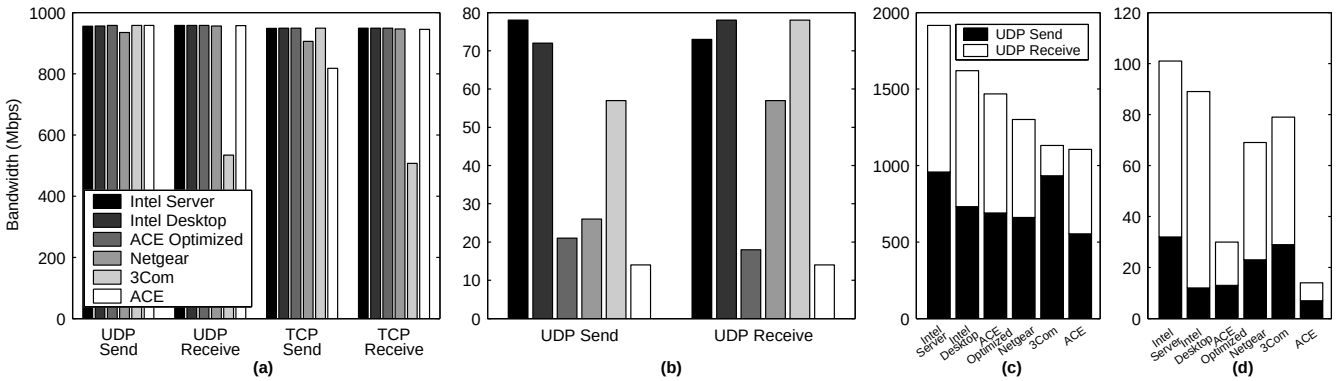

Figure 2: Send and receive throughput for maximum-sized UDP/TCP frames (a), minimum-sized UDP frames (b), and bidirectional throughput for maximum-sized (c) and minimum-sized (d) UDP frames.

call enables the microbenchmarks to isolate the performance of the NIC from most operating system effects; however, device driver performance can still impact the results. Second, all of the UDP microbenchmarks can be throttled to a specified rate, eliminating overheads from unsuccessful transmissions. Finally, the TCP send and receive tests support a configurable number of independent connections, eliminating latency and window size limitations.

3. MICROBENCHMARK RESULTS

Figure 2a shows the performance of various network interfaces in the tests which send and receive maximum-sized Ethernet frames. Each test involves two machines, one with the NIC under test and the other with the best NIC for receiving or sending large frames (Intel Server and 3Com, respectively). All of the NICs achieve near maximum throughput when sending large UDP datagrams and all but the unoptimized ACE achieve near maximum throughput when sending large TCP segments. The TCP send performance of the unoptimized ACE suffers because its firmware shares a single programmable processor between send and receive (e.g., ACK) processing; this is in contrast to the parallelized firmware of the optimized ACE. On the receive path, for both UDP and TCP streams, all NICs achieve near peak throughput except the 3Com NIC. The 3Com NIC's throughput drops substantially because of a workaround in the driver that limits the maximum length of a DMA transfer to avoid tripping a bug in the checksum offloading features of this NIC.

Figure 2b shows the UDP data throughput for minimum-sized frames (18 byte datagrams), using one machine with the NIC under test and another with the most efficient receiver (Intel Desktop) or sender (Intel Server). The achieved throughputs are not only dramatically lower than in Figure 2a, but they are also significantly lower than the maximum possible for Ethernet on minimum-sized frames (214 Mbps because of the overhead of protocol headers, checksums, and interframe gaps). These results indicate that these NICs are not designed for high performance on small frames and that per-frame overheads substantially limit performance.

Figure 2c shows the performance of each NIC when maximum-sized UDP datagrams are being sent to it by the best sender (3Com) and it is simultaneously sending maximum-sized UDP datagrams to the best receiver (Intel Server). These results show a clear trend among the NICs, as performance degrades from the left to the right in the figure, which matches the application-level performance trends. Only the Intel Server NIC achieves nearly the sum of its individual send and receive bandwidth. The other network interfaces most likely have some limited resources shared between the send and receive paths; possible limitations include PCI bandwidth (likely for Intel Desktop, which only has a 32-bit PCI interface with a theoretical maximum of 2 Gbps), a shared programmable processor (unoptimized ACE), or on-board memory bandwidth (since the NIC memory touches each bit of network traffic twice: once on the network side and once on the host side).

Figure 2d shows the performance of each NIC when minimum-sized UDP datagrams are being sent to it by the best sender (Intel Server) and it is sending minimum-sized UDP datagrams to the best receiver (Intel Desktop). In this test, no NIC approaches the sum of its individual send and receive bandwidth for minimum-sized datagrams. ACE saturates its single shared processor for either send or receive traffic alone, so it obtains the same total throughput on bidirectional traffic as it does on unidirectional traffic.

4. CONCLUSIONS

This paper promotes microbenchmarking of network interfaces as a tool to isolate the low-level behaviors that impact application-level performance. The microbenchmark results show that all NICs tested can achieve near wire-speed in sending large frames, but that the performance of these NICs varies greatly when processing bidirectional streams of large frames (up to 73% throughput difference between NICs), bidirectional streams of small frames (up to a factor of seven throughput difference), or unidirectional streams of small frames (up to a factor of five throughput difference). The differences in web server performance correlate most closely to bidirectional streams of large frames, despite the send-dominated volume of traffic in a web server. The extended version of this paper also includes a more detailed analysis using statistical methods [5].

5. REFERENCES

[1] Information Networks Division, Hewlett-Packard Company. *Netperf: A Network Performance Benchmark*, February 1995. Revision 2.0.

[2] H. Kim, V. S. Pai, and S. Rixner. Exploiting Task-level Concurrency in a Programmable Network Interface. In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 61–72, June 2003.

[3] L. McVoy and C. Staelin. lmbench: Portable Tools for Performance Analysis. In *Proceedings of the 1996 USENIX Technical Conference*, pages 279–295, January 1996.

[4] E. M. Nahum, T. Barzilai, and D. Kandlur. Performance Issues in WWW Servers. *IEEE/ACM Transactions on Networking*, 10(2):2–11, February 2002.

[5] V. S. Pai, S. Rixner, and H. Kim. Isolating the Performance Impacts of Network Interface Cards through Microbenchmarks. Technical Report EE0401, ECE Department, Rice University, March 2004.