Event-signaling within Higher Performance Network Subsystems

Jeffrey D. Chung, C. Brendan S. Traw and Jonathan M. Smith*
University of Pennsylvania
{jdchung,traw,jms}@cis.upenn.edu

Abstract

The Afterburner ATM Link Adapter has allowed us to evaluate three event-signaling schemes: polling, traditional interrupts and the clocked interrupts first investigated in our operating system work in AURORA. The schemes are evaluated in the context of a single-copy TCP/IP stack. The experimental results indicate that clocked interrupts can provide throughput comparable with traditional interrupts for dedicated machines (up to over 144 Mbps, the highest TCP/IP/ATM throughput reported), and better performance when the machines are loaded with an artificial workload. Polling, implemented to be used with an unmodified `netperf` measurement tool, was competitive for small TCP/IP socket buffer sizes (¡32KB). We conclude that clocked interrupts may be preferable for applications requiring high throughput on systems with heavy processing workloads, such as servers.

1 Introduction

1.1 Background

This research is one of a series of results from an exploration of Asynchronous Transfer Mode (ATM) computer/network host interface architectures at the University of Pennsylvania, begun in 1990 [Traw 93], as part of the ATM/SONET infrastructure of the AURORA Gigabit Testbed [Clark 93]. The goal of AURORA was to develop the technologies needed to support end-to-end gigabit per second networking between workstations.

The initial research goal of the interface work was to identify and experimentally verify a kernel of services that were suitable for hardware implementation. These data movement and formatting intensive services include ATM Adaptation Layer (AAL) processing, segmentation-and-reassembly (SAR) and ATM demultiplexing. The architecture partitioned the protocol processing activities between the hardware host interface and software running on the host processor. Concurrent with the hardware implementation, software support appropriate for high-throughput networking had to be designed and implemented. This was an opportunity to explore non-traditional approaches to reducing copying, event-signaling, and application-programmer interfaces for IPC. Our schemes for reducing copying and a shared-memory IPC model have proven themselves effective[Smith 93], and while our clocked interrupt scheme was shown to be effective, it was not conclusively characterized with respect to other event-signaling schemes.

*Work at Penn was supported by the National Science Foundation and the Advanced Research Projects Agency under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by the NSF under agreement CDA-92-14924, by Bell Communications Research under Project DAWN, by an IBM Faculty Development Award, and by the Hewlett-Packard Corporation.
Brendan Traw is now at the Intel Architecture Laboratory, Hillsboro, OR and can be contacted at `Brendan_Traw@ccm.jf.intel.com`.

1.2 Event-signaling and Clocked Interrupts

Event-signaling within the network subsystem between the hardware network interface device and the software device driver is typically accomplished via polling or device-generated interrupts. In our implementation of an OC-3c ATM host interface for the IBM RS/6000 family of workstations[Traw 93, Smith 93], we replaced the traditional forms of this crucial function with "clocked interrupts." Clocked interrupts, like polling, examine the state of the network interface to observe events which require host operations to be performed. Unlike polling, which requires a thread of execution to continually examine the network interface's state, clocked interrupts perform this examination periodically upon the expiration of a fine-granularity timer. In comparison to interrupts, clock interrupts are generated indirectly by the timer and not directly by the state change event.

1.3 Clocked Interrupt Discussion

We developed an analytical model for clocked interrupt performance in our earlier work[Smith 93], and argued that for application workloads characterized by high throughput, heavy multiplexing, and/or "real-time" traffic, clocked interrupts should be more effective than either traditional polling or interrupts. For these intensive workloads, our analysis predicted that clocked interrupts should generate fewer context switches than traditional interrupts and require fewer CPU cycles than polling without significantly increasing the latency observed by the applications. We note that for traditional interrupts with interrupt service routines which detect additional packets enqueued on the adapter, many of the same benefits may accrue. Ramakrishnan[Ramakrishnan 93] has noted a problematic performance overload phenomenon known as receive livelock which clocked interrupts can help alleviate.

Our clocked interrupt implementation for the ATM subsystem of the IBM RS/6000 proved to be effective, but could not be directly compared with interrupts as the network interface was (by design) incapable of generating them. A second implementation of the hardware portion of the host interface architecture has been built as an OC-12c rate ATM Link Adapter for the HP Bristol Labs "Afterburner"[Banks 93] card.

2 The HP Afterburner and UPenn ATM Link Adapter

The hardware infrastructure for this evaluation consists of HP 9000/700 series workstations equipped with Afterburner generic interface cards and ATM Link Adapters. Figure 1 shows an Afterburner and ATM Link Adapter. The remainder of this section briefly describes the architecture and implementation of the Afterburner and ATM Link Adapter.

2.1 Afterburner

The Afterburner [Dalton 93], developed by HP Laboratories in Bristol, England, is based on Van Jacobson's *WITLESS* architecture. It provides a high speed generic packet interface which attaches to the SGC bus of the HP9000/700 workstations. A large pool of triple ported Video RAM (VRAM) is provided by Afterburner. The random access port of the VRAM is visible on the SGC bus allowing the VRAM to be mapped into the virtual address space of the workstation. The two serial ports are used to provide a bidirectional FIFOed interface to a network specific *Link Adapter*. Several additional FIFOs are provided to assist in the management of VRAM buffer tags.

2.2 ATM Link Adapter

A Link Adapter provides an interface between the general purpose Afterburner and a specific network technology. The UPenn SAR architecture [Traw 93] is the basis for the ATM Link Adapter. This architecture performs all per-cell SAR and ATM layer function in a heavily pipelined manner which can be implemented in a range of hardware technologies. For the ATM Link Adapter the base SAR architecture has been extended to support a larger SAR buffer (up to 2 MB), AAL 5 including

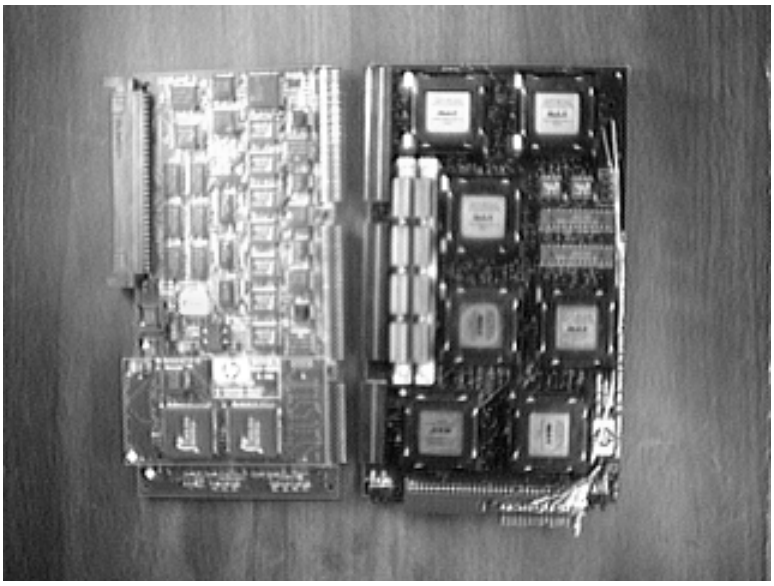

Figure 1: Afterburner (left) and ATM Link Adapter (right)

CRC32 generation and checking, and demultiplexing based on the full VPI, VCI, and MID. The performance of the implementation has been improved to 640 Mbps by using more advanced EPLD technology. Figure 2 shows the host/Afterburner/ATM Link Adapter configuration.

3 Implementation of the Clocked Interrupt Scheme on the Afterburner ATM Link Adapter

We implemented the ATM Link Adapter device driver to operate in conjunction with HP Bristol "Single-Copy" TCP/IP[Edwards 94]. The kernel was modified to support a fine-granularity timer, as the standard 100 Hz soft clock rate was inadequate. We had initially tried to simply increase the frequency, but the number of dependencies elsewhere in the kernel caused a large number of failures. Our strategy then became one of transparently supporting an increased clock rate for the specialized network subsystem code, while allowing the remainder of the system to retain the frequency. This

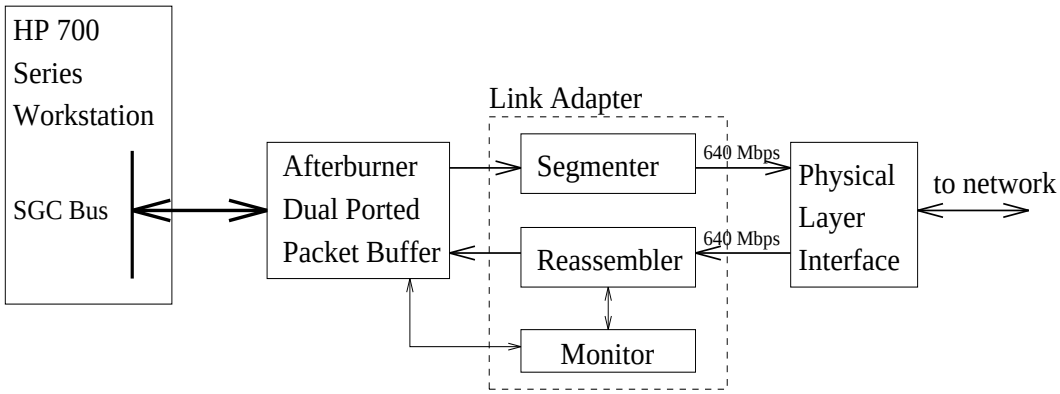

Figure 2: ATM Link Adapter

was done by modifying the operating system to increase the hardware clock interrupt rate, and changing the interrupt service vector to point to our specialized clock service routine rather than the usual `hardclock` interrupt service routine. Clock division is performed inside our code to call the `hardclock` interrupt service code at the proper rate. This approach, while aesthetically a bit crude, made the software engineering possible in a relatively short period. Thus, at each vector clock tick, occurring at the clocked interrupt clock rate, the link adapter is examined for packet arrivals. If packets are discovered the Interrupt Service Routine (ISR) for the ATM link adapter is invoked; this ISR provides the packet to the single-copy TCP/IP stack.

Polling requires a continuous thread of execution to examine the state of the I/O device. Because our version of HP-UX lacks preemptive kernel threads, polling was implemented with a preemptable user process. To minimize the number of system calls, the device status flag was appropriately memory mapped for access by a user process. This allowed a user process to continually examine the state of the device in a preemptable thread of execution, albeit at some cost in overhead. The user process invokes the ISR through an `ioctl()` call; for measurement purposes we wrote a small helper daemon which performed this function rather than modify `netperf`, again at a cost in overhead. Preemptive kernel threads would remove both these additional source of overhead.

Thus, the current implementation includes support for interrupt generation as well as the examination of the card via polling or clocked interrupts. With support for all three types of state change notification, we could perform a comparative experimental evaluation of these mechanisms.

4 Performance

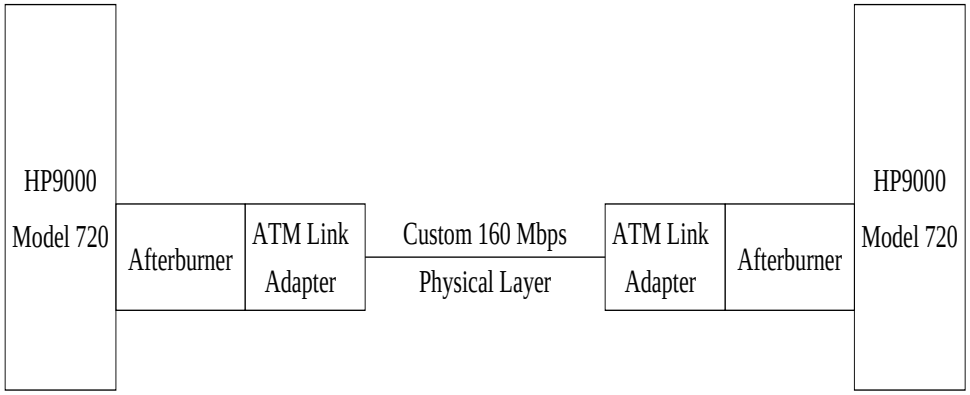

Figure 3: Experimental Setup

The hardware test configuration consists of the elements shown in Figure 3. Two HP 9000 Series 700 Model 720 workstations are connected back-to-back via their Afterburner ATM Link Adapter subsystems.

4.1 Measurements and analysis

We analyzed the throughput of the resulting network stacks using the `netperf` tool[IND 95]. We have experimented with both `ttcp` and `netperf`, and have drawn two conclusions from these experiments. First, `netperf` results are reproducible; `ttcp` measurements exhibit significant variation in reported throughput – up to 20% in some cases. Second, `netperf` results correspond very closely with maximum `ttcp` reported throughputs. What this suggests is that `netperf` better controls the variables under study, while reducing noise from other factors.

The results are given in Table 1. While behavior is more evident from graphs, space limitations precluded graphical presentation; the major observation is that polling does not keep up with the two

Socket Buffer Size (Kbytes)	Trad. intr.	Poll	Clock 500 Hz	Clock 1 KHz	Clock 2 KHz	Clock 4 KHz
1	6.75	6.34	2.60	3.92	5.88	6.67
2	12.45	13.24	5.02	7.36	9.81	11.94
4	20.82	22.43	9.28	13.40	18.17	21.57
8	30.80	37.27	16.20	22.96	26.58	35.35
16	51.73	50.03	21.72	42.03	45.64	50.35
32	66.83	64.02	37.95	52.26	61.72	64.27
64	76.25	76.78	57.17	65.27	70.91	73.22
128	124.97	81.57	95.00	110.03	117.24	121.09
256	144.05	82.62	143.76	144.10	143.59	143.81

Table 1: TCP/IP Throughput as measured by `netperf` with 32KB messages

other schemes above about 32KB. All checksums were enabled for all tests; the measurements were performed on dedicated processors, with no other activity except for necessary system background processes. The tests were run with symmetric configurations; that is, both sender and receiver were using the same signaling mechanism. We will examine asymmetric configurations in future work. It is clear from the figures shown that at high polling rates, the clocked interrupt scheme is able to keep up with the traditional interrupt scheme, which is almost everywhere the best performer, with the exception of polling, which does best for small packet sizes. In a lightly-loaded environment, interrupts would appear to be the best solution, except for some anomalous, but repeatable results which show polling best for small socket buffer sizes. It is important to note that the maximum possible theoretical throughput from the link on which these tests are run is 144.9 Mbps, indicating that other limitations are bounding the measured TCP/IP throughput.

Such a dedicated configuration is not characteristic of real environments, which are often loaded with other work and other network traffic. We created an artificial workload by continuously executing a "`factor 99121010311157`" command, which uses about 5.2 seconds of CPU time. This has a significant effect on the behavior of the three schemes, as can be seen by measuring the throughput with `netperf` with the artificial workload running on the receiver. In this case, for the socket buffer size of 262144 bytes, the traditional interrupts delivered 61.8 Mbps, polling delivered 48.6 Mbps, and clocked interrupts at 500 Hz gave 106.19 Mbps.

Another important factor in networking performance, and perhaps a more important parameter for distributed applications is the round-trip latency induced by the software supporting the adapter. Since the hardware was a constant, we could directly compare the software overheads of the three schemes. This was done with the following test. An artificial network load was created using `netperf` with a socket buffer size of 262144 bytes and operating it continuously. Against this background load, ICMP ECHO packets of 4K bytes were sent to the TCP/IP receiver, which was where the event-signaling performance differences would be evident. Sixty tests were done to remove anomalies. Our results showed that traditional interrupts and clocked interrupts at 500 Hz performed similarly, yielding minimum, average and worst case times of 5/12/18 ms, and 4/11/25 ms, respectively. When the systems were not loaded, the performances were 3/3/3 ms and 4/4/6 ms. This suggests that clocked interrupts performed slightly better under heavy load, but slightly worse under unloaded conditions.

5 Conclusions

Most importantly, the experiments reinforced the observation that packet size is the most important factor, by far, in maximizing observed throughput. However, such "jumbo-grams" are uncommon today (although they become more common with application such as video-on-demand). Our study has shown the following to be true. First, clocked interrupts can provide throughput equivalent to the best throughput available from traditional interrupts; both methods provide better performance

than polling as implemented here. Second, clocked interrupts provide higher throughput when the processor is loaded by a computationally-intensive process; this suggests that clocked interrupts may be a viable mechanism for heavily loaded systems such as servers, which might also suffer from Ramakrishnan's *receive livelock*. Third, clocked interrupts provide better round-trip delay performance when systems are heavily loaded for large ICMP ECHO packets.

Taken as a whole, the data suggest that clocked interrupts may be an appropriate mechanism for many of the high-performance applications now being proposed, such as video-on-demand servers. We would also like to note that we have obtained the highest reported TCP/IP performance for an ATM network adapter at this point in time.

6 Acknowledgments

John Lumley's group at Hewlett-Packard's European Research Laboratories (Bristol, UK) collaborated on the Afterburner ATM Link Adapter (particularly David Banks and Costas Calamvokis) and provided the single-copy TCP stack (Aled Edwards and Chris Dalton) as a starting point for the work reported here.

References

[Banks 93] D. Banks and M. Prudence, "A High-Performance Network Architecture for a PA-RISC Workstation," *IEEE JSAC*, 11(2), pp. 191-202 (Feb. 1993).

[Clark 93] David D. Clark, Bruce S. Davie, David J. Farber, Inder S. Gopal, Bharath K. Kadaba, W. David Sincoskie, Jonathan M. Smith, and David L. Tennenhouse, "The AURORA Gigabit Testbed," *Computer Networks and ISDN Systems* 25(6), pp. 599-621, North-Holland (January 1993).

[Dalton 93] C. Dalton et al., "Afterburner: A network-independent card provides architectural support for high-performance protocols," *IEEE Network*, pp. 36-43 (July 1993).

[Edwards 94] A. Edwards, G. Watson, J. Lumley, D. Banks, C. Calamvokis and C. Dalton, "User-space protocols deliver high performance to applications on a low-cost Gb/s LAN," in *Proceedings, 1994 SIGCOMM Conference*, London, UK, 1994.

[IND 95] Hewlett-Packard Information Networks Division, "Netperf: A Network Performance Benchmark (Revision 2.0)", February 15, 1995.

[Ramakrishnan 93] K. K. Ramakrishnan, "Performance Considerations in Designing Network Interfaces," *IEEE JSAC* 11(2), pp. 203-219 (Feb. 1993).

[Smith 93] Jonathan M. Smith and C. Brendan S. Traw, "Giving Applications Access to Gb/s Networking," *IEEE Network* 7(4), pp. 44-52, (July 1993).

[Traw 93] C. Brendan S. Traw and Jonathan M. Smith, "Hardware/Software Organization of a High-Performance ATM Host Interface," *IEEE JSAC* 11(2), pp. 240-253 (Feb. 1993).

[Traw 95] C. Brendan S. Traw, "Applying Architectural Parallelism in High Performance Network Subsystems," Ph.D. Thesis, CIS Department, University of Pennsylvania, January, 1995.