# Gigabit Networks

*Jonathan M. Smith*

Distributed Systems Laboratory, University of Pennsylvania
200 South 33rd St., Phila., PA 19104-6389
`jms@cis.upenn.edu`

## 1. INTRODUCTION

This chapter summarizes what we have learned in the past decade of research into extremely high throughput networks. Such networks are colloquially referred to as ''Gigabit Networks'' in reference to the billion bit per second throughput regime they now operate in. The engineering challenges are in the integration of fast transmission systems and high-performance engineering workstations.
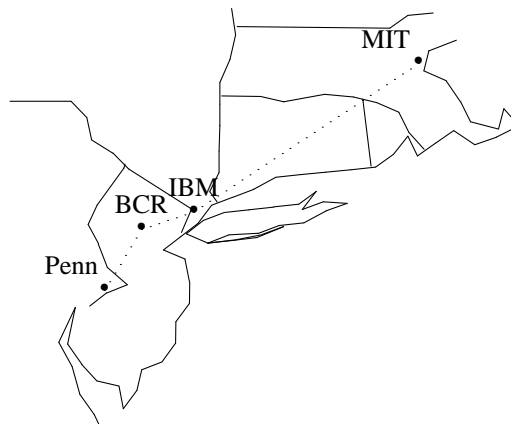


**Figure 1:** AURORA Geography

High-throughput fiber-optic networks, prototype high-speed packet switches, and high-performance workstations were all available in the late 1980s. A major engineering challenge was integrating these elements into a computer networking system capable of high application-to-application throughput. As a result of a proposal from D. Farber and R. Kahn (the ''Kahn/Farber Initiative'') the U.S. Government (NSF and ARPA) funded sets of collaborators in five ''Gigabit Testbeds'' [22]. These testbeds were responsible for investigating different issues, such as applications, MANs versus LANs, and technologies such as HIPPI, ATM and SONET [17]. The AURORA Gigabit Testbed linked Penn, Bellcore, IBM and MIT, with gigabit transmission facilities provided by collaborators Bell Atlantic, MCI and Nynex, and was charged with exploring technologies for gigabit networking, while the other four testbeds were applications-focused

and hence used "off-the-shelf" technologies. Results of AURORA work underpin today's high-speed network infrastructures.

Clark, *et al.* [5], set out the research goals and plans for the testbed, and outline major research directions addressed by the testbed. AURORA uniquely addressed the issues of providing switched infrastructure and high end-to-end throughput between workstation class machines. In contrast to supercomputers, these machines were, and are, the basis of for most computing today.

**Figure 1** and **Figure 2** provide illustrations of the AURORA geography and logical topologies, respectively.
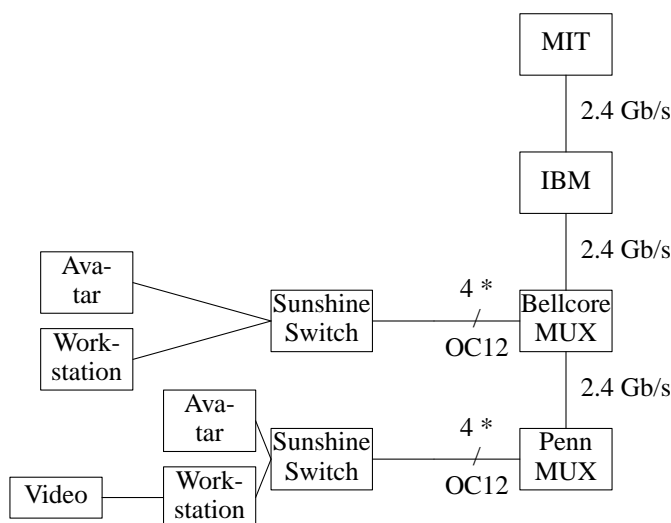


**Figure 2:** Partial AURORA Logical Topology

Sections 2, 3 and 4 describe ATM network host interface architectures that can operate in gigabit ranges, multimedia aspects of gigabit networks, and distributed shared memory as an applications programming interface for gigabit networks. Section 5 summarizes our state of knowledge.

*Table I* provides a compact summary of some major AURORA milestones.

## 2. ATM HOST INTERFACING

AURORA work showed that efficient, low-cost host/computer interfaces to ATM networks can be built and incorporated into a hardware/software architecture for workstation-class machines. This was believed problematic due to the nature of small, fixed-size ATM cells and their mismatch with workstation memory architectures. Penn designed [24] and implemented an ATM host interface for the IBM RISC System/6000 workstation which connects to the machine's Micro Channel bus. It translates variable-size application data units into streams of fixed-size ATM cells using dedicated segmentation-and-reassembly logic. The novel application of a Content-Addressable Memory, hardware-implemented Linked-List Manager and the reassembly pipeline structure allowed use of a low clock speed, and hence low-cost technologies. The cellification and decellification logic have measured performance which could support data rates of 600-700 Mbits/sec [26].

| Date | Milestone |
|---|---|
| 5/6/93 | 2.4 Gb/s OC-48 SONET backbone operational Penn <=> Bellcore |
| 5/7/93 | End-to-end data between workstations at Penn and Bellcore, interoperating Penn and Bellcore ATM host interfaces |
| 5/19/93 | Sunshine switches ATM cells between IBM RS/6000 at Penn and IBM RS/6000 at Bellcore |
| 6/7/93 | Penn and Bellcore ATM interfaces interoperate through Sunshine |
| 6/8/93 | End-to-end video over ATM from Penn workstation w/Penn video card to Bellcore workstation display |
| 10/26/93 | 2nd Sunshine operational, at Penn |
| 11/12/93 | Full-motion A/V teleconference over PTM/SONET, Penn <=> IBM |
| 12/31/93 | 25 Mbps TCP/IP over AURORA switched loopback |
| 2/25/94 | "Cheap Video" ATM appliance running over AURORA |
| 3/15/94 | "TeleMentoring" interactive distance learning over AURORA Penn <=> Bellcore using Cheap Video NTSC/ATM |
| 3/30/94 | 70 Mbps TCP/IP over AURORA between RS/6000s |
| 4/17/94 | MNFS/AIX solving differential heat equations over AURORA |
| 4/21/94 | Avatar, w/audio VC operational Penn <=> Bellcore, **AND** IBM PVS IBM <=> over PlaNET, **AND** VuNet Penn <=> MIT |
| 5/6/94 | Avatar in operation Penn <=> MIT |
| 12/31/94 | Link to IBM and MIT taken out of operation |
| 7/5/95 | HP PA-RISC/ Afterburner ATM Link Adapter achieves 144 Mbps TCP/IP |
| 8/22/95 | ATM Link Adapter achieves 215+ Mbps TCP/IP |

*Table I: AURORA Gigabit Testbed: Selected Milestones*

A major concern with advanced applications such as medical imaging and teleconferencing is privacy. Privacy transformations have traditionally been rather slow due to the "bit-complexity" of the substitution- and confusion- introducing operations. An augmentation of the network host interface with cryptographic hardware was designed [20]. It was based on observations by Broscius, *et al.* [3], which describes the use of parallelism to achieve high performance in an implementation of the NBS Data Encryption Standard. The board was implemented and achieved a measured performance of 100 Mbps. Among the interesting features were the use of GaAs PLAs for the substitution boxes in the cipher and a scheme for unrolling the embedded loops using multiple instances of the hardware. The difficulty of getting data to and from the encrypting hardware through a bus remained. Smith, *et al.* [20], describes the history and motivation for the architecture, and explains how to insert a high-performance cryptographic chip (for example the VLSI Technologies VM007 DES chip, which operates at 192 Mbps) into the ATM host interface architecture. The resulting system is able to operate at full network speed while providing a per-cell ("agile") per-VCI rekeying; both the performance and the operation are transparent to the host computer, while providing much greater key control than possible with link encryption approaches.

Traw, *et al.* [24], describes one of the two earliest workstation host interfaces for ATM networks, both done in AURORA. This interface chose an all-hardware solution, with careful

separation of functions between hardware and software implementation. Traw, *et al.* [25], reports on the implementation of the ATM host interface and its support software. The architecture is presented in detail, and design decisions are evaluated. Later work [21] focused attention on the adaptor to application path through software, and some of the key design decisions embedded in the software are examined. Of particular interest are the system performance measures where the adaptor operates with a significant application workload present.

The initial software subsystem provided an application programmer interface roughly equivalent to a raw IP socket, and was able to achieve more than 90% of the hardware subsystem's performance, thus driving an OC-3 at its full 155 Mbps rate. Key innovations were the reduction of data copying (through use of VM support - this direction was later followed by others, including the U. Arizona team [7] designing software for the Osiris [6] interface developed at Bellcore by Bruce Davie) and the partitioning of functions between hardware and software. As can be seen from *Table II* [8], this reduction in data copying was necessitated by the memory bandwidth limitations of early-1990's workstations.

|  | Memory (Mb/s, peak) | CPU/Memory (Mb/s, sustained) | | |
|---|---|---|---|---|
|  |  | Copy | Read | Write |
| IBM RS/6000 340 | 2133 | 405(.19) | 605(.30) | 590(.28) |
| Sun SS10/30 | 2300 | 220(.10) | 350(.15) | 330(.14) |
| HP 9000/720 | 1600 | 160(.10) | 450(.28) | 315(.20) |
| Dec 5000/200 | 800 | 100(.13) | 100(.13) | 570(.71) |

*Table II:* **Workstation Memory Bandwidths** (as tabulated by Druschel, *et al.*)

The bottleneck on the IBM RS/6000 was initially the workstation's system bus to I/O bus interconnect [25], however, improvements to the I/O subsystem architecture moved the bottleneck to the physical link. For the HP PA-RISC implementation [26], designed to demonstrate scaling of the host interface architecture to higher speeds, the bottleneck was the bus attachment (for this environment the SGC graphics bus served as the attachment point).

The HP PA-RISC/Afterburner ATM Link Adapter held a record for highest reported TCP/IP/ATM performance of 215+Mbps for almost one year. This performance was measured between two HP PA-RISC 755s, connected by a 320 Mbps SONET-compliant null modem, using the `netperf` test program. The best performance was achieved using a 32KB socket buffer size and 256 KB packets.

Custom physical layer interfaces were implemented as daughter cards so that alternate physical layers (e.g., AMD TAXI and HP GLINK [13] ) could be explored within the context of the AURORA testbed. The GLINK implementation allowed low-cost distribution of SONET-rate ATM over twisted pair in networks which are about the diameter of a laboratory work area (50 fit.); coaxial cable extends the operational limitations of electrical GLINK to 300 ft.

Software for the IBM RS/6000 ATM interface was enhanced by the addition of TCP/IP support [1], implemented as a Common I/O (CIO) loadable device driver. which allowed us to operate at 70 Mbps sustained over the testbed. For the AURORA testbed, this was the first and fastest operational TCP/IP which carried traffic over the WAN. It has been used since to carry MNFS distributed shared memory traffic over the testbed between Penn and Bellcore. When the IP is used as a component of the UDP/IP protocol stack, over 90 Mbps were obtained on an RS/6000 Model 580 connected to an RS/6000 Model 530.

Traw and Smith showed that host interfaces could be aggregated in a number of manners to support multiples of the bandwidth provided by a single adapter [27]. The results of a simulation study showed that for hardware implementations, striping at the byte or ATM cell level might be appropriate; in this model the host adaptor would provide a PDU interface to the host and perform the striping transparently; Bellcore's Osiris interface performed cell-striping and the IBM SIA performed byte-striping. A software-implemented solution would stripe most effectively by using multiple interfaces to send multiple concurrent IP packets; then TCP/IP's facilities for in-order delivery of packets would compensate for the skew between links.

## 3. MULTIMEDIA ARCHITECTURES

Multimedia architectures for gigabit endworking must be designed with scale, endpoint heterogeneity, and application requirements as the key driving elements. We devised an integrated multimedia architecture with which applications define which data are to be bundled together for transport and select which data are unbundled from received packages. This allows sources to choose the degree of resource allocation which they wish to provide; receivers choose which elements of the package they wish to produce. While potentially wasteful of bandwidth, the massive reduction in the multiplicity of customized channels allows sources to service a far greater number of endpoints and receivers to accommodate endpoint resources by reproducing what they are capable of. The scaling advantage of this approach is that much of the complexity of customization is moved closest to the point where customization is necessary - the endpoint.

Multimedia work included the development of custom hardware; for example an early video capture board used for experiments between Penn and Bellcore was developed [28]. Experiments with this MicroChannel Architecture adapter suggested that software-manipulated video would not operate with acceptable quality. This led to the all-hardware NTSC/ATM Avatar ATM appliance developed by Brendan Traw and Bill Marcus for use in TeleMentoring experiments linking Penn and Bellcore for purposes of undergraduate digital design projects focused on developing ATM hardware. The Avatar [11] card, which supports NTSC video and CD-quality audio, is the first example of an ATM appliance, with a parts cost of under three hundred dollars. Many of these cards were fabricated. They were used for distance teaching when connecting the Bellcore experimental Video Windows, for collaborative work between researchers at Penn and Bellcore, and for teleconferencing between Penn and MIT.

Much of the multimedia focus rested on the development of operating system abstractions which could support high-speed applications. These abstractions used the hardware and low-level operating system support developed for the IBM RISC System/6000 workstations equipped with the AIX operating system, an IBM implementation of UNIX. Key new ideas included a more general model of Quality of Service (QoS) requirements, and technical means for evaluating how any bandwidth allocation implementation requires support from the Operating System scheduling mechanism for true "end-to-end" service delivery. Nahrstedt [14] identified the software support services needed to provide Quality-of-Service (QoS) guarantees to advanced applications which control the characteristics of their networking system, and adapt within parameterized limits. These services form a "kernel", or a least common subset of services required to support advanced applications.

A logical relationship between applications-specified Quality of Service (QoS) [16], as well as operating system policy and mechanism, and network-provided QoS was developed. An example challenge is the kinematic data stream directing a robotic arm, which can tolerate neither packet drops nor packet delays - unlike video or audio, which can tolerate drops but not

delays. The approach used of a bidirectional translator (like a compiler/ uncompiler pair for a computer language) which resides between the network service interface and the application's service primitives. This can dynamically change QoS as application requirements or network capabilities change, allowing better use of network capacity, which can be mapped more closely to applications current needs than if a worst-case requirement is maintained. The implementation [14] outlined the complete requirements for such a strategy, including communication primitives and data necessary for translation between network and application. For example, an application request to zoom and refocus a camera on the busiest part of a scene will certainly require peer-to-peer communication between the application and the camera management entity. The network may need to understand the implications for interframe compression schemes and required bandwidth allocations. The translation method renegotiates QoS as necessary.

These ideas were described in Nahrstedt, *et al.* [15], which describes a mechanism to provide bi-directional negotiation of Quality-of-Service parameters between applications and the other elements of a workstation participant in advanced networked applications. The scheme is modeled on a "broker", a traditional mechanism for carrying on back-and-forth negotiations while filtering implementation details irrelevant to the negotiation. The QoS Broker reflects both the dynamics of service demands for complex applications and the treatment of both applications and service kernels as first class participants in the negotiation process. The QoS Broker was implemented in the context of a system for robotic teleoperation implemented over ATM in cooperation with Penn's General Robotics and Sensory Perception (GRASP) laboratory. The Broker was implemented and evaluated as part of a complete end-to-end architecture presented by Nahrstedt [14].

In the system, application requirements are determined by a negotiation protocol at startup. This turned out to be a major cost in the system, as the worst-case scheduler consumed considerable time in testing the feasibility of resource guarantees. Nonetheless, the system was capable of providing guaranteed services; a complete implementation, including a novel real-time protocol stack, is available in source-code form with anonymous FTP from `ftp.cis.upenn.edu`.

Gigabit multimedia is desired by the applications community; Bajcsy, *et al.* [2], describes the need for network support for a broader class of applications than audio/video. In particular, it has become clear that interaction with the physical world is among the most challenging applications for networking, as the QoS requirements for many systems will be sufficiently complex to cause interaction and competition between requirements. An example would be a tradeoff between throughput and reliability, which would tend one way for real-time video, while in the opposite direction for force-feedback data. The results could have considerable bearing on critical national challenges such as agile manufacturing, as software for a reliable gigabit network infrastructure providing end-to-end guarantees could be developed on the principles described in the thesis.

## 4. DISTRIBUTED SHARED MEMORY COMMUNICATIONS

Farber proposed distributed shared memory [9] (DSM) as a technology solution for integrating computation and communications more closely. This was one of the major investigations of the AURORA testbed, and the MNFS (for Mether-NFS) [12] distributed shared memory has been used to support applications over the AURORA WAN such as a parallel heat equation solver. There were four major questions we sought to answer in the experimental evaluation of DSM in the AURORA Project. Each of these were answered, as we outline below; a more sweeping perspective was given by Farber in his 1995 ACM SIGCOMM Award Lecture [9]. These four

research questions were:

1. **Is DSM a reasonable abstraction for distributed programming?** Yes, it is, as demonstrated by applications ported directly from shared-memory supercomputers. DSM is an abstraction for distributed applications programming. It has the ability to support programming with distributed control and shared data across a wide range of interconnected computer models. Distributed Shared Memory (sometimes also called Distributed Virtual Memory) is an interesting communications paradigm for gigabit networks. DSM may provide the best path for optimizing the construction of distributed systems requiring high-speed networking, especially where the traditional balance between network speed and processing speed has been inverted. The rationale is that memory management is well-understood, and that memory speeds represent the best case achievable for interprocess communication. A combination of cacheing and a cache policy known as prefetching can shape the traffic produced by the application.

2. **Can DSM work over WANs?** Yes, it can, and appears to work reasonably well, although many optimizations remain, such as better programming language interactions, cache management, techniques for pre-loading caches, and reductions in false-sharing due to data layout.

3. **What effect does increased bandwidth (e.g., Gbps) have on DSM performance delivered to applications?** Shaffer's thesis [19] showed that distributed shared memory was a viable technology for parallel applications even in a WAN environment. This speaks to the fundamental scientific questions about the relationships and tradeoffs between bandwidth and communications latency induced by propagation delay. A key insight was that for data-intensive applications, observed latency can be more a function of throughput than of physical propagation delay. This is due to the fact that as Protocol Data Units (PDUs) are used at higher levels of protocol architectures, the PDU does not ''arrive'' until its last bit has arrived. This means that throughput has a significant effect on latency observed at any layer other than physical, where the PDU can be considered to be a bit.

4. **What effect does combining high bandwidth and high delay (in high** *bandwidth \* delay* **product) networks have on the DSM performance delivered to applications?** The key issue in the testbed specialization of the Distributed Shared Memory paradigm was the effect of increased propagation delay on application performance. Shaffer demonstrates [19] that application-measured latency is a function of both propagation delay and the *throughput delivered end-to-end*. While the propagation delay is clearly a fundamental limit given speed-of-light limitations and the like, it may not be the dominant cost. The consequence of this is that high bandwidth networks can reduce delay simply by reducing the latency component associated with throughput. This is especially true of data objects of a large enough size to be affected by throughput considerations - for example virtual memory page sizes are typically 4096 or 8192 bytes. The DSM experiments on the testbed itself required the entire experimental ATM infrastructure built for AURORA. After the Bellcore to Penn span was terminated, an experimental output port controller [10] (OPCv2) designed by Bill Marcus, was programmed to emulate selected delay characteristics. The experimental configuration studies the effect of a large *bandwidth \* delay* link by replacing an MNFS client machine connected via Ethernet LAN to one connected through the ATM WAN, either directly (when the WAN operated) or by emulation with OPCv2. A parallel computation, a heat equation solver, uses a central difference method solution. This problem is extremely computation-intensive, since the computational complexity of matrix solution is at least

quadratic in terms of the problem size.  **Figure 3** plots completion time for a large problem instance (1200 by 1000 elements) against the delay induced by the OPCv2 for the ATM-connected host.
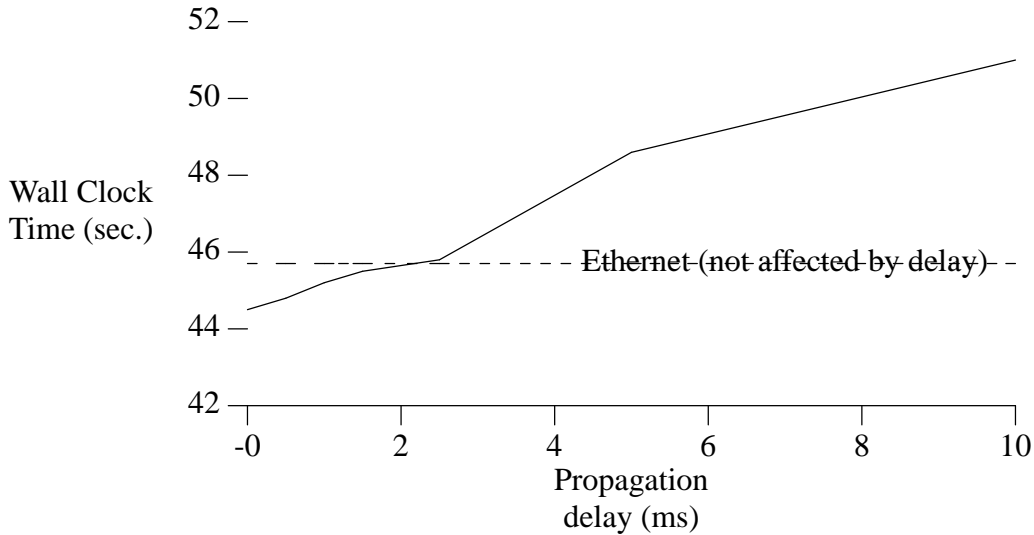


**Figure 3:**  Performance of distributed heat equation solution, DSM/ATM

The key observations to make are that the ATM solution outperforms the Ethernet solution, with the same problem on the same software on the same machines, for a variety of emulated distances. Each millisecond is equivalent to 130 miles of fiber distance; thus a 1 millisecond delay, where the ATM configured system outperforms the Ethernet LAN, represents the distance to Bell Communications Research from Penn. The delay measured to Bellcore and back through the AURORA WAN was 2.1 milliseconds, or equivalent to 1.05 milliseconds on this plot.  The measured completion times for the computation are consistent between the real and emulated environments.

The experiments have shown that parallel applications running over the AURORA infrastructure execute as quickly as when run over a local Ethernet LAN, giving one data point in the space of bandwidth*delay tradeoffs. The OPCv2 allows the space to be explored more completely once the two measurements from the testbed configuration are used to anchor any LAN-based emulations to true WAN delay values.

## 5.  CONCLUSIONS

*"There is an old network saying: Bandwidth problems can be cured with money. Latency problems are harder because the speed of light is fixed -- you can't bribe God."*  - D. Clark [18],

The AURORA gigabit testbed research had a fundamental influence on the design and development of gigabit network technology.

The ATM host interface work answered concerns about the viability of ATM Segmentation-and-Reassembly (SAR).  It is now overwhelmingly clear that ATM SAR can operate at Gigabit/second rates, and thus the performance concerns expressed when the testbeds were begun were largely non-issues.  ATM host interface hardware developed in AURORA has influenced all

available commercial products, which resemble either the Penn ATM host interface with hardware SAR (though implemented with an ASIC [4] rather than PALs) or the Bellcore Osiris (with software-directed SAR). Operating systems research at Penn and later work at the University of Arizona showed how to reduce copying between host interfaces and applications through careful management of DMA, buffer pools and process data access semantics; these ideas are now appearing in software from vendors such as Sun Microsystems [4] and Silicon Graphics. It is thus also clear that operating systems can deliver gigabit range throughputs to applications with appropriate restructuring and rethinking of copying and protection boundary-crossing. Issues we identified at Penn such as IPC latency, are now being studied by others such as Cornell University, using commercial ATM host interfaces; this work has had a considerable effect on the operating systems community -- several SOSP and USENIX papers have been stimulated by it. Among our other ATM contributions was a collaborative effort with Bellcore that produced an early ATM appliance, the Avatar NTSC/ATM video board developed at Penn and Bellcore for TeleMentoring.

The still unanswered questions revolve around the discovery and evaluation of mechanisms that deliver a practical reduction in latency to applications. These include better cache management algorithms for distributed shared memory as well as techniques for lookahead referencing, or prefetching. Another important area is the reduction in operating system induced costs in network latency; the software overhead is equivalent to about 200 miles of fiber in the AIX implementation Penn did for AURORA. Of course, it should be noted that the primary target was high applications *throughput*.

The multimedia work in the gigabit networking research community has had impact on the operating systems and robotics communities. It has also pointed out some issues to be avoided in adapter design, such as head-of-line blocking observed in serial DMA of large ATM AAL3/4 CS-PDUs on the IBM RISC System/6000 system. The work influenced industry (particularly IBM Heidelberg) and has been covered in a reference work on multimedia [23].

The DSM work remains controversial in the systems community as custom and inertia make new ideas slow to accept. The important observation we can draw from the work in AURORA is that if the mechanism becomes more widely accepted, there are now algorithms which can aid in the location and prefetching of data developed, and significant experimental evidence supporting the hypothesis that higher network bandwidths can aid distributed applications of any type to achieve higher performance.

## 6. REFERENCES

[1] D. Scott Alexander, C. Brendan S. Traw, and Jonathan M. Smith, "Embedding High Speed ATM in UNIX IP," in *USENIX High-Speed Networking Symposium*, Oakland, CA (August 1-3, 1994), pp. 119-121.

[2] Ruzena Bajcsy, David J. Farber, Richard P. Paul, and Jonathan M. Smith, "Gigabit Telerobotics: Applying Advanced Information Infrastructure," in *1994 International Symposium on Robotics and Manufacturing*, Maui, HI (August 1994).

[3] Albert G. Broscius and Jonathan M. Smith, "Exploiting Parallelism in Hardware Implementation of the DES," in *Proceedings, CRYPTO 1991 Conference*, ed. Joan Feigenbaum, Santa Barbara, CA (August, 1991), pp. 367-376.

[4] J. Chu, "Zero-Copy TCP in Solaris," in *Proceedings, USENIX 1996 Annual Technical*

*Conference*, San Diego, CA (Jan. 22-26, 1996), pp. 253-264.

[5] 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).

[6] Bruce S. Davie, ''The Architecture and Implementation of a High-Speed Host Interface,'' *IEEE Journal on Selected Areas in Communications (Special Issue on High Speed Computer/Network Interfaces)* **11**(2), pp. 228-239 (February 1993).

[7] P. Druschel, L. L. Peterson, and B. S. Davie, ''Experiences with a High-Speed Network Adaptor: A Software Perspective,'' in *Proceedings, 1994 SIGCOMM Conference*, London, UK (Aug. 31st - Sep. 2nd, 1994), pp. 2-13.

[8] Peter Druschel, Mark B. Abbott, Michael A. Pagels, and Larry L. Peterson, ''Network Subsystem Design,'' *IEEE Network* **7**(4), pp. 8-17, Special Issue: End-System Support for High-Speed Networks (Breaking Through the Network I/O Bottleneck) (July 1993).

[9] David J. Farber, *The Convergence of Computers and Communications - Part 2*, ACM SIGCOMM Award Lecture, August 30, 1995.

[10] W. S. Marcus, ''An experimental device for multimedia experimentation,'' *IEEE/ACM Transactions on Networking*, to appear (1996).

[11] William S. Marcus and C. Brendan S. Traw, ''AVATAR: ATM Video/Audio Transmit And Recieve,'' DSL Technical Report (March 1995).

[12] R. G. Minnich, ''Mether-*NFS*: A Modified NFS which supports Virtual Shared Memory,'' in *Experiences with Distributed and Multiprocessor Systems (SEDMS IV)*, USENIX Association, San Diego, CA (1993), pp. 89-107.

[13] A. M. Moore and C. B. S. Traw, ''GLINK as a Solution for Local ATM Distribution,'' in *Proceedings of the 1995 Design SuperCon conference on digital communications design*, San Jose, CA (February 1995), pp. 5:1-5:20.

[14] Klara Nahrstedt, ''An Architecture for Provision of End-to-End QoS Guarantees,'' Technical Report, CIS Dept., University of Pennsylvania (1995). Ph.D. Thesis

[15] Klara Nahrstedt and Jonathan M. Smith, ''The QoS Broker,'' *IEEE Multimedia Magazine* **2**(1), pp. 53-67 (Spring, 1995).

[16] Klara Nahrstedt and Jonathan M. Smith, ''Design, Implementation and Experiences of the OMEGA End-Point Architecture,'' *IEEE Journal on Selected Areas in Communications (Special Issue on Multimedia Systems)*, (to appear) (1996).

[17] Craig Partridge, *Gigabit Networking,* Addison-Wesley, Reading, MA (1993). ISBN 0-201-56333-9

[18] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach (2nd Ed.),* Morgan Kaufmann, San Francisco, CA (1996).

[19] John H. Shaffer, ''The Effects of High-Bandwidth Networks on Wide-Area Distributed Systems,'' Ph.D. Thesis, University of Pennsylvania CIS Dept. (1996).

[20] Jonathan M. Smith, C. Brendan S. Traw, and David J. Farber, ''Cryptographic Support for a Gigabit Network,'' in *Proceedings, INET '92*, Kobe, JAPAN (June 15-18, 1992), pp. 229-237. (Inaugural Conference of the Internet Society)

[21] Jonathan M. Smith and C. Brendan S. Traw, "Giving Applications Access to Gb/s Networking," *IEEE Network* **7**(4), pp. 44-52, Special Issue: End-System Support for High-Speed Networks (Breaking Through the Network I/O Bottleneck) (July 1993).

[22] Computer Staff, "Gigabit Network Testbeds," *IEEE Computer* **23**(9), pp. 77-80 (September, 1990).

[23] Ralf Steinmetz and Klara Nahrstedt, *Multimedia: Computing, Communications, and Applications,* Prentice-Hall, Englewood Cliffs, NJ (1995).

[24] C. Brendan S. Traw and Jonathan M. Smith, "A High-Performance Host Interface for ATM Networks," in *Proceedings, SIGCOMM 1991*, Zurich, SWITZERLAND (September 4-6, 1991), pp. 317-325.

[25] C. Brendan S. Traw and Jonathan M. Smith, "Hardware/Software Organization of a High-Performance ATM Host Interface," *IEEE Journal on Selected Areas in Communications (Special Issue on High Speed Computer/Network Interfaces)* **11**(2), pp. 240-253 (February 1993).

[26] C. Brendan S. Traw, *Applying Architectural Parallelism in High Performance Network Subsystems,* CIS Dept., University of Pennsylvania (1995). Ph.D. Thesis

[27] C. Brendan S. Traw and Jonathan M. Smith, "Striping within the Network Subsystem," *IEEE Network*, pp. 22-32 (July/August 1995).

[28] Sanjay K. Udani, *Architectural Considerations in the Design of Video Capture Hardware,* University of Pennsylvania, School of Engineering and Applied Sciences (April, 1992). M.S.E. Thesis (EE)