LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

LA-UR -89-3155

CONF-8910309--2

LA-UR--89-3155

DE90 001839

TITLE MULTIPLE CROSSBAR NETWORK: A SWITCHED HIGH-SPEED LOCAL NETWORK

AUTHOR(S) RANDY L. HOEBELHEINRICH
RICHARD G. THOMSEN

SUBMITTED TO 14TH CONFERENCE ON LOCAL COMPUTER NETWORKS
MINNEAPOLIS, MINNESOTA
OCTOBER 10-12, 1989

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

MASTER

Multiple Crossbar Network
A Switched High Speed Local Network

Randy Hoebelheinrich **Richard Thomsen**

Computer Network Engineering
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, New Mexico

Abstract

The Multiple Crossbar Network (MCN) is a prototype High-Speed Local Network at the Los Alamos National Laboratory. It will interconnect supercomputers, network servers and workstations from various commercial vendors. The MCN can also serve as a backbone for message traffic between local area networks. The MCN is a switched local network of switching nodes called Cross-Point Stars (CP*s). Hosts and CP*s are connected by 800-Mbit/s (100-Mbyte/s) point-to-point ANSI High-Speed Channels. CP*s include RISC-based network protocol processors called Crossbar Interfaces and a switching core called the Crossbar Switch. Protocols include physical, data link, intranet, and network access functionality. Various internet and transport protocols are intended to run above the MCN protocol suite. A network management and simple naming service is also included within the Los Alamos Network Architecture. Immediate applications include visualization. The MCN is intended to also serve as a framework for multicomputer applications.

The Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36. This work was performed under auspices of the U.S. Department of Energy.

Introduction

Parallel processing is being utilized in ever increasing ways to benefit applications in computing. As a collection of processors, a computer network is another means of applying parallelism to increase performance in computer applications. This can be accomplished in two broad areas including providing the interconnection network for cooperative computation on a large scale and using parallelism in the very elements of the communication subnet that provide that service.

A modular, integrated interconnection network of fast packet switches forms the basic building blocks for a rich set of topologies useful in cooperative computation on a large scale. Allowing the addition of specialized nodes by means of a standard interface will also provide the flexibility of a heterogeneous environment. Supporting a loosely coupled, coarse-grained, extensible network framework will be a powerful, versatile tool for investigations into multicomputer applications.[7,10,12,18]

The primary purpose of a communication subnet is high performance and reliable transfer of information from a source to an intended destination or destinations. The relative importance of these characteristics varies with the instance of the information transfer. Achieving these goals occurs by means of the methods and equipment used. The focus here will deal with current aspects of a prototype network architecture to increase network performance for a local set of computers at Los Alamos National Laboratory.

Motivation for a prototype high speed network came from four areas. First, after 15 years the local computer network of the Central Computing Facility of the Integrated Computing Network (ICN)[5] at Los Alamos Alamos National Laboratory (LANL) had served the laboratory well. However, work needed to be done on a next generation network. Secondly, the technology was available to implement some innovative switching concepts. Thirdly, significant work and progress in the areas of interconnection networks,[2,3,8,9,11,13] multiprocessors[4,14,16,19,21] and parallel computation allowed use of these ideas both in the network proper and in new machines to be attached to the network. Finally, applications in complex systems, distributed simulation [31] and particularly visualization required a framework to support efforts in these fields.

The following two sections will provide an overall view of the goals and major architectural elements of the MCN. Sections four and five will describe the hardware components followed by software protocols and services. A discussion of the current state of implementation and chronology of events leading up to that point will compose section six. Finally, a brief look into future research, applications and development will conclude the paper.

MCN Goals

There are four general goals applicable to the MCN. They are performance, reliability, flexibility and security. Each of these goals have been met in some form in our current network, however potential for vast improvement in performance for example warranted a new approach to networking.

Performance

Given the fundamental purpose of transferring data between two points, increasing performance translates into lowering network latency. The core computer network at Los Alamos is a packet switched network connecting heterogeneous commercial hosts with 50 Mbit/s 16 bit parallel point-to-point channels.[6] Major components affecting network latency are data copying from user to network space,[35] data movement into and out of the hosts, the number of channels, bandwidth on a single channel, switch latency and protocol processing both on hosts and switches. We chose to concentrate on a higher-speed channel, special purpose protocol processors and an intermediate crossbar switch as the basic building blocks for a next generation packet switch network. Details and the reasoning for these decisions will be discussed later.

Reliability

Four goals for increased reliability were included in the architecture. These goals were to develop a combination of hardware and software for a more robust system of fault isolation, followed by the use of information available from this system in a proactive rather than a reactive network management policy. An introduction of increased condition and error information exchange between internal protocol layers [22,23] and a rich connection scheme of channels to provide both alternate and redundant paths in the network would combine to move error information where it was needed.

Flexibility

Flexibility as a general goal was important at several levels. The most visible goal was a standard high-speed channel for commercial vendor access to the network. In addition, a standard high speed network access protocol was desirable. It was felt to be premature to standardize a network access protocol considering the need for research and experience with the new network. The final requirement centered around the use of a switching framework which allowed a myriad of options for different topologies, reconfiguration [28,29] and levels of indirection between logical and physical addresses. This will allow use of the MCN as a supercomputer framework or backbone for geographically distributed local area networks at Los Alamos.

Security

Security has always been a requirement of paramount importance. The interesting point about security in a new architecture was the opportunity to make it an intimate and integral part of the services expected to be a part of this system. This opportunity has not been overlooked and has received considerable attention. A significant amount of effort remains in this area.

Immediate Goals

The most immediate goals for the MCN are proof of concept for the switch we called the Crosspoint Star or CP*, specification and implementation of the High-Speed Channel [1] and the establishment of a small testbed. (Figure 9) Future goals are discussed later and are included more for perspective and as possibilities than as any commitment on our part. The potential for areas of investigation is so great that we are forced to limit our scope to the immediate goals above.

MCN Overview

The MCN may be composed of any number of packet switches called CP*s (Figure 1). CP* is a loosely connected set of protocol processors called Crossbar Interfaces (CBI) which surround and are connected to a crossbar switching core called the CBS. The connections are High-Speed Channels (HSC).[1] The CBIs are boards containing a RISC-based processor, 2 HSCs of opposing direction and 1 Mbyte of 4 register VRAM. The purpose of the CBI is to act as a protocol processor either strictly for routing internal to the network or as an intelligent network interface for hosts at the boundary of the network. The CBS is a stack of crossbar chips with HSC interfaces attached to the pins. This allows a full HSC cable to pass-through the crossbar core. The CBS also has a controller which acts as the arbiter for connection requests from a source HSC through the crossbar to a destination HSC. The controller acts on one word of information to implement a physical switching capability. No further intelligence exists on the CBS controller. The HSC is a simplex 800/1600 Mbit/s 32/64 bit parallel copper channel. Work is now ongoing in ANSI to specify a fiber channel which includes HSC. It must be pointed out that use of the HSC with an intermediate switch is not and has not been proposed as a standard. We have chosen to take advantage of a characteristic of the HSC connection sequence to implement a physical switching capability.

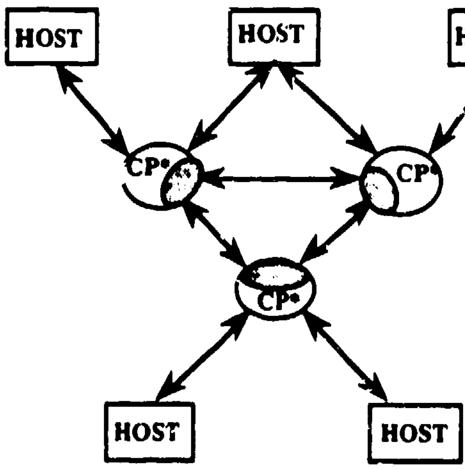

Figure 1. Multiple Crossbar Network

Protocols of the MCN include a data link in the CBI which uses routing information for the HSC I-Field at HSC connect time to select a destination HSC. The algorithm to select a destination specifies either an absolute channel address(es) in the I-Field for use by the CBS or a multiple address to channel bitmap for use by the CBS. Internally to the MCN we have specified an intranet routing protocol which is a connectionless service. A network access protocol which resides on the host and a CBI attached to the host currently specifies a connectionless service also. Provision has been made to provide a connection oriented service at a later date. In addition, consideration has been given to combining the network access and intranet protocols into a hybrid cut-through protocol using messages rather than packets or connections. This would essentially be a synthesis of connection and connectionless data transfer services.

Services of the MCN will initially include a nameserver and network management. The nameserver was intended to act as a network server which keeps track of addresseable entities. These entities include hosts, servers, devices, complex data types such as a data base, sessions and users. By requiring each of these entities to become known by introduction at an access point of a fixed physical address, entities or objects can be dynamic (i.e. move about), associated in a hierarchical fashion to some other object and by virtue of location and topology have a computable route to any other object in the system. By assigning capabilities to these objects, access controls and allocation algorithms can be applied by a managing object, the distributed operating system. The end result is a foundation of two services which will evolve and have the potential of combining directory, security, resource allocation and network management functionality into an object-oriented distributed operating system.[4,32,33,34]

MCN Hardware

It is useful to compare a view of a packet switch similiar to the current switches used in the ICN (Figure 2) with an approach to what eventually became CP* (Figure 3). The idea was to distribute and allow parallel efforts wherever possible. The function of the switch of course was to direct packets from an input device to the appropriate output device. The CPU and memory were central to and shared by all devices. Access to the CPU and memory was serialized on a bus. So the question was asked, why not replace the bus with a general interconnection network like a crossbar and allow each device a local processor and memory? This would allow independent I/O and protocol processing with potential for parallel and simultaneous data transfers through the switch. What was needed was a method for each processor to arbitrate a path through the crossbar. Since any source could transmit to any destination, either a distributed or centralized access method was necessary. Given available technology a centralized controller for the crossbar was chosen. We already had the need for a high-speed channel. The channel was designed with the ability to present destination selection information to the crossbar controller during the initial connection sequence of the channel. The controller could then use this information to select and access an available destination channel. The only remaining central path would then be the controller on the crossbar. The result was CP*.

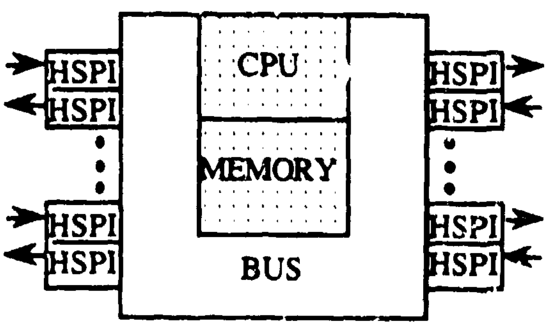

Figure 2. Centralized Packet Switch

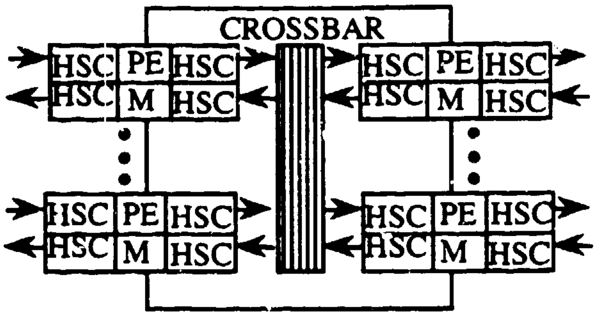

Figure 3. Distributed Packet Switch

Crosspoint Star

The CP* can be thought of as a packet-switching node made up of three major elements (Figure 4). These elements were: (1) the channel or HSC, (2) the packet switch interconnection network or CBS, and (3) the packet processor or CBI. CP* was designed to increase performance by having distributed special-purpose protocol processing on each channel and incorporating physical layer switching between these processors. The physical layer switching is accomplished on an HSC with the aid of an intermediate CBS controller. This CBS is strictly dedicated to switching links to minimize switching latency and provide fast packet switching. The distributed protocol processors or CBIs provide optimal packet throughput. By using this design, we will in a broader sense, have distributed the functions of the traditional packet-switching node over many processors and controllers at the channel end points as well as "on the wire." This is replicated on all links for parallel simultaneous transfers at any CP* node. Finally, CP* forms the lower level of a hierarchy of general interconnection networks. This hierarchy is formed by a uniform group of CP*s with the potential of collecting these groups into a larger network. A flexible address space will then allow partitioning of the network into multi-level, varying size, sets of hosts.

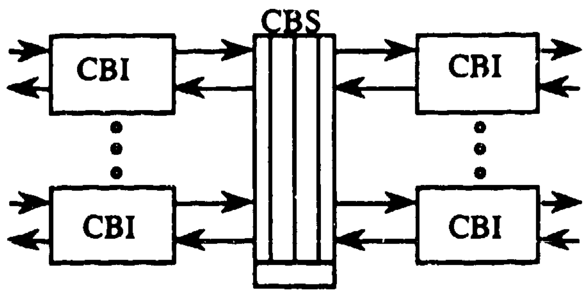

Figure 4. Crosspoint Star

High Speed Channel

There were a number of goals for the HSC. The most important goal was high speed. Another goal was to keep it simple. An 800-Mbit/s channel already existed on the Cray computers, so matching that data rate seemed appropriate. We also knew from experience wth the High-Speed Parallel Interface (HSPI) [6] at Los Alamos that standardizing the HSC interface for vendor implementations was highly desirable. The question was, would industry see sufficient need for such a high-speed point-to-point channel to standardize it? History will show there was considerable interest. Another goal was to move some data link functionality into the channel. In specifiying a new parallel channel, we had the opportunity to incorporate this data link functionality into the signalling. These functions included framing and flow control. Flow control was handled using a Ready signal at the destination HSC. Framing was specified by signals defined for multi-word burst(s), packet(s) and a physical connection. Error detection and notification were also items the HSC could do for us. For ease in implementation a VRC/LRC error detection scheme with a length field was chosen with the ability to notify a controlling entity of these errors. Finally, and this is the most interesting goal, physical layer switching by means of an intermediate controller on the HSC was needed. This would provide an integrated service of packet transfers on a directed connection and prove crucial in the overall architecture of a new high-speed network. Figure 5 illustrates possible HSC configurations.

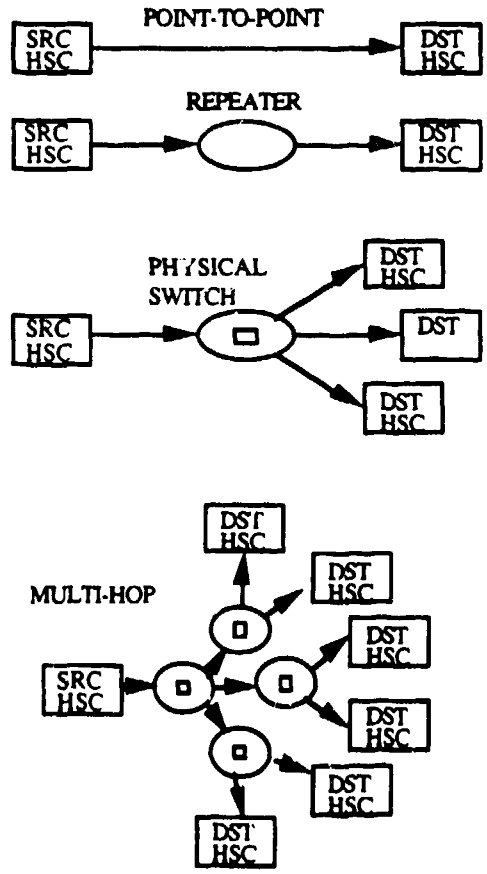

Figure 5. HSC Configurations

Crossbar Switch

The sole purpose of the CBS is to connect a Source HSC to a requested Destination HSC (Figure 6).This minimized switch latency. The request is made by a single parameter supplied by the Source HSC at connect time. The controller need only interpret the I-Field of the HSC connect sequence to select a destination HSC. The first prototype CBS controller polls HSC Request signals. Upon seeing a Request for connect the controller commands the switch to connect the source HSC to the destination HSC. If the destination is busy it is possible to specify and select alternate or redundant paths due to a regular network topology [9,10] and understanding of a structure of redundant paths by the intranet routing protocol. There are potential problems for deadlock with this method. [20] An alternative is for the CBI to back off and try again after an HSC Request timeout. Minimum latency through the switch from this process is 350ns. All channels are polled except channel 0. Channel 0 is used by the controller and unavailable for connections. One of the more difficult problems was designing the interface connector and pin arrangement for the switch. The unit consists of an array of crossbar chips mounted 1 chip to a board with an HSC interface using 1 pin on each of the chips. The first prototype is a 48 bit 16X16 crossbar with plans for a 32X32 crossbar after that. Fiber optics will be used in future implementations.

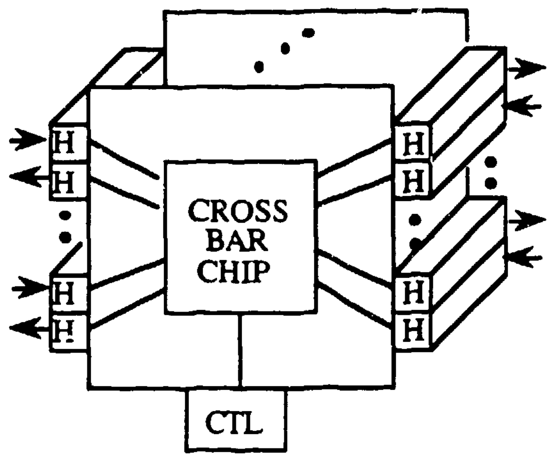

Figure 6. Crossbar Switch Core

Crossbar Interface

The CBI hardware (Figure 7), which has been designed and built by Digital Equipment Corporation (DEC), is a protocol processor for CP* [15]. The CBI's primary purpose is to act as a specialized processor and buffer for packets traversing the network. This will lower protocol processing overhead on the switches and offload protocol processing from hosts using the network. Use of an AM29K RISC processor, four register VRAM and hardware FIFOs streamlines packet processing by distributing the workload to two HSC packet streams. For further details on the CBI see the DEC paper referenced above.

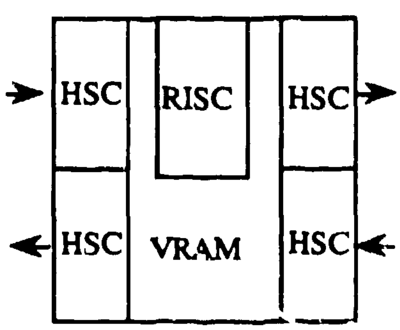

Figure 7. CBI Hardware View

MCN Software

Services and Protocols

Services and protocols provided for the MCN include the physical HSC protocol [1], a data link protocol over the HSC with a channel access capability, intranet routing and network access protocols, and both a network management and name-server capability. See Figure 8 for a software view of the CBI.

Link Configurations

The link configurations in the MCN are of two general types. One is a point-to-point simplex HSC. The second is a point-to-point simplex HSC with an intermediate CBS between the source and destination HSC entities. It is helpful to view this latter configuration from the perspective of three different elements. These elements are the HSC, the source data link entity, and the intermediate crossbar switching core. The HSC is viewed as a point-to-point link where the switching core is transparent. The data link entity will view the link as a multipoint configuration with physical switching between several HSCs. The crossbar switching core will also view the set of HSCs as a multipoint topology. In all cases, the duplexity of the link is simplex.

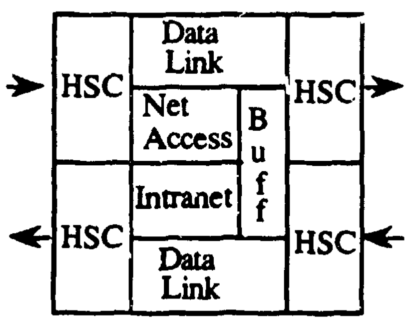

Figure 8. CBI Software View

Data Link

The data link for the MCN provides a connectionless data transfer service using HSC physical connections between data link entities. The data link will use physical address to alternate or redundant channel mapping and the HSC connection sequence as a channel access control mechanism. This is accomplished by utilizing the underlying HSC, CBS, and I-Field to access one of many possible destination HSCs. The data link entity should be viewed as contending for an available HSC in a multipoint configuration of data link entities. The data link entity does not control multiple HSCs simultaneously. It does not, therefore, provide a downward multiplexing or splitting capability. The data link is able to use the framing capability of the HSC to provide a hierarchy of data transfer units ranging from a short burst of less than 1 Kbyte to multi-packet connections or packet trains. The data link is not capable of multi-hop paths without the aid of a binary routing word [9] utilized by the intermediate switches. See Figure 5. Providing this routing word and knowing the network topology is a functin of the intranet protocol.

Intranet

The intranet accepts packets from a network access entity at the boundary of the network and transfers these packets over a series of HSC links to a destination network access entity. The intranet is a local connectionless data transfer service. Its primary function is to take each packet and determine the routing for the packet. The intranet receives a destination physical network address from the source network access entity when it is given a packet to transmit. This address is used for link selection over the CBS. Selecting these links involves more subtlety than knowing the physical route. It may also mean knowing which portions of the subnet may be traversed. [22] Important areas of investigation regarding this protocol will be in partitioning [9] and masking [25] for purposes of allocation or capability routing. Routing based on capability would require a close relationship with the overlying directory and access control services of the network.

Network Access

The purpose for a distinct network access protocol in the MCN is to provide a memory-to-memory transfer of data between logical network hosts using very large blocks of data. For our present purposes the network access accepts packets from a transport entity and transfers these packets through a data link to a local destination via the intranet. Transport protocols will be used for reliable end-to-end transfers. A separate network access protocol provided a means to hide the workings of the internal subnet from communicating hosts and allow an explicit security and access boundary for our network. Security and access information are checked after entry to and before exit from the MCN. This will provide insulation between the attached host and network.[21] We also want to provide a local reliable data transfer which could take advantage of future enhancements in computer architectures and network interfaces. The host-network interface is an important area of investigation from a performance and architectural point of view. Considerable work will be needed at this interface to address issues of performance and reliability from the physical media aspect all the way to the application layer.

Internet and Transport

Initially, we considered implementing protocols up to and including the transport layer on the CBIs. For reasons of expediency and timeliness we chose not to implement these protocols on the CBI. We have provided the flexibility to allow any internet or transport protocol to access the MCN through the network access service interface.

Name Server

The primary functions of the name-server service are twofold. First, it is used to provide a name to logical address translation database for name translations on the network access portion of a host and logical to physical address translations for the intranet protocol. The latter capability is designed to allow flexibility for configurations and system movement by adding a level of indirection to addresses. Second, the naming service will allow introduction, maintenance, access control, and accountability of objects that are or become part of the network. These objects include, hosts, processes, users, and user sessions.

Network Management

Network management will consist of the capability to address CBIs in the MCN for purposes of diagnostics, fault isolation,[26] configuration management, gathering statistics and tracing or logging activities. Network management is considered central to the architecture of the MCN, but will not be dealt with further in this discussion.

Implementation

What has been accomplished at this point? HSC or HSC-like implementations have been completed and, in some cases, announced as products by various commercial interests. DEC has built four CBIs and installed them at Los Alamos where they are undergoing testing with the Los Alamos CBS and HSC interface. The first level of testing included passing data without checking by the RISC processor and HSC interfaces on the CBI. The CBS has also been included in the loop and demonstrated to work. We have also transferred a graphics image from an IBM 3090-600 through a CBI to a display. We expect to have CBIs communicating to attached hosts by late 1989. DEC has implemented data link, intranet and network access protocols on the CBI. These protocols still need to undergo extensive testing. The network management and naming services that reside above the intranet and network access protocols are being designed and expected to be implemented at a later date. The HSC drivers are currently under various stages of design, implementation and testing by commercial vendors. Implementations of the data link, and network access protocols on vendor machines are in preliminary stages and will be included in an MCN testbed. These machines will include supercomputers, workstations, and frame buffers. All of this equipment will access the MCN with HSCs. The following is a chronology of events leading up to the current stage of this project.

September 1985	Network Modernization Project
April 1986	Ultra-High-Speed Graphics Project
July 1986	Initial HSC Standard Proposed by LANL
July 1986	CP* Concept Proposed
January 1987	MCN Concept Developed
December 1987	LANL/DEC CP* Collaboration
January-July 1988	Service and Protocol Specification
January 1988	Crossbar Interface (CBI) Design Initiated
May 1988	Initial HSC Data Link Proposed by LANL
July 1988	CBS Project Initiated
November 1988	Fiber HSC Standard Initiated
January 1989	CBS Assembly
February 1989	HSC Interface for CBI
February 1989	CBI Delivery
May 1989	HSC Public Review
March-May 1989	Base Level 0 CBI-CBS Testing
May-June 1989	Base Level 1 Data Link Testing
June-September 1989	Base Level 2-5 Intranet, Network Access Testing

Future Efforts

We are now working with a prototype CP* that will soon be connected to three supercomputers, a frame buffer, and two workstations (Figure 9). All of these systems are from different vendors and with the HSC they have some hope of utilizing the MCN testbed. Efforts in the future will include connecting a full set of hosts to the MCN and developing a full range of tests and applications to verify and improve the protocols and services at all levels. In particular, the network access connections in both hardware and software need to be investigated to begin reducing data copy, allocate network address space to users, and verify security and access control. We are also interested in experimenting with cut-through or desperation routing as well as developing some form of capability based routing for partitioning and need-to-know computing. The MCN may also act as a good testbed to investigate the characteristics of self-organizing phenomena, emergent computation and cellular automata in applications like congestion avoidance [24] and network management. [36] We will be monitoring progress in such technologies as fiber optic channels, optical or photonic crossbars [27] and parallel computer architectures. The MCN will be explored for the feasibility and flexibility of being used as either a network-centered [8] or processor-centered transport system for a loosely-coupled, coarse-grained system of processors. We will eventually be able to view networks of this type as a collective computer with hosts effectively used as peripherals to an overall system controlled by a distributed operating system.[4,32,33,34] In addition, it will be advantageous to think of the host front-end to the MCN as a peer or coprocessor to the host itself. In this way, the host front-end processor handles all distributed operating system, session, and communication tasks, while the other processor handles the computation or specific task it was designed to do. Programs and data would be in shared memory or executed in a batch mode for all jobs needing a particular computational resource. Finally, Los Alamos has an extensive network of local area networks at different sites. A need exists for an extensive high-speed backbone for geographically distributed local area networks across approximately a 45 square mile area at the laboratory (Figure 10). The MCN may fill that requirement.

Conclusion

We have described the need and goals for a high-speed local network which can both provide a framework and utilize parallelism for applications requiring high bandwidth, low latency and a flexible network topology. Progress has been made demonstrating the feasibility of a high-speed channel and crossbar technology to build a network of this type. The time appears right for significant effort and progress in integration of parallelism, networks and the applications that use them.

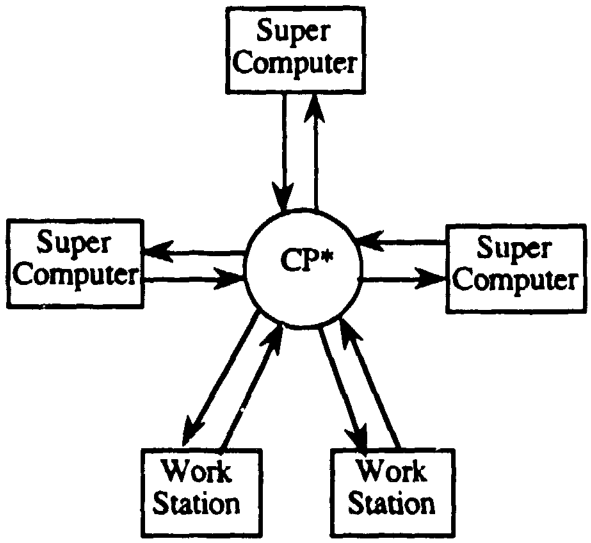

Figure 9. Multiple Crossbar Network Testbed

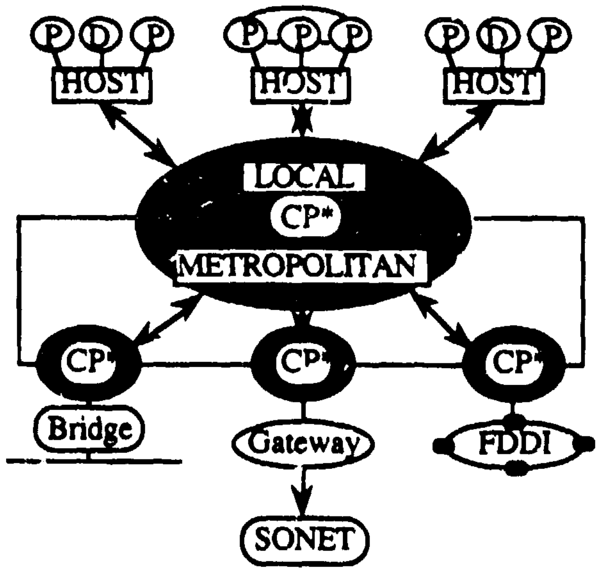

Figure 10. Potential Network Hierarchy

Acknowledgments

We would like to acknowledge the efforts of Michael McGowen for the basic concepts for the HSC, the CP*, and for the concept and design of the CBS. The efforts of the Network Modernization Project and Ultra-High-Speed Graphics Project also played a role in early motivation of this project. Valuable contributions were also made by Don Tolmie, Gene Dornhoff, Steve Tenbrink, John Morrison, Dave Dubois, Allan Meddles and Eric Vandevere. We are also indebted to Norm Morse for supporting this project and Karl-Heinz Winkler for demonstrating the need. We are grateful to the members of DEC's Southwest Engineering group in Albuquerque who contributed greatly to this effort. They include. Bruce Ellsworth, Paul Brooks, Rich Lewis, Bill Hedberg, Marty Halvorson, Mike Martinez, Joel Kaufmann, Win Quigley and contractors Gary Mendelsohn and Mary Monson. Finally, credit goes to all companies and attendees of the HSC Working Group for their work in developing the HSC standard. This has truly turned into an industry-wide effort.

References

[1] High-Speed Channel (HSC) Mechanical, Electrical, and Signalling Protocol Requirements, ANSI X3T9.3/88-023, Rev. 6.4, BSR X3./83 198x.

[2] T. Y. Feng, "A Survey of Interconnection Networks," *Computer* 14, 12, December 1981, pp. 12-27.

[3] G. J. Lipovki and M. Malek, *A Theory for Multicomputer Interconnection Networks*, Tech. Rep. TRAC-40, University of Texas at Austin, March 1981.

[4] Richard F. Rashid, Avadis Tevanian, Jr., Michael W. Young, et al., "Machine-Independent Virtual Memory Management for Paged Uniprocessor and Multiprocessor Architectures," *IEEE Trans. Comp.*, C-37(8):896-908, August 1988.

[5] H. Ferman Kelso, *Overview of LASL Integrated Computer Network* Los Alamos Scientific Laboratory report LA-6756-MS (January 1977).

[6] D. E. Tolmie, A. G. Dornhoff, and S. C. Tenbrink, *Interconnecting Computers with a High-Speed Parallel Interface*, Los Alamos National Laboratory report LA-9503-MS (August 1982).

[7] Emmanuel A. Arnould, Francois J. Bitz, Eric C. Cooper, et al., *The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers*, Third Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS III), April 1989.

[8] Mark A. Franklin and Sanjay Dhar, "Interconnection Networks: Physical Design and Performance Analysis," *J. Parallel and Distributed Comp.* 3 (1986), pp. 352-372.

[9] Howard Jay Siegel and Robert J. McMillen, "The Multistage Cube: A Versatile Interconnection Network," *IEEE Comp.* 14, 12 (December 1981), pp. 65-76.

[10] William C. Athas and Charles L. Seitz, "Multicomputers: Message-Passing Concurrent Computers" *Computer*, 21(8):9-24, August 1988.

[11] Daniel A. Reed and Dirk C. Grunwald, "The Performance of Multicomputer Interconnection Networks" *Computer*,20(6):63-73, June 1987.

[12] David Notkin, Andrew P. Black, Edward D. Lazowska, Henry M. Levy, Jan Sanislo and John Zahorjan, "Interconnecting Heterogeneous Computer Systems" *Communications of the ACM*, 31(3):258-273, March 1988.

[13] Howard Jay Siegel, "Interconnection Networks for Large-Scale Parallel Processing" *Lexington Books*, 1985

[14] Marshall C. Pease, "The Indirect Binary n-Cube Microprocessor Array" *IEEE Transactions on Computers*, 26(5):458-473, May 1977

[15] William F. Hedberg, "Multiple Crossbar Network: The Crossbar Interface", *Proceedings of the 14th Local Computer Network Conference*, Minneapolis, MN. October 1989

[16] Laxmi N. Bhuyan, Qing Yang and Dharma P. Agrawal, "Performance of Multiprocessor Interconnection Networks" *Computer*, 22(2):25-37, February 1989

[17] Randy Hoebelheinrich and Richard Thomsen, "Multiple Crossbar Network: Integrated Supercomputer Framework", *LA-UR 89-1627*, Los Alamos National Laboratory

[18] Jonathan S. Turner, "Design of a Broadcast Packet Switching Network", *IEEE Transactions on Communications*, 36(6):734-743, June 1988

[19] Charles L. Seitz, "The Cosmic Cube", *Communications of the ACM*, 28(1):22-33, January 1985

[20] William J. Dally and Charles L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks", *IEEE Transactions on Computers*, 36(5):547-553, May 1987

[21] H. Kanakia and D. Cheriton, "The VMP Network Adaptor Board (NAB): High-Performance Network Communication for Multiprocessors", *Computer Communications Review*, 18(4), August 1988

[22] Gigabit Working Group, B. Leiner, Editor, "Critical Issues in High Bandwidth Networking", *RFC 1077*, Stanford University, November 1988

[23] K.K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer", *Computer Communications Review*, 18(4), August 1988

[24] Van Jacobson, "Congestion Avoidance and Control", *Computer Communications Review*, 18(4), August 1988

[25] Howard Jay Siegel, "Analysis Techniques for SIMD Machine Interconnection Networks and the Effects of Processor Address Masks", *IEEE Transactions on Computers*, 26(2):153-161, February 1977

[26] George B. Adams III, Dharma P. Agrawal and Howard Jay Siegel, "Fault-Tolerant Multistage Interconnection Networks", *Computer*, 20(6):14-27, June 1987

[27] A.A. Sawchuk, B.K. Jenkins, C.S. Raghavendra and A. Varma, "Optical Crossbar Networks", *Computer*, 20(6):50-60, June 1987

[28] Eli Opper and Miroslaw Malek, "On Resource Allocation in Multistage Interconnection Network-Based Systems", *Journal of Parallel and Distributed Computing* 1, 206-220 (1984)

[29] O. Kramer and H. Muhlenbein, "Mapping strategies in message-based multiprocessor systems", *Parallel Computing* 9 (1988/89) 213-225

[30] Yoshua Bengio, Regis Cardin, Renato De Mori and Ettore Merlo, "Programmable Execution of Multi-Layered Networks for Automatic Speech Recognition", *Communications of the ACM*, 32(2):195-199

[31] Jayadev Misra, "Distributed Discrete Event Simulation", *Computing Surveys*, 18(1), 39-65, March 1986

[32] Leonard Kleinrock, "Distributed Systems", *Communications of the ACM*, 28(11):1200-1213, November 1985

[33] Richard W. Watson and John G. Fletcher, "An Architecture for Support of Network Operating System Services", *Computer Networks*, 4(1980) 33-49

[34] Richard W. Watson, "The Architecture of Future Operating Systems" *UCRL-Preprint Lawrence Livermore National Laboratory* ,Cray Users Group Meeting, Tokyo, September 1988

[35] Willy Zwaenepoel, "Protocols for Large Data Transfers over Local Networks", *Technical Report TR-85-23, Department of Computer Science, Rice University*

[36] Roy Maxion, "Emergent Behavior of a Fault Management System in a Network Community", *Proceedings of the Ninth Annual International Conference on Emergent Computation*, Los Alamos, May 1989