[36] Clark, D. D., Tennenhouse, D. L.
Architectural Considerations for a New Generation of Protocols
Proceedings of Sigcomm–90, pp. 200–208.

[37] Multicomputers
Chapter 5 in *Developments in Concurreny and Communication*
Ed. Hoare, C. A. R.
Addison–Wesley Publishing Co., 1990.

[38] SCI – Scalable Coherent Interface
Draft Report P1596: Section 1/D0.85. IEEE.

[39] Cheriton, D. R.
Sirpent$_{TM}$: A High–Performance Internetworking Approach
Proceedings of Sigcomm–89, pp. 158–169.

[40] Kermani, P., Kleinrock, L.
Virtual Cut–Through: A New Computer Communication Switching Technique. *Computer Networks*, 3(4), pp. 267–286, September 1979.

[41] Dally, W. J., Seitz, C. L.
Deadlock–Free Message Routing in Multiprocessor Interconnection Networks
IEEE Transactions on Computers, Vol. C–36, No. 5, May 1987.

[42] Flaig, C. M.
VLSI Mesh Routing Systems
California Institute of Technology, Computer Science Department 5241:TR:87, May 1987.

[13]

[14] Seitz, C. L., Mead, C., Conway, L.
Introduction to VLSI Systems
Addison-Wesley, 1980.

[15] Ngai, John. Y.
A Framework for Adaptive Routing in Multicomputer Networks
California Institute of Technology, Computer Science Department Caltech–CS–TR–89–09.

[16] Arnould, E., Bitz, F., Cooper, E., Kung, H. T., Sansom, R., Steenkiste, P.
The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers
Proceedings of the Third International Conference on Architectural Support for Programming Languages and Operating Systems, ACM 1989, pp. 205–216.

[17] Tobagi, F. A.
Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks
Proceedings of the IEEE, Vol. 78, No. 1, January 1990, pp. 133–166.

[18] Fibre Channel: Physical and Signalling Interface (FC–PH) Rev. 2.2 Working draft proposed American National Standard for Information Systems, January 24, 1992.

[19] Schroeder, M. D., Birrel, A. D., et al.
Autonet: a High–speed, Self–configuring Local Area Network Using Point–to–point Links
IEEE Journal on Selected Areas in Communication, Vol. 9, No. 8., October 1991, pp. 1318–1335.

[20] Davie, Bruce S.
A Host–Network Interface Architecture for ATM
Proceedings of SIGCOMM '91, pp.307–315.

[21] Leslie, I., McAuley, D.
Fairisle: An ATM Network for the Local Area
Proceedings of SIGCOMM '91, pp.327–336.

[22] Davie, B. S.
An ATM Network Interface for High–Speed Experimentation
IEEE Workshop on the Architecture and Implementation of High–Performance Communication Subsystems HPCS '92.

[23] Cooper, E., Menzilcioglu, O., Sansom, R., Bitz, F.
Host Interface Design for ATM LANs.
IEEE Proceedings of 16th Annual Conference on Local Computer Networks. Oct. 1991, pp. 247–258.

[24] Schroeder, M. D.
Presentation given at the *IEEE Workshop on the Architecture and Implementation of High–Performance Communication Subsystems* HPCS '92.

[25] Clark, D. D., Tennenhouse, D. L.
Research Program on Distributed Video Systems
Personal communication.

[26] Van Der Jagt, L.
Wiring Media
Journal of Data & Computer Communications
Summer 1991, pp. 14–21.

[27] Gusella, R.
A Measurement Study of Diskless Workstation Traffic on an Ethernet
IEEE Transactions on Communications
Vol. 38, No. 9, September 1990, pp. 1557–1568.

[28] Falaki, S. O., Sorenson, S–A.
Traffic Measurements on a Local Area Computer Network
Computer Communications
Vol. 15, No. 3, April 1992, pp. 192–197.

[29] Jain, N., Schwartz, M., Bashkow, T. R.
Transport Protocol Processing at GBPS Rates
Proceedings of Sigcomm–90, pp. 188–199.

[30] Cohen D., Finn, G., Felderman, R., DeSchon, A.
ATOMIC: A Very–High–Speed Local Area Network. *Proceedings of IEEE HPCS '92: Workshop on the Architecture and Implementation of High Performance Communication Subsystems*.

[31] Partridge, C.
How Slow is One Gigabit Per Second
Computer Communication Review, Vol. 20, No. 1, January 1990.

[32] Jain, N., Schwartz, M., Bashkow, T. R.
Transport Protocol Processing at GBPS Rates
Proceedings of Sigcomm–90, pp. 188–199.

[33] Kanakia, H., Cheriton, D.R.
The VMP Network Adapter Board (NAB): High–-Performance Network Communication for Multiprocessors. *Proceedings of Sigcomm–88*, pp. 175–187.

[34] Zitterbart, M.
Parallel Protocol Implementations on Transputers. Experiences with OSI TP4, OSI CLNP and XTP.
Proceedings of the IEEE HPCS '92: Workshop on the Architecture and Implementation of High Performance Communication Subsystems.

[35] Cooper, E. C., Steenkiste, P. A., Sansom, R.D. and Zill, B.D.
Protocol Implementation on the Nectar Communication Processor
SIGCOMM 90, ACM 1990, pp. 135–144.

sparse topologies. Those extensions make it practical to use multicomputer networking technology to create gigabit LANs.

The small size of these network interfaces, coupled with their performance, allows a workstation architect to replace the system bus with a network composed of point-to-point gigabit channels. This produces an architecture in which major devices reside on a multi-gigabit network fabric, each directly accessible to the LAN.

By building the architecture around at network rather than a bus, communication capacity scales uniformly with growth, which allows indefinite expansion. A greater reliance is placed on the use of network routing hardware, rather than software demultiplexing, to direct packets and their data to peripherals, which enhances performance. In addition, the architecture inherently supports parallel or clustered computing due to its reliance on the multicomputing model of communication.

Since peripheral devices in this architecture interact with one another by exchanging packets, their model of operation is abstractly presented by their network protocols and those protocol's behavior. By presenting to the world a virtual rather than physical model of operation, physical interface details can remain largely hidden. This extends to those devices networking's great advantage of interoperability.

7.0 References

[1] Athas, W. C., Seitz, C. L.
Multicomputers: Message–Passing Concurrent Computers
IEEE Computer, pp. 9–24, August 1988.

[2] Seitz, C. L.
The Cosmic Cube
Communications of the ACM, pp.22-33, January 1985

[3] Seitz, C. L.
Concurrent Architectures
Chapter 1 in *VLSI and Parallel Computation*
Ed. Suaya, R., Birtwistle, G.
Morgan and Kaufmann, 1990.

[4] *Submicron Systems Architecture Project Semiannual Technical Report*
California Institute of Technology, Computer Science Department Caltech–CS–TR–90–05.

[5] *Submicron Systems Architecture Project Semiannual Technical Report*
California Institute of Technology, Computer Science Department Caltech–CS–TR–91–03.

[6] Cohen, D., Finn, G. G., Felderman, R., DeSchon, A.
The Use of Message-Based Multicomputer Components to Construct Gigabit Networks
ACM Computer Communication Review
Vol. 23, No.3., July 1993.

[7] Frank, E. H., Lyle, J.
SBus Specification B.0
Sun Microsystems,1990.

[8] TURBOchannel Hardware Specification
EK-369AA-OD-007A
Digital Equipment Corporation, Sept. 1991.

[9] VL-Bus 1.0 Standard
Video Electronic Standards Association, Aug. 1992.

[10] Finn, G.
An Integration of Network Communication with Workstation Architecture
ACM Computer Communication Review, Vol. 21, No. 5, October 1991.

[11] Hayter, M., McAuley, D.
The Desk Area Network
ACM *Transactions on Operating Systems*, October 1991, pp. 14–21.

[12] Pennebaker, W. B., Mitchell, J. L.
JPEG - Still Image Data Compression Standard
Van Nostrand Reinhold., 1993.

ment and debugging, and it provides flexibility to determine where it is appropriate to physically locate a layer's functions.

5.0 Future Expectations

VLSI feature size and operating voltage will continue to decrease for some time to come. As a result, the area on a chip needed for a network interface will shrink while its bandwidth increases to beyond a Gb/s. If the past is an indicator, greater functionality will continue to be designed into future microprocessors, DSPs and ASICs.

5.1 Network Interface Inside the Microprocessor

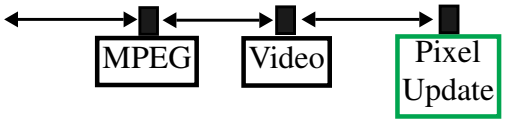
Mosaic-C is an example of a single-chip network host. It includes a network interface plus a DMA port to memory. In 1.2 micron CMOS the interface consumed approximately 9 mm^2. This suggests an evolutionary path that places a network interface inside microprocessor chips, tightly coupling the network to the cache/memory subsystem.

Whether or not manufacturers decide to do that remains to be seen. However, the increasing importance of networking, link speeds that approach memory bandwidth, and the emergence of high-demand network applications, suggest that this be given serious thought.

5.2 Network Pipeline

If it becomes practical to incorporate the network interface into a microprocessor, it will be equally practical to incorporate it into selected DSPs, ASICs and MCMs. Doing so provides the architect with additional freedom.

In an NVD, packets that contain MPEG data would normally be addressed to an MPEG module. The MPEG module's output would normally be directed to the Video session module, whose output would be addressed to the Pixel-Update module. This arrangement resembles a processing pipeline.



The network address here assumes the function of a memory-mapped device address in conventional architecture. However, unlike conventional architecture, each module can be located in principal anywhere on the network.

Network channels are sharable and can multiplex many different streams of traffic. By providing a separate address for each module, packets not addressed to a 'pipeline' stage bypass it. Pixel-Update packets generated elsewhere on the network can be addressed directly to a Pixel-Update module.

5.3 Interface Standardization

The layered modularity of an NVD can be directly reflected in its hardware organization. In principal, each layer is logically isolated from another by the network. The only necessary external interface to any intermediate NVD layer is the network.

As a result, the hardware module that implements an intermediate NVD layer can be designed to interact only via its physical network channels. Its only necessary external contacts are power, reset, and its network channels.

If the same network physical layer between processing modules is used throughout, it is realistic to adopt a standard pin-out for such modules, making it possible to add, remove, or update them as needed on an NVD motherboard. The performance of multicomputer-derived network channels, their small signal count and sub-chip sized interface, suggests an avenue by which to develop a standardized physical data interface.

6.0 Summary

The development of massively parallel multicomputers coincided with the development of smaller and faster network technology. Multicomputer network interfaces are now sub-chip in size and operate at a rate of 500 Mb/s or more per channel.

Although designed for communication over short distances and in networks with a regular topology, it has been possible to modify multicomputer networking hardware to both greatly extend that distance and to allow irregular and

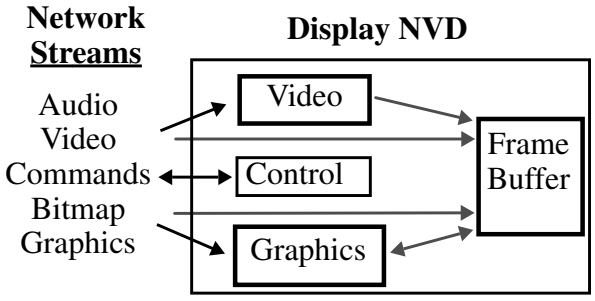
formats that are sufficient to capture display function. This suite must remain largely invariant across the NVD type, which could include CRT, addressable-matrix, micro-tip panel, or micro-mirror display devices, among others.

What we expect from displays is changing. Audio/video and still-frame capabilities are beginning to augment bitmap graphics. As a result, display-driver design is rapidly changing. Audio/video protocols must be a part of a practical display NVD definition.

The computer network audio/video arena is unsettled. For NTSC-quality video one could use as the standard format packets that encapsulate CCIR 601. However, the shifts away from interlaced to progressively scanned displays, and towards JPEG [12], MPEG or MPEG-II compressed transmission, argue that these digital formats are better candidates for the transmission of video and audio/video data to displays.

4.1 Identifying Common Display Protocols

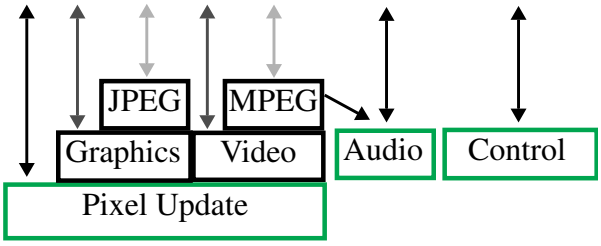
Consider collaborative conferencing. Streams of data and commands from participating sites arrive at the display. Each type of stream requires it own special processing. A graphics stream may require rendering. A video stream may require decompression.



Ultimately, display operations result in changes either to a display's control state or to its frame buffer.[2] The two protocols fundamental to display operation will be those for device control and frame-buffer pixel update.

[2] The term *frame buffer* here means the stored image. An addressable-matrix display may not have a frame buffer in the same sense that a CRT does.

In a display that contains internal support for session audio, video and still-frame, compressed MPEG or JPEG streams would feed into the audio/video and still-frame session protocols. The output of the session protocols would feed into the pixel-update protocol. This defines a display NVD protocol stack, similar to the FTP/TCP/IP Internet stack.



Display NVD Protocol Stack

4.1.1 Pixel Update Layer

A degree of standardization is already emerging for pixel update. Examples are display PostScript and Sun Microsystem's Pixrect interfaces. Display Postscript assumes the existence of a rendering engine associated with the frame buffer. The commands passed across the interface are stated in terms of PostScript primitives.

The Pixrect interface is more suitable as a starting point for defining a lowest-layer protocol. It provides a procedural interface to various types of frame buffers to perform rasterop and draw vectors. When translated into remote procedure calls, Pixrect is sufficient for the operation of window software.

4.2 Layer Modularity

The objective of an NVD definition is to define all interactions in terms of packet interactions. Properly designed, each stack layer will have well-defined interactions with the layers immediately above and below it. In principal, only the lowest layer need be contained within the NVD itself. For a display NVD this implies that the session and encoded-data layers could reside elsewhere.

Since the physical interface to an NVD is that of a network, the higher protocol layers could reside within hosts elsewhere on the network. This provides flexibility in software develop-

fers, the data must first be copied from the source device buffers into main memory, and subsequently, re-copied into the destination device buffers. This requires two otherwise unnecessary accesses to memory.

2.4 Direct Local Network Access

A Netstation may be thought of as having its own private network, but it should be constructed using the same networking components that are used in the LAN. This eliminates the need for a gateway between it and the LAN, both lowering latency and removing a probable performance bottleneck.

That allows each Netstation node to directly send or receive packets over the LAN or the Internet. In particular, video streams can be sent or received directly by camera or display nodes. This has obvious performance advantages for applications such as multimedia conferencing and remote visualization.

2.5 Configuration Flexibility

Workstation architectures have sharply defined limits to their expansion. Electrical constraints limit high-speed buses to a few expansion 'slots'. However, data communication based upon multicomputing technology supports indefinite expansion. Netstation nodes are logically independent and can be housed separately if desired. The only physical connection necessary is to the network.

2.6 Cluster Computing Support

By analogy to a multicomputer, a cluster of workstations on a LAN has the potential to cooperate to solve a compute-intensive task. In theory, by scavenging the largely unused computation capacity of idle workstations, one can achieve the performance of an expensive supercomputer at a fraction of the cost.

But that objective is impractical using the slow transmission speeds on typical LANs. Injecting a 50 byte packet into an Ethernet requires approximately 50 microseconds. By contrast, it takes one microsecond on a Mosaic channel, across which data transfers at memory speed.

By utilizing as a base a communication technology designed for multicomputers, the difference between a cluster of Netstations and a multicomputer is minimized. This makes cluster computing physically practical.

3.0 Network Virtual Device

Netstation nodes interact solely through the network. A node must publicize its model of operation before other nodes can use it. This requires defining node-specific protocols.

Nodes of similar function group somewhat naturally into types. Display nodes form a type. Although displays use differing hardware technologies and possess varying levels of performance, their function is similar. Cameras, disks and processor/memory modules are other examples of node types.

Functional similarity within a node type allows definition of a suite of common data and command protocols that allow remote access to those functions. A type of node in conjunction with its protocols forms a Network Virtual Device (NVD). Device-control functions, that occur in today's peripherals via memory mapped command registers or special I/O bus lines, here occur as a result of packets received over a network command port.

Within a type of NVD, specific hardware implementations may differ radically, having different internal designs and driver software. By presenting a virtual rather than physical model of operation, physical interface details can remain largely hidden.

When it is necessary to know the configuration details of a specific NVD, access to them can be provided by the command protocols. For example, window systems need to know the bitmap image size supported by a display. This would be accomplished by sending an enquiry packet over that NVD's command port, which would return a reply.

To evaluate, test and develop these ideas, the creation of both Display and Camera NVD types is being undertaken at USC/ISI. The development of a Display NVD is now outlined.

4.0 Display NVD

A successful NVD specification for displays requires defining a suite of protocols and data

vides the flexibility to create a topology that ensures redundancy, and allows routing software to provide alternate or reserved routes to meet quality-of-service constraints.

2.0 Networked Architecture

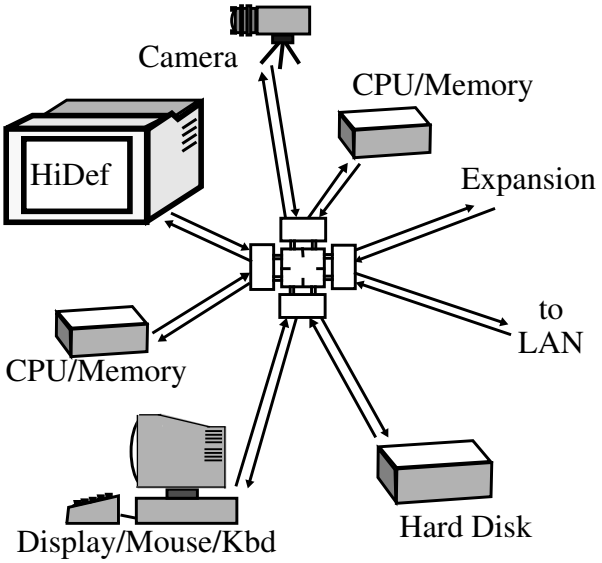
Most workstation system buses have bandwidths of from 1/2 to 1 Gb/s [7][8][9]. The channel bandwidths of the Ethernet at 10 Mb/s and the emerging 100 Mb/s LANs are a small fraction of workstation system-bus and memory bandwidth. However, as network channel bandwidth reaches a Gb/s, those buses become effectively dedicated to a network channel when that channel is active. To redress that imbalance would require system buses of bandwidth of from 5 -to- 10 Gb/s.

The small size and low cost of an ATOMIC host interface suggests another solution, to replace the system bus with a network fabric. Instead of connecting peripheral devices to a system bus, connect them to the network, making them host nodes on the network. *Netstation* architecture replaces the system bus with multiple gigabit data paths, transforming the workstation into a heterogeneous multicomputer [10].[1]

2.1 Architectural Advantages

Transforming the workstation into a multicomputer is attractive for other reasons than the bus-sharing problem. It provides features not generally available to workstations:

1. Communication capacity scales uniformly with growth.
2. Peripheral nodes can route data directly to one another without first passing data through the workstation's main processor/memory.
3. Network routing hardware, rather than software demultiplexing, can direct packets and their data to peripherals.
4. Indefinite expansion is allowed.
5. Support for parallel or clustered workstation computing is inherent.
6. Nodes can exchange data directly with one another *outside* the workstation without first passing it through their workstation's main processor/memory.



2.2 Scalable

Buses are broadcast devices that permit only one transmitter at any one time. They have a fixed bandwidth ceiling. Once that limitation is reached, achieving greater performance requires a different bus.

By way of contrast, a network fabric composed of two-way point-to-point channels supports simultaneous two-way traffic. Netstation architecture inherits the scalable, parallel communications characteristics of a multicomputer. Every channel added to the network increases total system bandwidth.

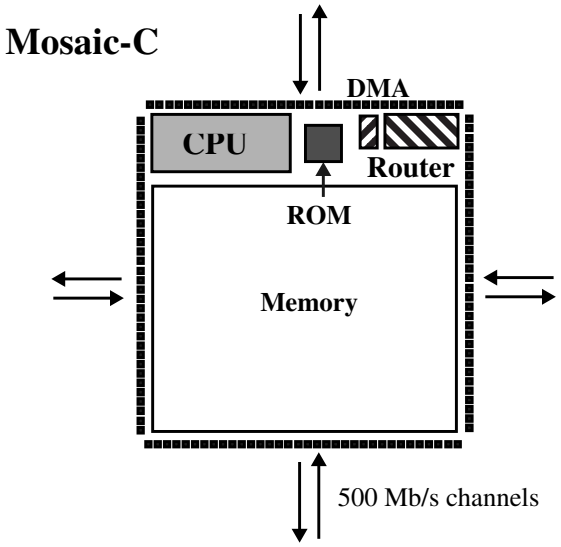
2.3Direct Inter-device Communication

Each node attached to a Netstation's communication fabric may directly send and receive packets. A disk may transmit directly to a display, a camera directly to a disk, and so on. Packets are routed by hardware to the intended destination device.

This is fundamentally more efficient than the procedure that occurs in a workstation where data transfers require the execution of protocol-stack code by the main processor. In addition, unless the system bus and operating system both support peer-to-peer data trans-

[1] An ATM network fabric for workstations is also being developed [11]. However, ATM requires additional fragmentation and reassembly.

The Caltech research effort culminated in the creation of a single-chip multicomputer node, Mosaic-C [4][5]. The Mosaic point-to-point router and DMA interface is capable of sourcing or sinking messages at a rate equivalent to the memory bandwidth of the node, which is approximately 500 Mb/s. The approximate floor plan of Mosaic-C illustrated below shows that the network interface fits onto a corner of the chip.



1.2 Transforming a Multicomputer Network into a LAN

TCP/IP assumes a datagram model of communication. Carrying datagrams at gigabit rates is precisely what Mosaic channels are designed to do. This suggested that adapting Mosaic technology to create a gigabit LAN would be a profitable experiment. A prototype LAN, called ATOMIC, was built from Mosaic components at USC/ISI in late 1991 [6].

A typical ATOMIC host interface contains two Mosaic chips, one devoted to incoming traffic and the other to outgoing traffic. Each shares memory with the workstation microprocessor and performs buffer management and network controlling tasks.

A mesh of Mosaic-C chips is programmed to act as an ATOMIC interhost router or crossbar. Prototype host interfaces sustain a 900 Mb/s transfer rate and are connected via their Mosaic channels to the crossbar or to other host interfaces. ATOMIC is actually a physically-distributed multicomputer that is *programmed* to behave like a LAN.

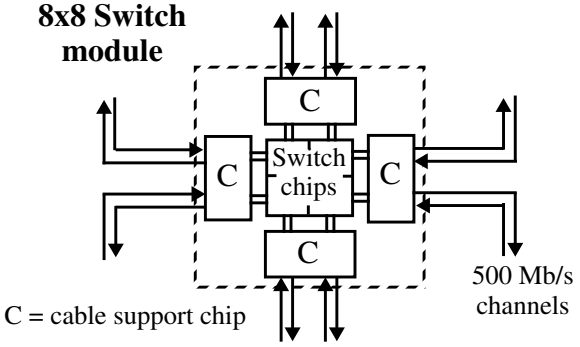
1.3 Adaptation to a LAN

Experience with ATOMIC has shown that although Mosaic technology is suited to the creation of a LAN, it is by no means ideal. The difficulties are principally: (1) the topology of a LAN is irregular and sparse by comparison with that of a multicomputer, (2) adjacent hosts on a LAN are often more than 30 meters apart, while the distances between adjacent multicomputer nodes are on the order of a meter or less, and (3) a network of Mosaic links is not robust in the presence of component or cable failure.

Mechanisms were developed to overcome those difficulties by Professor Seitz's group at Caltech and USC/ISI. They are being incorporated into a new generation of Mosaic-based chips being produced for use in LANs.

The newly developed chips are:

1. The host-interface chip, which contains a two-channel network port, a CPU, and local-bus support with 1 Gb/s of memory bandwidth.
2. An multiport switch chip.
3. A cable-support chip that provides channel robustness and also supports up to 40 meter point-to-point transmission.



The switch chip functions similarly to a crossbar, but with improved fairness. When configured as an interhost router, each port is associated with a cable-support chip to create a basic 8x8 interhost switch module.

Ports on a switch may be connected to host-interface chip ports or to a port on another switch. This allows indefinite expansion, pro-

Netstation Architecture
Multi-Gigabit Workstation Network Fabric

Gregory G. Finn, Paul Mockapetris
USC/Information Sciences Institute

Abstract

Advances in multicomputer research have produced a sub-chip sized gigabit network interface. This fundamentally changes the scope of problems to which networking can be applied. It is now practical to re-architect a workstation around a multi-gigabit network fabric, eliminating the system bus by attaching major peripheral subsystems to the network. This removes the restrictions on slot count and channel length associated with high-speed buses, while it produces an architecture with more attractive scaling characteristics.

Processors and peripherals interface to one another across the network logically, via protocols. The complex details of the physical-device interfaces remain hidden. This greatly enhances interoperability. Network access lets data flow across the network directly between processors, displays, cameras, disks, and so on. Cooperative computing, cluster processing, and conferencing applications are more naturally supported by this heterogeneous multicomputer architecture.

1.0 Multicomputers & Networking

Multicomputers are a class of parallel computers that rely upon message-passing rather than the use of distributed, shared memory [1]. Each node of a multicomputer consists of a processor with an isolated and independent memory. Nodes communicate with one another exclusively by exchanging messages across a network.

That is similar to the interactions between workstations on a LAN. But although similar, until recently, local-area networking and multicomputer networking technologies developed in different directions.

1.1 Background

First-generation multicomputers used microprocessor motherboards for their nodes [2], and in some cases their message-passing networks were implemented with Ethernet chips. It was quickly determined that multicomputers would greatly benefit from much better network performance than that offered by LAN chips available in the early 1980's. Research at Caltech began to focus on achieving a 100-fold improvement in message-passing latency for short messages [3].

Multicomputer performance improved on three fronts. Smaller and faster processing nodes, and faster interconnection networks were built. As the node size shrank, cost per node was reduced, and thus for a constant expenditure the degree of parallelism that could be applied to a problem grew. As interconnection network performance rose, the time required to transmit a message from one node to another fell. This decreased the idle time spent by nodes that were waiting to receive a message, and so improved both efficiency and speed of execution.

This research was sponsored by theAdvanced Research Projects Agency under Contract No. DABT63-93-C-0062. Views and conclusions contained in this report are the authors' and should not be interpreted as representing the official opinion or policies, either expressed or implied, of ARPA, the U.S. Government, or any person or agency connected with them.

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90202

Internet: Finn@isi.edu