# Myrinet -- A Gigabit-per-Second Local-Area Network

(Based on a keynote talk presented by Charles L. Seitz)

Nanette J. Boden, Danny Cohen, Robert E. Felderman,
Alan E. Kulawik, Charles L. Seitz, Jakov N. Seizovic, and Wen-King Su

Myricom, Inc.
325 N. Santa Anita Ave.
Arcadia, CA 91006
(http://www.myri.com)

**Abstract.** Myrinet is a new type of local-area network (LAN) based on the technology used for packet communication and switching within "massively-parallel processors" (MPPs). Think of Myrinet as an MPP message-passing network that can span campus dimensions, rather than as a wide-area telecommunications network that is operating in close quarters. The technical steps toward making Myrinet a reality included the development of (1) robust, 25m communication channels with flow control, packet framing, and error control; (2) self-initializing, low-latency, cut-through switches; (3) host interfaces that can map the network, select routes, and translate from network addresses to routes, as well as handle packet traffic; and (4) streamlined host software that allows direct communication between user processes and the network.

**Background.**

In order to understand how Myrinet differs from conventional LANs such as Ethernet and FDDI, it is helpful to start with Myrinet's genealogy. Myrinet is rooted in the results of two ARPA-sponsored research projects, the Caltech Mosaic, an experimental, fine-grain multicomputer [1], and the USC Information Sciences Institute (USC/ISI) ATOMIC LAN [2, 3], which was built using Mosaic components. Myricom, Inc., is a startup company founded by members of these two research projects.

**Multicomputer Message-Passing Networks.** A multicomputer [4, 5] is an MPP architecture consisting of a collection of computing *nodes*, each with its own memory, connected by a *message-passing network*. The Caltech Mosaic was an experiment to "push the envelope" of multicomputer design and programming toward a system with up to tens of thousands of small, single-chip nodes rather than hundreds of circuit-board-size nodes. The fine-grain multicomputer places more extreme demands on the message-passing network due to the larger number of nodes and a greater interdependence between the computing processes on different nodes. The message-passing-network technology developed for the Mosaic [6] achieved its goals so well that it was used in several other MPP systems, including the medium-grain Intel Delta and Paragon multicomputers, the Stanford DASH multiprocessor, and the MIT Alewife multiprocessor.

In common with LANs, multicomputer message-passing networks send and receive data in the form of packets. Any node may send a packet to any other node. A packet consists of a sequence of bytes starting with a routing *header*, which is examined by routing circuits that steer the packet through the network. The header is followed by an arbitrary-length *payload*, which is the data delivered to the destination. The packet is terminated by a *trailer* (or *tail*), which may include a checksum. The maximum length of the packet, known in networking circles as the maximum-transmission unit (MTU), may be limited by considerations of fairness in the arbitration in the routing circuits and by buffer-size limits in the sending and receiving nodes. If a program sends a message larger than the MTU, the operating system fragments the message into a series of packets, and the receiving node reassembles the packets.

In contrast with LANs, the distinctive characteristics of MPP message-passing networks include:

High data rates. Channel data rates today are in the range from hundreds to thousands of Megabits/s (Mb/s). These individual channels are commonly organized in full-duplex pairs called *links*. A Myrinet link composed of a full-duplex pair of 640Mb/s channels is reasonably referred to as a 1.28Gb/s link in comparison with 10Mb/s "10-Base" or 100Mb/s "100-Base" Ethernet, in that Ethernet channels carry packets in only one direction at once.

Regular topologies and scalability. The network is constructed by interconnecting elementary routing circuits in a mathematically regular topology. After the early hypercubes, most multicomputers adopted low-dimension topologies, such as the two-dimensional mesh illustrated in Figure 1. Unlike LANs such as Ethernet and FDDI, in which all packet traffic shares a single physical medium, a network such as a two-dimensional mesh is said to be scalable. The aggregate capacity grows with the number of nodes because many packets may be in transit concurrently along different paths. The regularity of the network allows simple, algorithmic routing that avoids deadlocks that might otherwise occur due to cyclic dependencies in the routes. See [7] for a collection of papers on these technical issues.

Very low error rate. Inasmuch as a multicomputer message-passing network operates in an intra-computer environment, typically as an active backplane, the occurrence of bit errors or lost packets can be made extremely rare, and of undetected communication errors still more rare. This high reliability, in contrast with the usual assumption in LANs that communication is unreliable, has many implications. If a multicomputer's message-handling software had to assume that the physical communication were unreliable, it would need to employ more complex communication protocols to make end-to-end communication reliable. These protocols also require additional storage to keep temporary copies of packets sent. At the communication rates of MPPs, executing such protocols would add significant overhead for each packet.

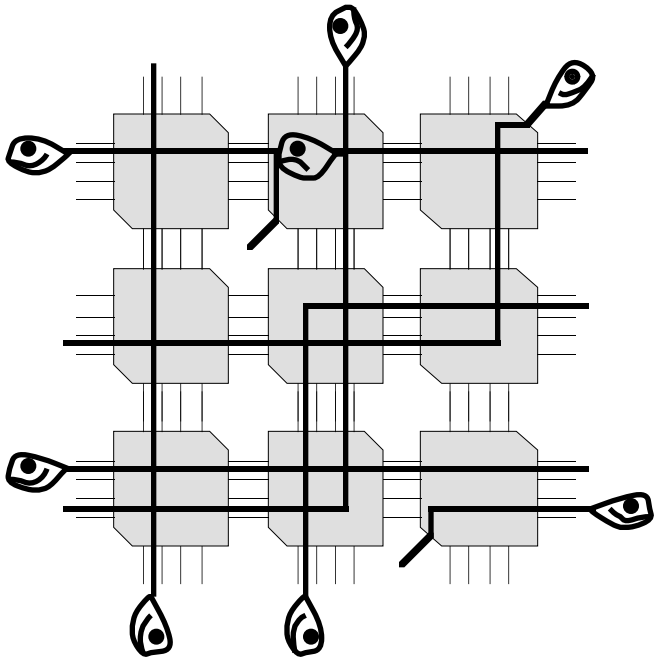

Figure 1: An illustration of packets flowing concurrently through part of a two-dimensional-mesh MPP network. The packets injected at the lower left and dejected at the upper right of routing circuits are those to and from the computing nodes. As soon as the header has been decoded at an input channel, the cut-through-routing circuits advance the packet into the required outgoing channel if it is not already in use; otherwise, the packet is blocked, as in the top-center node, until the outgoing channel becomes available. The *x*-then-*y* dimension-order routing illustrated eliminates cyclic dependencies in the routes, and hence avoids deadlock.

Cut-through routing. In an environment in which communication is reliable, the routing circuits can employ an aggressive form of routing known as cut-through routing. In conventional store-and-forward routing, the entire packet is buffered and its checksum verified in each intermediate node before the packet is sent on the required outgoing channel. In cut-through routing, the packet is advanced into the required outgoing channel as soon as the header is received and decoded. It is possible in either case that the required outgoing channel is already in use, in which case the packet must be held or blocked until the channel becomes available.

The use of flow control on every communication link. If the required outgoing channel is already in use, a store-and-forward packet must remain queued in a routing circuit or node, which is assumed to have a substantial amount of memory for packet buffering. The usual strategy in cut-through routing is to block a packet with flow control if the required outgoing channel is already in use. In this way, the cut-through-routing circuit requires no packet buffering, but each link must provide flow control. The

common mechanism for providing flow control is to acknowledge each flow-control unit, typically a byte.

**USC/ISI ATOMIC LAN.** MPP routing networks and LANs evolved from different requirements, assumptions, and techniques. There is, however, a need for higher speed LANs. In the ten years since UNIX workstations first appeared on the market, processor speeds have advanced by at least 100-fold, whereas most workstations are still shipped with the same common-denominator 10Mb/s Ethernet. New applications, such as the increasing use of distributed computing and the storage and communication of video, place additional demands on network performance, both bandwidth and latency.

In 1991, a research group at USC/ISI started a farsighted project to use MPP components to construct a high-speed LAN. Within a matter of weeks after receiving several Mosaic host interfaces and a software toolkit, the ATOMIC project demonstrated a small network performing standard TCP/IP communication at burst rates within the network of 400Mb/s. The Mosaic interfaces chained in one dimension, and packets could be addressed to any other node up or down the chain. The testbed grew to include the equivalent of crossbar switches based on two-dimensional-mesh Mosaic multicomputer arrays. In order to deal with the irregular topology of chains of varying length connected through switches, ATOMIC added automatic network mapping and the translation from network addresses to routes. Mapping and address-to-route translation were the key insights needed to adapt an MPP network to a LAN environment.

The ATOMIC testbed also demonstrated an experimental result of particular interest, the transfer of more than $10^{15}$ bits without a single bit error or dropped packet.

Although ATOMIC was an innovative research project, the ATOMIC testbed had practical limitations listed here so we may refer to them later.

1. The communication links employed asynchronous request/acknowledge signaling, which was designed for distances up to 1m. The links operated correctly over much longer distances, but at progressively slower data rates due to the flight time of the asynchronous signals. In addition, the absence of an acknowledge on a disconnected channel caused any packet directed to this channel to block, leading eventually to deadlock of the entire network.

2. The network topology, 1-D chains of interfaces connected by 2-D-mesh multicomputers used as switches, was complex for network mapping and did not allow for hosts within chains to be powered off. No good use was found for the substantial computing power and memory hidden in the ATOMIC switches.

3. ATOMIC's performance was limited by the lack of a DMA engine in the host interface; however, the ability of the interface to execute a complex control program was crucial for managing the network interface.

4. The end-to-end data rates that could be achieved were, not surprisingly, limited by the TCP/IP "protocol stack" in the host operating system.

**Myrinet Components and Operation**

The design of Myrinet was based directly on the ATOMIC experiences, good and bad. There was, however, no constraint for Myrinet to use an existing MPP network. Instead, our group developed communication schemes, routing techniques, custom-VLSI chips, and network-control software specifically for the LAN environment, while still drawing heavily on MPP technology. The specifications that govern the operation of a Myrinet LAN are published and open [8].

Figure 2 is an example meant to provide context for the detailed descriptions that follow. A Myrinet LAN is composed of point-to-point, full-duplex links that connect hosts and switches. The multiple-port switches may be connected by links to other switches and to the single-port host interfaces in any topology, including those with cycles.

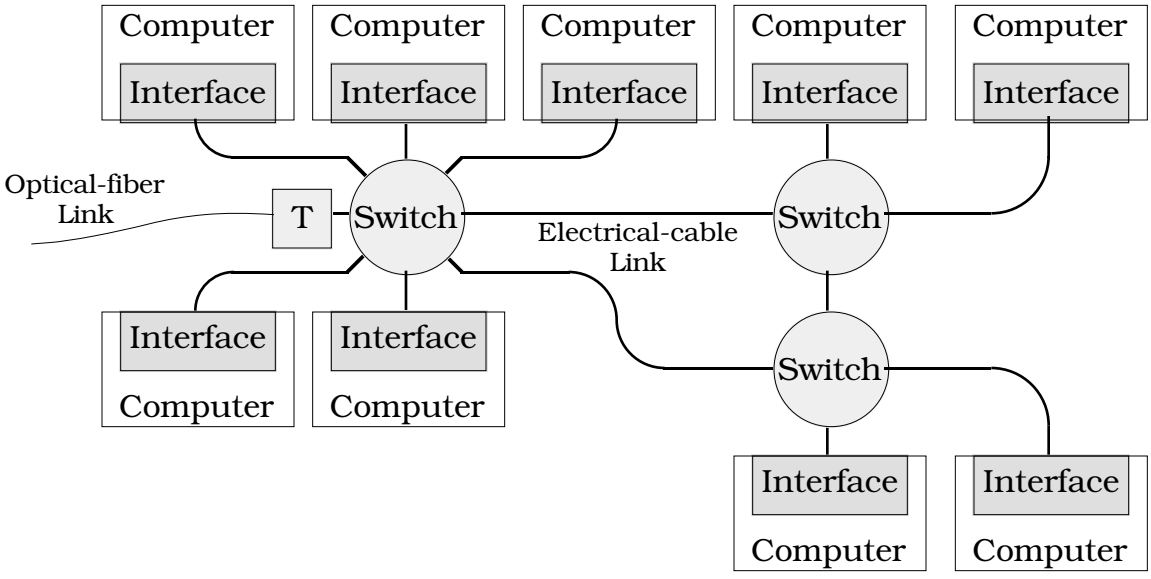


Figure 2: A possible configuration for a Myrinet local-area network.

**Electrical-Cable Links.** The standard Myrinet cable is composed of 18 twisted pairs, 9 in each direction, and is driven and sensed in a balanced, differential mode with ~1V signals and ~2V differences. The cable is physically ~1cm in diameter, shielded, flexible, and UL-approved type CL-2.

These load-terminated transmission lines have a characteristic impedance of ~105Ω, and a propagation velocity of ~0.6*c*.

Transmission is synchronous at the sending end at a rate of 80M 9-bit characters/s. The 9-bit character may be either an 8-bit data byte or one of five *control symbols*. The signal encoding is non-return-to-zero (NRZ), in which a signal transition encodes a binary 1 and the absence of a transition encodes a binary 0. The all-zeros character is the IDLE control symbol, which is discarded by not being detected.

The receiver circuits are asynchronous; a character may arrive at any time relative to the clocks in the receiving system, and is later synchronized to those clocks by pipeline-synchronizer circuits [9]. Delay variations and crosstalk between cable pairs and delay variations between line drivers and receivers skew the transitions at the receiving end of a channel. The receiver circuits employ a sampling window technique that tolerates skew up to half the 12.5ns character period. These conventional encoding and sensing techniques result in a bit error rate (BER) well below $10^{-15}$ on cables up to 25m long. The signal distortions, some of which are quadratic with length, and cumulative skew make low-error-rate operation over longer distances progressively more difficult, so 25m was chosen as the maximum length for electrical cables. Longer distances may be spanned either using cable repeaters or the optical-fiber links described later.

Flow control is accomplished by the receiver injecting the control symbols STOP and GO into the opposite-going channel of the link. There can be up to 23 characters in transit on a round trip of a 25m Myrinet cable, and the STOP and GO symbols may require several character periods to generate and decode. The receiver, accordingly, includes a queue-organized "slack buffer" that operates conceptually as pictured in Figure 3. If the downstream flow is blocked so that the slack buffer fills to the STOP line, the receiver generates a STOP control symbol so that the flow will stop before the buffer overflows. When the downstream flow starts again, the receiver generates a GO control symbol when the level reaches the GO line. The top part of the buffer prevents over-runs; the bottom part of the buffer prevents data starvation. The buffer positions between GO and STOP provide hysteresis that assures that STOP and GO control symbols will not consume excessive bandwidth on the opposite-going channel.

The GAP control symbol marks the end of a packet, and may, along with GO and STOP, appear redundantly. The sender is required to emit a non-IDLE character periodically as a mechanism analogous to carrier sensing for detecting open links. There are also long-period timeouts for detecting packets blocked for more than ~50ms, as may occur if a software error or a bit error in a header has caused a deadlock. This long-period timeout causes the blocked part of the sender's packet to be dropped, and a forward-reset (FRES) control symbol to be sent to the receiver.

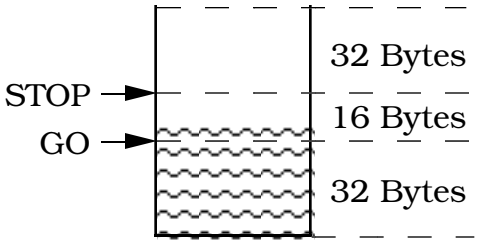


Figure 3: Operation of the Slack Buffer.

These Myrinet links completely solved the first set of practical limitations of the ATOMIC LAN.

**Packet Format and Routing.** The format of a Myrinet packet is illustrated in Figure 4. When a packet enters a switch, the leading byte of the header determines the outgoing port, and is then stripped off of the packet. When a packet enters a host interface, the leading byte identifies the type of packet, e.g., a mapping packet, a network-management packet, an packet with an IP packet as its payload, or data carried by a light-weight protocol. The most-significant bit of each header byte distinguishes between "to-switch" and "to-host" bytes. If all packets traveled known routes, this bit would be redundant. However, the redundancy in the encoding of the header allows the interfaces to detect and deal with faults that cause misrouting, and simplifies network mapping by allowing switches to drop "to-host" packets and hosts to drop "to-switch" packets.

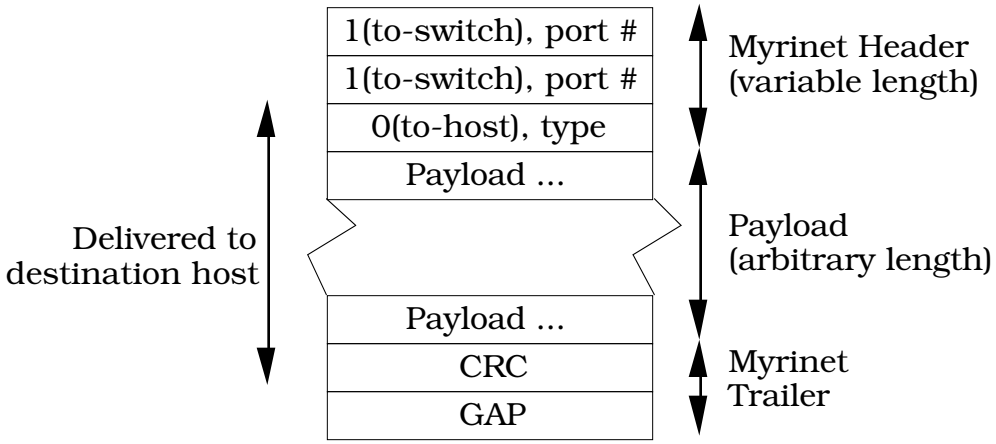


Figure 4: Format of a Myrinet Packet

The payload of a Myrinet packet is of arbitrary length; hence, it can carry any other type of packet (e.g., an IP packet) without an adaptation layer.

The 8-bit cyclic-redundancy-check (CRC) character is computed on the entire packet, including the header, and is carried in the packet trailer. Because the packet header is modified at each switch, the CRC is recomputed on each link. If the CRC on a packet is incorrect when it enters a switch, it will be incorrect in the same bit positions when it leaves the switch. Thus, if there is an error on any link on a routing path, the error can be detected at the destination.

**Myrinet Switches.** Myricom is currently shipping 4- and 8-port switches, and developing 16- and 32-port switches. These switches employ exactly the same blocking-cut-through (wormhole) routing [6] used in the message-passing networks of such MPP systems as the Intel Paragon and the Cray T3D. The worst-case (path-formation) latency through an 8-port switch is 550ns. The core of the switch is a pipelined crossbar, which introduces no internal conflicts between packet flows. In addition, recirculating-token arbitration assures fairness (no head-of-line priority problems).

These switches are based on two custom-VLSI chips, a crossbar-switch chip that forms the switch itself, and a dual-Myrinet-interface chip that performs the autonomous parts of the Myrinet-link protocol, including the flow control. Based on current chip areas and performance in 0.8µm CMOS technology, there is a great deal of "headroom" for continued evolution of these switches.

Figure 5 is a photograph of a 4-port switch. The principal design criteria for these switches were that they be physically small, low-power, self-initializing devices that can be placed in wiring closets, over ceilings, or in any other location convenient for cabling the network. This 4-port switch requires ~15W of +12V ± 3V power from either or both of two supplies, so that battery backup can be provided easily. These switches operate correctly at ambient temperatures up to 55C. Because the switch contains only transient state and no software, there are no security issues concerning which hosts are allowed to download programs or routing tables into the switch. All the switch does is steer packets.

The Myrinet packet format and switches have entirely resolved the second practical limitation of the ATOMIC LAN.

So much has been written and argued about MPP-network topologies -- hypercubes, meshes, tori, fat trees, and others -- that the question often arises how to classify Myrinet. The Myrinet topology is arbitrary, but is based on high-degree switches. To make a specific performance comparison, let's look at how well Myrinet can simulate or replace a two-dimensional mesh. A single 32-port-switch chip (~900 pins) could replace the 16 mesh-routing chips (132 pins each) in the 4x4 routing module that serves 16 nodes in the Intel Delta multicomputer. Sixteen switch links would connect to the nodes, and 16 would simulate the edges of the 4x4 mesh. The high-degree crossbar switch would replace 16 chips with one, halve the total pin count, introduce fewer internal conflicts, and reduce the

diameter of the network. Indeed, since the external links could be long, one would not even need to connect the 16-node modules in a mesh, but could use a lower-diameter network. The moral is that the precise details of the topology have less influence on network performance than the degree of the switches from which the network is constructed.

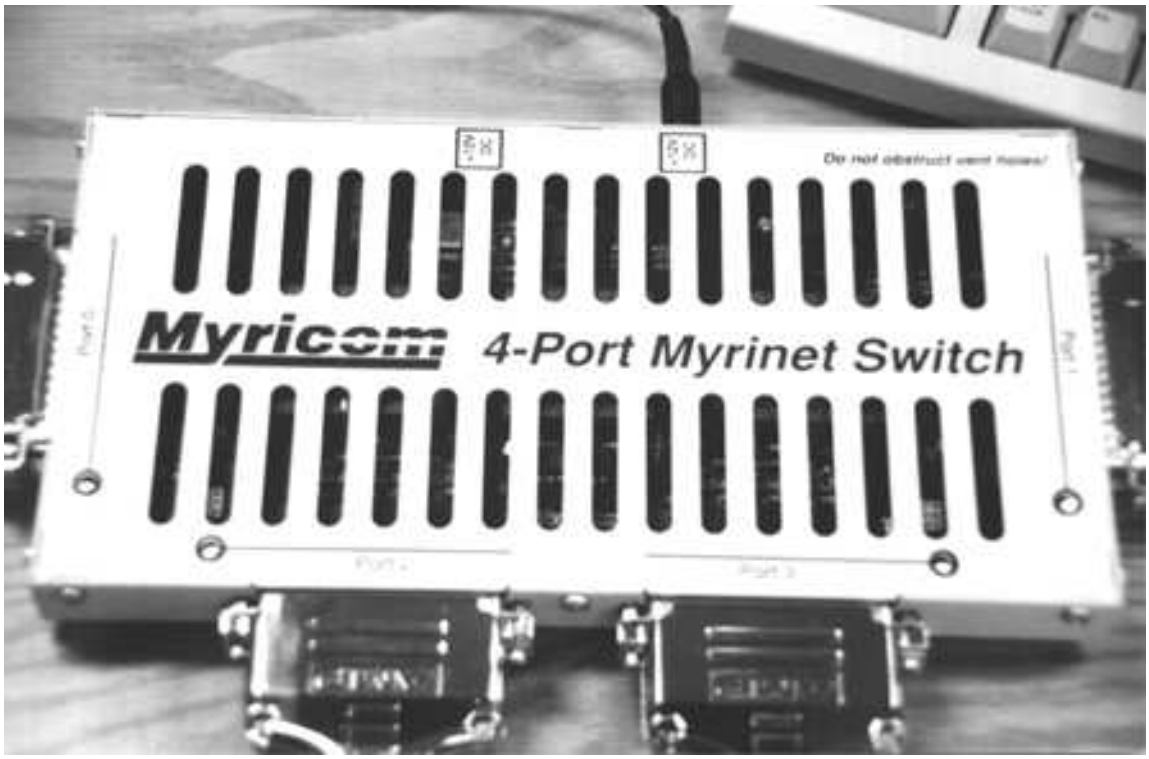


Figure 5: Photograph of a 4-Port Myrinet Switch. Note the keyboard for scale. This switch has a bisection bandwidth of 2.56 Gb/s, and operates on 15 Watts.

**Myrinet Host Interfaces.** Figure 6 shows the organization of a host interface and the internal details of the custom-VLSI LANai chip on which the interface is based. This microarchitecture provides a flexible and high-performance interface between a generic bus called the E-bus and a Myrinet link. Myricom is shipping Myrinet/SBus interfaces for Sun SPARCstations -- a post-card-size circuit board, thanks to the level of integration provided by the LANai chip --, and is developing interfaces to several other buses, including PCI.

The SRAM is used to store the Myrinet Control Program (MCP) and for packet buffering. This 32Kx32 memory is accessed twice in each clock period, once by the E-bus and once by the processor or packet interface. Because E-bus accesses are not arbitrated, the LANai appears from the E-bus

as synchronous memory, and can be placed as a bus slave on practically any 32-bit memory or I/O bus. In addition, when the DMA engine is used to provide addresses, the LANai can act as a bus master to transfer data blocks between the E-bus and SRAM. The DMA engine also computes the Internet checksum of the data it transfers.

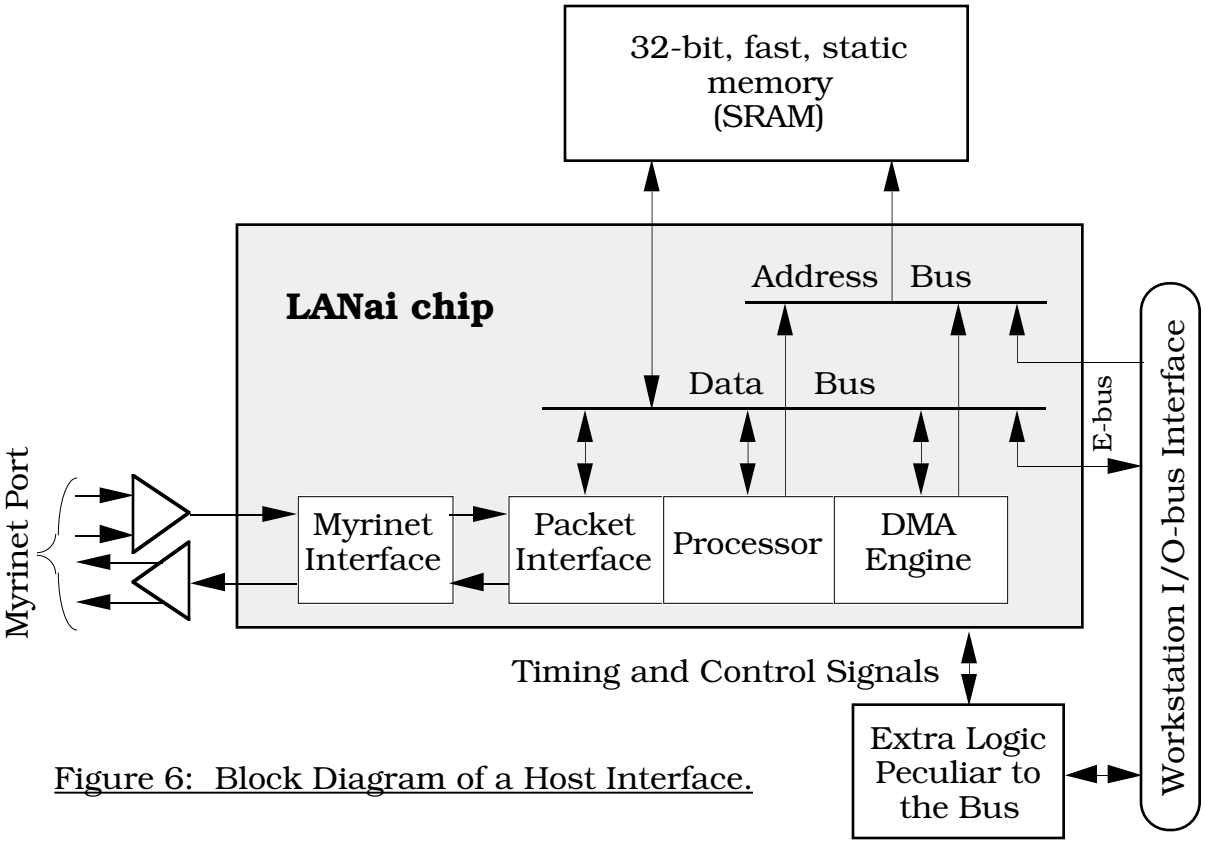


Figure 6: Block Diagram of a Host Interface.

Myrinet host interfaces retain all of the best characteristics of the ATOMIC host interfaces, but the addition of the DMA engine with Internet-checksum computing resolves the third practical limitation of the ATOMIC LAN.

**Optical-fiber Interfaces.** Myrinet optical-fiber interfaces connect on one side to an electrical Myrinet port and on the other side to a fiber pair. In terms of the ISO Reference Model for computer networks, the optical-fiber link and interfaces can act both as a level-2 bridge and as a level-3 router.

A LANai chip, its associated memory, and a specialized version of the MCP handles the optical-fiber interface's electrical Myrinet port, and provides buffering for the ~1500B/Km of slack required to maintain flow control over the optical link. When two optical-fiber interfaces connected by fiber are used as a level-2 bridge, they maintain the same logical characteristics as a Myrinet link. However, optical-fiber links may be used to connect different Myrinets, for example, in different buildings. It may be needless for each

host in one building to maintain a map of the network in another building, so the optical-fiber interfaces allow the networks to be separate. As a level-3 router that employs a "MessageWay" protocol, an optical-fiber interface participates in the mapping of its own network, but advertises itself as a route to another Myrinet. A packet routed to an optical-fiber interface may then contain an address in another network, and the translation from this address to a route in the other network is performed after the packet is sent across the optical fiber link.

**Software**

The software that controls and provides access to a Myrinet is divided into the Myrinet Control Program (MCP) that executes on the processors in the host interfaces, and the device driver and operating system that execute in the host.

**Myrinet Control Program.** The MCP is loaded initially by the device driver when the host boots, and starts executing as soon as the device driver releases a reset signal. The MCP thereafter interacts concurrently with its host and with the network.

On the host side, the interface is controlled by a set of single-producer-single-consumer command and acknowledgment queues. A typical command from the host is for the MCP to perform a gather operation on a set of data blocks at specified host addresses and word counts, to generate the Internet checksum on the resulting packet and insert it in a specified location in the packet, to send the resulting packet to the host that has a specified network address, and to signal the completion of these operations in an acknowledgment queue and optionally by producing an interrupt for the host. Performing such a command involves controlling the DMA engine, a translation from a network address to a route, prepending the route and packet type to the outgoing packet, and controlling the packet interface. In the receiving direction, the MCP checks the validity of the incoming packet, interprets headers, transfers packet data to specified scatter buffers in the host memory, and signals the arrival of a packet in an acknowledgment queue and optionally by producing an interrupt for the host.

Interleaved with these interactions with the host, the MCP performs continuous mapping and monitoring (remapping) that makes the network self-configuring and self-healing, route selection, and store-and-forward multicast. One of the host interfaces in each Myrinet is selected to map the network by sending mapping packets to other interfaces and to itself. The selection of the mapper can be done either manually, by network-manager intervention, or automatically by the Myrinet itself. In particular, if the mapping interface's host is turned off, or if faulty links cut the Myrinet into disjoint networks, one or more new mappers are selected automatically. The map of the network is distributed by the mapping interface to the rest. Each interface then computes the routes from itself to all other interfaces, and all of these routes are guaranteed to be deadlock-free.

The only difficult part of the mapping algorithm is the identification of switches that do not connect to a host. Host interfaces have a network address that identifies them uniquely, but switches do not. If a switch connects even to one host interface, it can be identified easily, but if it connects only to other switches, its identity must be inferred by the routes through the switch. Although the details of these operations are beyond the scope of this paper, Myricom regularly provides the MCP source code and programming tools to its research customers so that they may tailor interfaces to their own needs, and plans eventually to publish the source code of the MCP as part of our policy of making Myrinet interfaces open.

In the interest of network security, the MCP includes no "back doors" that might allow another network host to substitute a different version of the MCP. Indeed, the segment of interface memory in which the MCP resides is write-protected from the MCP itself, and can be loaded only from the host.

**Host Software.** The Myrinet host software provides the interface between UNIX user processes and the host-interface board. Myricom delivers both a standard TCP/IP and UDP/IP interface, and a streamlined Myrinet Application Programming Interface (API). Figure 7 is a "copy diagram" that illustrates the steps involved in getting information from a user process to the network. For receiving rather than sending, just reverse the arrows.

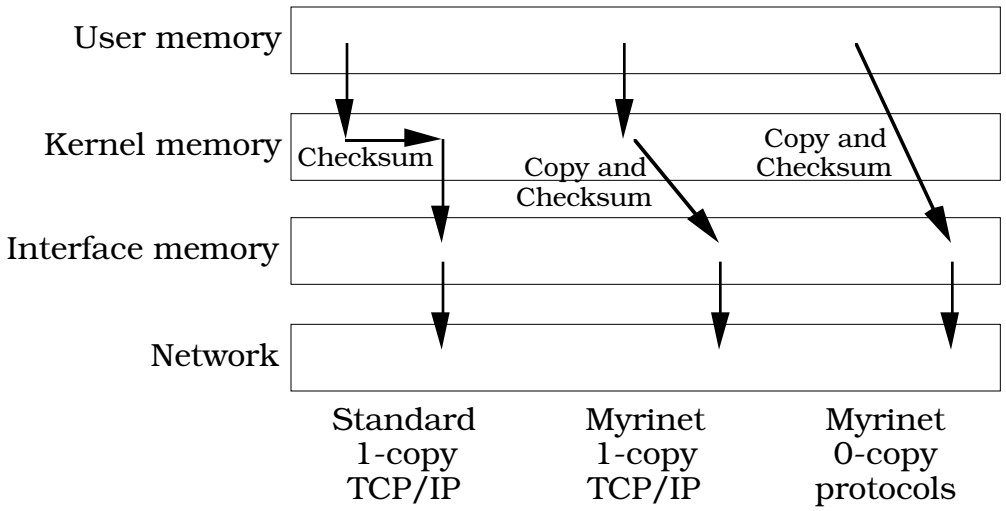


Figure 7: Copy Diagram for different software-interface implementations.

In the standard, "one-copy" TCP/IP interface, UNIX copies the specified block from user space to kernel space, computes the checksum either on a separate pass through the block or as part of the copy, and gives the host interface a command to send the packet. The MCP copies the data to the LANai memory using the DMA engine, and then transfers the packet to the network using the packet interface. The data is moved at least three times,

but the DMA and packet-interface operations are (nearly) free for the host, so this scheme is referred to as a "one-copy" implementation. In operating-system implementations that allow it, the Myrinet host interface can off-load from the host the computation of the Internet checksum. Indeed, the Myrinet host interface was designed to allow a zero-copy mode of operation directly from user space, but the workstation manufacturers do not distribute a version of the TCP/IP or UDP/IP protocol stack that can fully exploit the capabilities of the interface.

In order to approach the performance of the I/O bus, let alone of the network, it is necessary to bypass the operating system. The Myrinet API sets itself up by system calls that allocate a number of unswappable pages of memory used as a data-exchange area, and thereafter avoids UNIX system calls. User processes can manage the command and acknowledgment queues using a library of Myrinet-API functions.

| | |
|---|---|
| Channel data rate | 640 Mb/s |
| **SPARC-2, 20MHz SBus** | |
| DMA data rate | 444 Mb/s |
| Raw speed test | 380 Mb/s |
| Myrinet API (8KB packets) | 250 Mb/s |
| TCP/IP (0-copy, hardware checksum) | 70 Mb/s |
| UDP/IP (1-copy, hardware checksum) | 55 Mb/s |

Figure 8: Measured Performance (on 4 August 1994)

Figure 8 tabulates several benchmarks performed on a Myrinet connecting SPARC-2 workstations. The SPARC-2 has a 20MHz SBus but supports 16-word burst transfers, with the result that the DMA performance is better than for 25MHz SBus machines that implement only smaller burst sizes. A raw speed-test program achieves one-way, end-to-end rates on 8KB data blocks that are nearly as fast as the SBus. Myrinet API functions achieve one-way, end-to-end rates of 250Mb/s on 8KB blocks, a performance level that is limited by the number of instructions that the interface's processor executes to handle each packet. The Myrinet-API performance figures between UNIX processes on workstations are entirely comparable to those between nodes of commercial MPP multicomputers. The performance of the standard UDP/IP interface with the hardware performing the Internet-checksum computation is less than 1/10th of the channel bandwidth, but is high enough to satisfy the needs of many applications. The zero-copy

TCP/IP result was achieved with an experimental protocol stack developed by researchers at Sun Microsystems.

The Myrinet software deals with certain aspects of the fourth practical limitation of the ATOMIC LAN, notably by the MCP handling much of the network control in the interfaces themselves, and by the streamlined access provided by the Myrinet API. To reach the full potential of fast networks, however, it will be necessary for workstation and PC manufacturers to provide faster I/O buses, and particularly operating systems that can exploit the capabilities of interfaces that can interact directly with user processes.

**Conclusions**

Our group is particularly pleased to be described at Hot Interconnects II as another success story of carrying basic-research results -- in this case in multicomputer-technology and the ATOMIC-LAN projects, both sponsored by ARPA -- into commercial practice. We have taken on this technology transition because we believe that Myrinet will have a positive impact on computing practice in several areas.

Myrinet demonstrates the highest performance per unit cost of any LAN of which we are aware. Features such as self-configuration and fault tolerance make it useful also as an I/O fabric, high-reliability MPP network, or platform bus. As might be expected from its MPP genealogy, Myrinet is ideal for cluster-computing applications. With the addition of optical-fiber links, Myrinet can also provide high-bandwidth, low-latency, low-error-rate communication of mixed packet types on networks up to Kilometers in diameter. Myrinet does not need to scale beyond the cabinet-to-campus range of diameters to be useful. For larger areas, it is straightforward to provide gateways between Myrinet and wide-area networks. For smaller areas, we can continue to use buses.

**References**

[1] Charles L. Seitz, Nanette J. Boden, Jakov Seizovic, Wen-King Su, "The Design of the Caltech Mosaic C Multicomputer," *Proceedings of the University of Washington Symposium on Integrated Systems,* pp. 1-22, MIT Press, 1993.

[2] Felderman, R., DeSchon, A., Cohen, D., Finn, G., "ATOMIC: A High Speed Local Communication Architecture," *Journal of High Speed Networks,* Vol. 3, No. 1 (1994) pp. 1-29.

[3] Finn, G. G., "An Integration of Network Communication with Workstation Architecture," *Computer Communication Review,* October 1991.

[4] William C. Athas, Charles L. Seitz, "Multicomputers: Message-Passing Concurrent Computers," IEEE COMPUTER 21(8): 9-24, August 1988.

[5] Charles L. Seitz, "Multicomputers," Chapter five in *Developments in Concurrency and Communication,* edited by C. A. R. Hoare, Addison-Wesley, 1990.

[6] Charles L. Seitz, Wen-King Su, "A Family of Routing and Communication Chips Based on the Mosaic," *Proceedings of the University of Washington Symposium on Integrated Systems,* pp. 320-337, MIT Press, 1993.

[7] *Interconnection Networks for High-Performance Parallel Computers,* I. D. Scherson and A. S. Youssef, eds., IEEE Computer Society Press, Los Alamitos, Calif., 1994.

[8] "Myrinet Link and Routing Specification," http://www.myri.com/myricom/documents.html.

[9] Jakov N. Seizovic, "Pipeline Synchronization," *Proceedings of the International Symposium on Advanced Research in Asynchronous Circuits and Systems,* November 3-5, 1994, Salt Lake City, Utah, IEEE Computer Society Press, 1994.