For: finn
Printed on: Fri, Aug 26, 1994 15:57:35

Document: Multicomputer-Parts
Last saved on: Wed, Jan 13, 1993 13:42:14

The Use of Message-Based Multicomputer Components to Construct Gigabit Networks

Danny Cohen, Gregory G. Finn
Robert Felderman and Annette DeSchon [1]
USC/Information Sciences Institute

Abstract

The typical node of a message-based multicomputer consists of a microprocessor, router and memory. At the California Institute of Technology, the Mosaic project has integrated such a node onto a single chip. That reduction in scale fundamentally changes the scope of node application, since nodes become both very small, and inexpensive.

Mosaic nodes may be employed to process, to generate, or to receive data. Since the router in a Mosaic node is independent of the microprocessor, computation and routing take place simultaneously. These nodes may be used to create general purpose gigabit LANs. They may also be used to create special purpose gigabit networks to interconnect instrumentation within spacecraft or aircraft.

The ATOMIC project at USC/ISI is using Mosaic components to prototype a gigabit LAN testbed. This testbed is operational. Networking and administration software provides full TCP/IP compatibility. Packets have been exchanged between two interfaces at a rate above one gigabit per second (Gb/s).

An individual ATOMIC interface is both inexpensive and small, consisting of one Mosaic chip, four SRAM chips and clock logic. Two interfaces easily fit onto a postcard-sized circuit board. Their low cost makes it practical to include several interfaces within a host, providing an interior Gb/s distribution network, multiple access points to the LAN for greater performance or redundancy, and other capabilities that are not yet fully explored. The results reported in this paper represent actual data obtained from the prototype.

1. Overview

Advances in integration now make it possible to place processing, memory, routing, and channel-access logic within a single chip. This is the result of a decade-long trend in message-based computer architecture. These developments allow designers to include support for gigabit point-to-point channels in their architectures.

Successive refinements are seen in the progression from the Caltech Cosmic Cube of the early 1980s, to the Intel iPSC, the AMETEK Series 2010 and today's Intel Touchstone message-based supercomputers [1]. Closely related is the development of the INMOS Transputer chip series.

An investigation into the new issues that arise when a Gb/s of traffic arrive or depart from a host is needed. Will current protocols adapt to these rates? Are the operating system methodologies to support networking sufficient? Are workstation architectures adequate to send/receive Gb/s streams or should they be radically changed? Will special service types and bandwidth reservation be needed? What new applications might arise, assuming that Gb/s service is available? To answer questions such as these it is useful to have a Gb/s testbed in which experiments can occur.

This suggests the need for flexibility. The ultimate in flexibility is achieved when *all* the components of the network are programmable. The ATOMIC effort at ISI grew out of recognition that Mosaic multicomputer components provide that flexibility.

Many routing strategies can be implemented by programming the nodes, and the source-routing strategy currently adopted in the testbed is only a starting point.

This research was sponsored by the Defense Advanced Research Projects Agency under Contract No. DABT63-91-C-0001. 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 DARPA, the U.S. Government, or any person or agency connected with them.

[1] University of Southern California/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292
Internet: Finn@isi.edu

Distributed routing algorithms, among others, will also be investigated.

Utilizing Mosaic technology allows this project to create a Gb/s LAN testbed where all these issues can be examined and possible solutions studied. This paper discusses how the ATOMIC project applies Mosaic components to create that testbed, and in the process, what obstacles to Gb/s networking have been discovered and what approaches show promise for overcoming them.

The next section of the paper acquaints the reader with some aspects of the Mosaic technology developed by Chuck Seitz's research group at Caltech. The Mosaic project is supported by DARPA and is aimed at exploring fine-grained multicomputers.

What follows that is a discussion of the testbed itself, measurements taken from it, and a discussion of issues closely related to the questions mentioned above. Readers who are familiar with Mosaic can skip ahead to Section 3, although the material in Section 2e is recent.

2. Aspects of Mosaic Technology

Mosaic messages are of variable (even) byte length. Depending upon the chip version, from four to eight simplex channels are supported. Each channel operates at a nominal rate of 0.5 Gb/s, while prototypes have operated at 0.8 Gb/s. A full-duplex link constructed from a pair of these channels provides 1 Gb/s of transfer capacity.

Alternative point-to-point gigabit multicomputer technologies are being developed elsewhere, such as the Scalable Coherent Interface or SCI, IEEE P:1596 [2]. For several reasons Mosaic is a more appropriate choice for creating a Gb/s testbed. Principal among these were Mosaic's use of CMOS rather than ECL logic levels and its greater flexibility.

2a. Mosaic Nodes

Each Mosaic node contains a 16-bit microprocessor, an independent router and a DMA message interface between them [3]. The router is implemented by self-timed logic. It can route traffic simultaneously over all of its external channels without using any processor resources. Messages are source-routed in two dimensions.

Source routing has been suggested as a means to create very high-speed networks [5]. Source routing is flexible and responsive. Each message is independently routed and no end-to-end connection set-up is required. The routing decision in a Mosaic router requires no more than 25 ns, and state information in a router is kept only for the duration of the message being routed.

Two categories of Mosaic nodes exist, differentiated by whether or not they support external memory. A

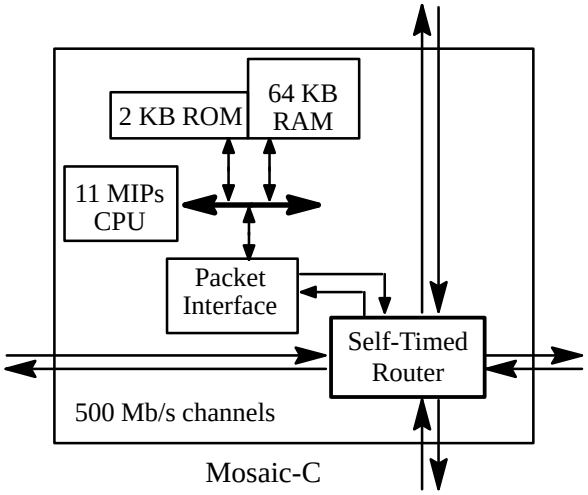


Figure 1.

Mosaic-C node is depicted in Figure 1. It contains 64 KBytes of RAM, 2 KBytes of ROM and communicates over eight external point-to-point channels, four in the **X**-direction and four in the **Y**-direction [6]. All eight channels may be active simultaneously. Software can be remotely loaded via incoming channels.

Figure 2 depicts two types of Memoryless Mosaic nodes. They are distinguished by having an external memory bus and fewer channels.[1] Each supports 128 KBytes of external dual-access memory for program, data and message storage. Memoryless Mosaic nodes are well suited for interfacing to peripherals. These two Mosaic node types are sufficient to construct a Gb/s LAN testbed.

Unless a node is the source or destination for a message, messages pass through its router on their way to other nodes without interrupting the processor and without being stored at that node. When a node is either source or destination, packet data is transferred to or from node memory by a DMA controller in the packet interface.

A node can be utilized to filter messages, execute protocols and arrange that data be delivered in the form expected by an application or virtual device specification. A supervisor can control, reload or augment a node's software dynamically by sending it messages.

A Mosaic chip interfaces *directly* to its communications medium. No additional circuitry is required to interconnect two nodes as long as wire lengths between them are kept within certain parameters (see Section 2e).[2]

2b. Some Mosaic Properties

The router in Mosaic supports a form of cut-through routing [7] called *wormhole routing* [3][4], which does not utilize intermediate buffering when a route is blocked. Routing is two-dimensional, position-relative, and free of deadlock [8][9]. Each message is an

[1] The subset of channels in Memoryless Mosaics is due to the desire to keep the pincount down.

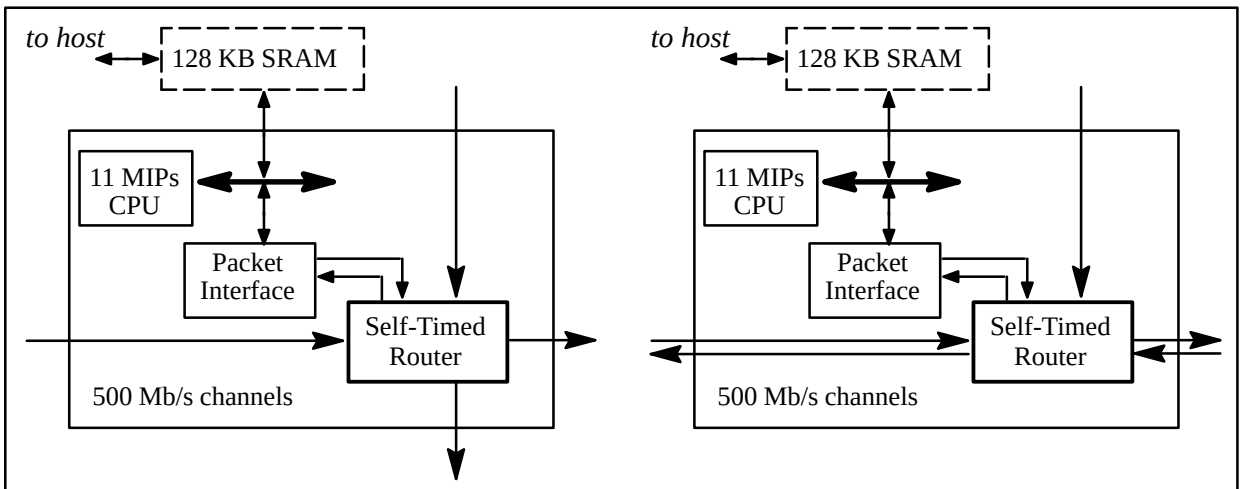


Figure 2. Memoryless Mosaic Nodes

even number of bytes in length and contains a source route prefix that consists of a Δ**X**-byte followed by a Δ**Y**-byte. Each routing byte can specify from -127 to +127 hops, with the sign controlling the +/– (East/West) direction for the Δ**X**-byte or the +/– (North/South) direction for the Δ**Y**-byte.

X-direction routing occurs before any **Y**-direction routing to avoid deadlock. Hop counts are decremented as they pass through each router. When the Δ**X**-byte reaches zero, the **X**-direction portion of the source route is exhausted and the Δ**X**-byte is stripped off to expose the Δ**Y**-byte. The **X**-channel portion of that node's router then passes the message to its **Y**-channel portion. A node receives a Mosaic message when the Δ**Y**-byte has reached zero. By implication, messages traverse a restricted two-dimensional topology that allows a single **X**-to-**Y** transition.

A path is opened between source and destination routers by advancing a message hop-by-hop along its path, allocating channels as it progresses. All the routers along that path ship the message toward its destination as a cooperative pipeline. At any node, a message may be blocked because the outgoing channel needed is already being used by another message. The arbiter in that node will allow the blocked message to advance when the outgoing channel becomes free. Arbitration operates on a first-come first-served basis.

The routers use Request and Acknowledgement (REQ and ACK) signals to implement hop-by-hop, byte-by-byte self-timed flow control on each channel. See Chapter 7 of *An Introduction to VLSI Systems* [10]. A TAIL signal is used to indicate the end of a message. As the tail of a message passes through a router, the channels that were allocated to it are freed.

If a message is blocked due to congestion, data flow ceases. This freezes the pipeline until the packet is again allowed to proceed. Once a message head reaches the destination node, all necessary channel resources have been allocated and data moves through the pipeline as fast as the source provides it and the destination is able to accept it.

Byte propagation time from one Mosaic router to another is nominally 12.5 ns while a routing decision consumes approximately 25 ns. Cable delay is 5 ns/meter and becomes the dominant factor in determining data rate for distances greater than 60 cm. The issue of how to convey data across channels at Gb/s rates over distances of up to 100 meters is discussed below.

2c. Comparison with a Bus

Figure 3 depicts four messages simultaneously in transit between nodes. Point-to-point channel segments may be used simultaneously. A message sent from node 1 to node 7 interferes neither with messages sent from node 4 to node 6 nor from node 5 to node 2. Neither does the message sent from node 8 to node 9 provide interference. Although the chain of Mosaic channels 1-6 is topologically similar to a bus with six taps, its linear bandwidth scaling properties make it much more attractive for block data movement.

2d. Mosaic-C Mesh

A set of 64 Mosaic-C nodes are mounted on an approximately 20cm × 20cm printed circuit board using TAB (Tape Automated Bonding) with their channels interconnected to form an 8×8 mesh. The resulting 64 node multicomputer is smaller in area than this journal page and has an aggregate processing rate of at least 700 MIPs. These mesh boards form the basic element of the Mosaic multicomputer. Emerging from the edges of a

2 For comparison, an interface to the 32-bit variant Futurebus+ backplane may require ten chips. Source: National Semiconductor, National Anthem vol. 23, March/April 1991, p. 3.

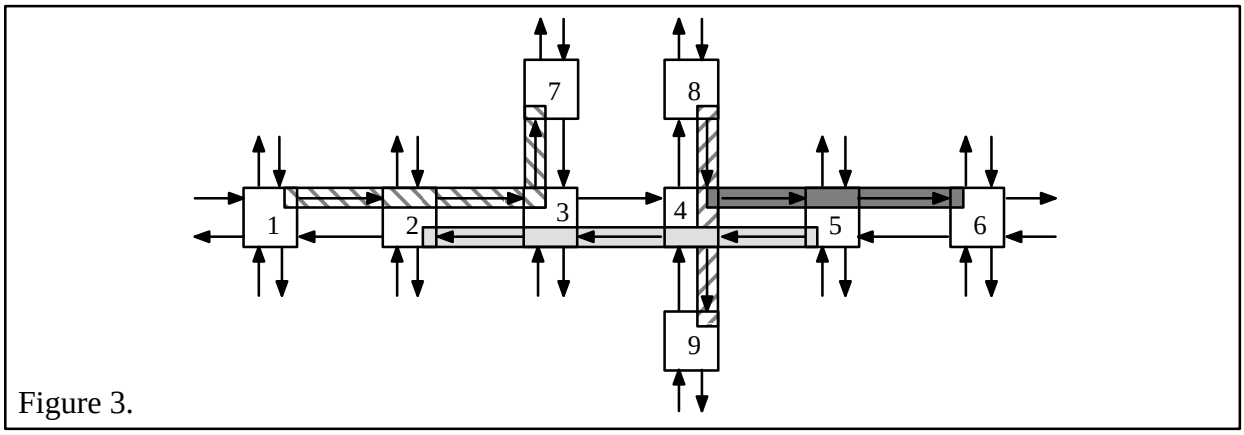


Figure 3.

mesh board are 32 duplex channels. These channels can be directly connected to another mesh board or can be connected via cables to remote Mosaic nodes. The current cost of this board including the 64 Mosaic-C chips is less than $5,000.

Extensive modelling and simulation has been performed at Caltech to determine the message routing performance of a supercomputer constructed of a tiled mosaic of mesh boards [3][11]. The channel-bisection of a mesh is the minimum number of channels that must be cut to partition the mesh in either the **X**- or **Y**-direction. For an $n \times n$ mesh that is $2n$ channels in either direction.

Assume that each node continuously sends messages to other nodes in the mesh with the destination nodes chosen randomly. Under those conditions the achievable throughput across a bisection is approximately 50% of its maximum theoretic capacity. Since individual Mosaic channels nominally provide 0.5 Gb/s of capacity, each channel in a bisection should provide 0.25 Gb/s of capacity in its direction. An 8×8 mesh is expected to provide 4 Gb/s of throughput across each of its four edges.

2e. Interconnecting Mosaic Nodes Over Distances of Many Meters

Mosaic channels may be interconnected using a number of technologies. Simple ribbon cables are adequate for distances up to a few meters without any additional logic. Ribbon cables provide a robust and very inexpensive interconnection. Their 26 conductors alternate ground with the 11 channel signal lines. The Mosaic chip output drivers are designed to drive 3 meter cables. No data errors have been seen in the LAN even while driving 9 meter cables. Channels are self-timed. No clock signal need be propagated.

Data Integrity

Mosaic channels exhibit extremely high data fidelity. Signalling is non-interfering, with the sending router transmitting 8 Data, plus REQ and TAIL signals to the receiving router, that subsequently returns ACK. The sending router's signal hold time and byte-by-byte flow is controlled by the receiving router.

Mosaic channels transmit no error detection or correction information. This was a cause of some concern. A series of data transfer validation tests were run over Mosaic channels to characterize their bit-error rates. The Mosaic channel data was transmitted via on-board plated traces and between hosts by ribbon cables.

Various checkerboard data patterns were sent from Mosaic nodes over ATOMIC and checked upon their reception. Typically, test sources sent packets over a multi-hop path. For short distance transmission via ribbon cables and chip-to-chip on-board transmission, the bit-error rate is very low. No errors have been seen to date, with 1.0 Petabit (10^{15} bits) of data transmitted in over 2,000 hours of validation testing.

This suggests that neither CRC nor parity are needed for Mosaic channel transmission within a chassis or when the channel distance between adjacent Mosaic chips is within a few meters. This level of fidelity should not necessarily be expected for longer distance transmission over differing physical media.

Cable Data Rate

The REQ/ACK negotiation of Mosaic channel transmission makes the transmission rate sensitive to internode distance. Nominally, a Mosaic channel may turnaround in 12.5 ns. This is true for internode distances over ribbon cables of up to 60 cm. As internode distance rises above that, the channel transmission rate falls.

Retaining the 12.5 ns channel cycle rate over distances of up to 100 meters typically encountered in a LAN is accomplished by decoupling local ACK reception from remote ACK generation. The transmitting end allows locally generated 'fake' ACKs to get ahead of the ACKs received from the far end by an amount no greater than the size of a reception FIFO. Each stage of the FIFO provides an additional 60 cm or so of 'slack' distance

while preserving full data rate over the cable. These SLACK chips were designed by Wen-King Su at Caltech. Samples will soon be tested in ATOMIC.

SLACK functionality combined with ribbon cables does allow nodes to be located many meters apart while preserving the channel data rate. Ribbon cables are not an attractive medium to be running above ceilings and inside walls. Furthermore, dispersion between signal lines must be considered over many-meter distances. It may be necessary to utilize repeaters every several meters along a cable. The ATOMIC project will soon use ribbon cables, with SLACK repeaters installed, to separate workstations from one another over tens of meter distances.

Fiber-Optic Transmission

The practical solution to the problem of carrying channel data over longer distance may arrive when fiber-optic transceivers for Mosaic channels become available. This is an area actively being pursued by the ATOMIC project. Link controller chips that incorporate SLACK functionality are being commercially designed that provide an interface between Fibre Channel fiber-optic transceivers and Mosaic channels.

Summary

In the absence of blockage the rate at which data is carried by a Mosaic channel is determined by the following factors: the rate at which the source provides data, the router processing rate, by the cable segment delay, and finally, the rate at which the destination accepts data.

3. Creating a LAN using Mosaic Technology

Mosaic technology was developed for message–based multicomputing. The channel specifications and memory configuration of Mosaic nodes reflect that application domain. In particular, node memory bandwidth is roughly equivalent to the nominal rate of one channel. The ATOMIC project is using Mosaic nodes to create a gigabit LAN and they adapt well to that task. However, if nodes were designed specifically for networking, some aspects of their current design would likely change.

The combination of channels, router, storage and processor inside Mosaic nodes offers great flexibility when designing a network. Memoryless Mosaic nodes can be used in host interfaces while Mosaic-C nodes can be used much like routers (IMPs) in the ARPANET to implement hop-by-hop, store-and-forward routing. Node processors could run distributed routing algorithms, implement fair queueing, and so on.

In that type of network packets are written into node memory and then read out again. Therefore, channel performance is limited to no more than 1/2 the node memory bandwidth. A single Mosaic channel can operate at 100% of the memory bandwidth of a Mosaic–C node. However, a node router can forward data from four channels simultaneously. For that reason we are investigating network designs in which channel speed rather than memory bandwidth is the limiting factor on performance.

ATOMIC uses mesh boards to interconnect the hosts attached to it. Each host has a network interface that contains at least one Memoryless Mosaic node. The channels that emerge from it are typically attached via cables to a mesh board. Alternatively, hosts may be chained together, with one or both ends of the chain attached to a mesh board. Extensibility is provided by interconnecting mesh boards to one another via channels not already allocated to hosts. This is somewhat analogous to the way that Ethernet hosts are attached to concentrators. It is also similar to HUB-HUB interconnection scheme used in Nectar [12]. This is illustrated in Figure 4, where 4×4 meshes are illustrated for clarity. In practice, 8×8 meshes are used.

3a. Implication of Topological Irregularity

Any node within a mesh of Mosaic-C nodes can send a message to any other node in the same mesh using Mosaic's link-layer source routing. A path between any two nodes can be expressed as a number of **X**-hops followed by some number of **Y**-hops. However, the symmetry of mesh interconnection is broken when remote nodes are connected to the edges of a mesh.

That lack of symmetry prevents Mosaic's link-layer source routing from being sufficient to interconnect any two nodes. In Figure 4, node **A** can reach node **B** using the (*x*,*y*) source route prefix (3,4). Three hops in the +**X** direction reach the highlighted node, from which four hops in the +**Y** direction reach **B**. However, **B** cannot reach **A** using a single **X**-then-**Y** route, since the route **B**⇒**A** necessarily requires a **Y**-direction to **X**-direction transition somewhere in **B**'s column and that is forbidden. Note that node **C** is partitioned from both **A** and **B**.

Each Mosaic node contains its own processing and storage capability. One solution to this interconnection problem is to use those capabilities to implement store-and-forward functions. Each Mosaic node is loaded at initialization with software that implements an ATOMIC network layer allowing complex source routes to be composed of a series of Mosaic routes. Since Mosaic routing is deadlock-free, a source route composed of several successive Mosaic routes is also deadlock-free.

The ATOMIC source route **B**⇒**A** could now be the reverse of **A**⇒**B**, expressed as [(0,-4) (-3,0)]. The first Mosaic route from **B** is (0,-4), which delivers the packet to the highlighted node in the lower-right corner. That node then sends the packet with the source route [(-3,0)]. The (-3,0) Mosaic route causes the packet to arrive at **A**, which is the destination for this packet. Similarly, the path **B**⇒**D** is [(0,-4)(4,-1)(-1,0)].

This facility provides many paths between nodes and often there is a set of equivalent shortest paths

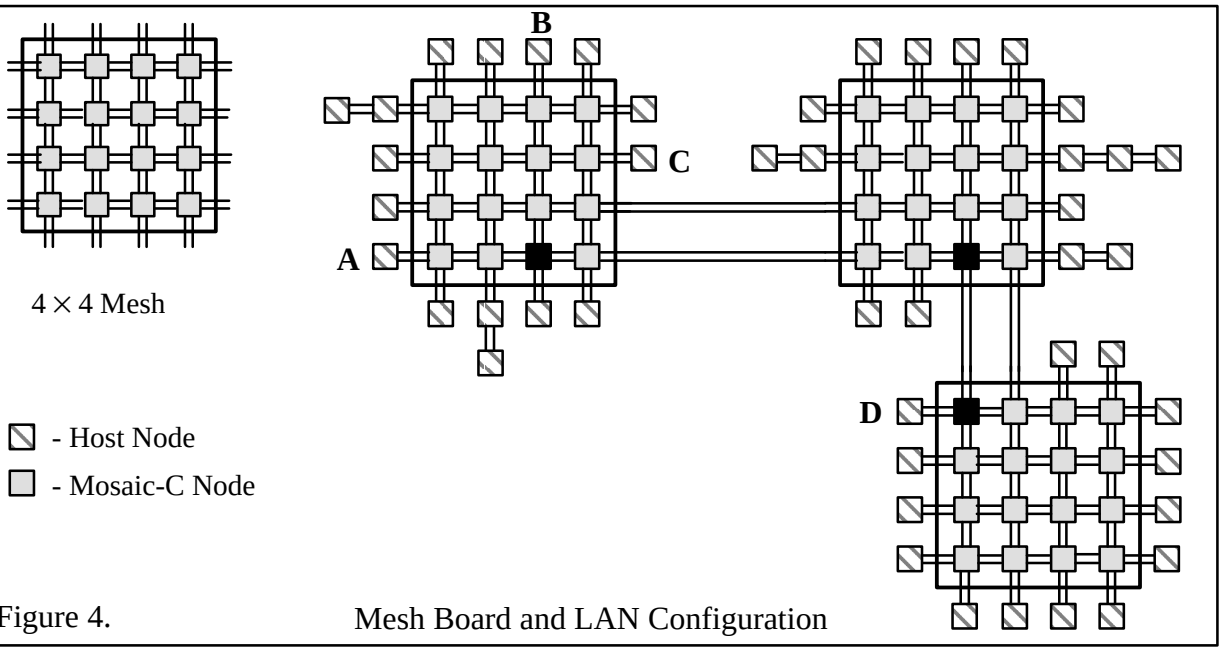


Figure 4. Mesh Board and LAN Configuration

between nodes. In the case of **A⇒C** or **C⇒A** those sets have four elements. This property could be used for bandwidth reservation, quality of service assurance or redundancy in case of component failure. Each of these is an issue that we expect to pursue in future work.

Although store-and-forward provides complete routing generality, it does affect the network transfer rate. Internal memory bandwidth is limited to the channel data rate. When a packet arrives at a node, the router transfers it into an internal buffer. Store-and-forward requires that a packet be both written and read, that memory bandwidth limitation restricts a store-and-forward operation to 1/2 the nominal 500 Mb/s channel transfer rate.

Since internal buffering is used for the ATOMIC layer store-and-forward routing, packets can be lost if intrnal buffers overflow. Under an assumption that 8 KBytes is reserved for software, 48 KBytes of buffering is available within each mesh node. That represents 780 μs of buffering at the 500 Mb/s channel rate. We intend to study flow control algorithms suitable for this environment.

3b. Dual-Connecting Nodes to a Mesh

Notice that any Mosaic source route that begins at the East or West edge of a mesh can reach any node at a North or South edge. This property can be exploited to connect remote nodes to a mesh so that any node can reach any other *without requiring the store-and-forward operation of an ATOMIC composite route*. Dual-connection to a mesh can be seen in Figure 5, where the **Y**-direction input of a Memoryless Mosaic host node is used.

Conflict for channel resources may only occur when more than one source is transmitting to the same destination node. There is one shortest path between any two nodes and that path does not require an ATOMIC composite route, which is a distinct performance advantage. Dual-connection of remote nodes to a mesh does in-

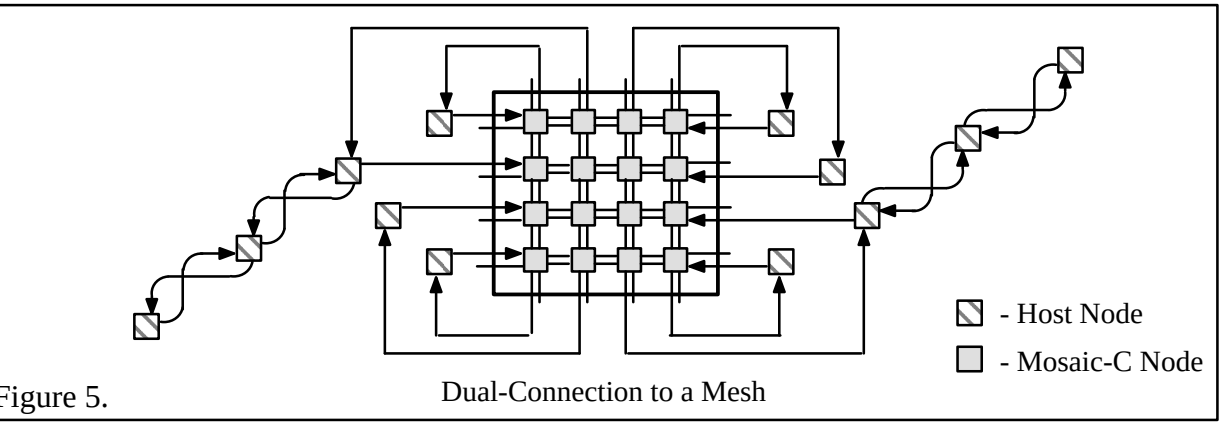


Figure 5. Dual-Connection to a Mesh

cur a cost. The number of nodes that can be directly connected to an $n \times n$ mesh is reduced from $4n$ to $2n$. However, packets cannot be lost in a mesh due to buffer overflow since no store-and-forward buffering is used.

3c. Y-to-X Transformation

The router determines whether a message should go out an **X** or **Y** channel based solely on which type of channel the message arrived on and the magnitude of the first byte. A message entering on an **X**-channel goes out an **X**-channel if the first byte is non-zero. A message entering on a **Y**-channel goes out a **Y**-channel if the first byte is non-zero. If the message enters on an **X**-channel and the first byte is zero, the message is passed to the appropriate +/– **Y**-channel after stripping the zero byte. A message that arrives on a **Y**-channel with a zero leading byte is delivered to the Mosaic processor after stripping the zero byte.

An outgoing **Y**-channel may be connected to an incoming **X**-channel or vice versa. The routing result remains well defined. However, there is a type of 'parity' restriction. The final **Y** routing byte stripped must preserve the evenness of the delivered message's length.

3d. Mesh Interconnection Using Y-to-X Transformation

As discussed in 3b, if a mesh only attaches to dual-connected nodes, any message entering this mesh via an **X** channel can reach any node strictly by means of Mosaic routing without store-and-forward. In Figure 6 we show two meshes interconnected by two **Y**-to-**X** channel transformations. The lower mesh via one of its **Y** output channels can reach an **X** input channel of the upper mesh. Similarly, the upper mesh via one of its **Y** output channels can reach an **X** input channel of the lower mesh. If both meshes only attach to dual-connected nodes, any node in either mesh can reach any other node strictly by means of Mosaic routing.

As an example, consider the route (4,3,3,2) from host **A** to **B** in Figure 6. With four +**X** hops the packet from **A** reaches the highlighted node at the right edge of its mesh. The current exhausted routing byte is discarded by that node, which now begins to route in the +**Y** direction using the remaining routing data (3,3,2). With three +**Y** hops the +**X** input of node **I** is reached. The current exhausted routing byte is discarded by **I**, which now begins to route in the +**Y** direction using the remaining routing data (3,2). One hop in the +**Y** direction reaches the **X**-direction input of a leftmost node in **B**'s mesh with the remaining routing data (2,2). The remaining two hops are now treated as +**X** hops by the nodes in **B**'s mesh. When these are exhausted the highlighted node in **B**'s mesh is reached, where the next routing byte (2) is treated as **Y**-routing information and **B** is reached with the final two +**Y** hops. The route from **B** to **A** is (-3,2,-3,-4). The use of node **I** is necessary to guarantee that the destination Mosaic node receives packets of even byte length.

3e. Induced Routing Cycles

Mosaic routing avoids deadlock by preventing cycles from forming in any route. This is accomplished by routing completely in the **X**-direction before routing in the **Y**-direction. A **Y**-to-**X** transformation violates that restriction by explicitly allowing **X**-direction Mosaic routing after **Y**-direction routing, making deadlock possible. Routing deadlock can be procedurally prevented by imposing an ordering on meshes to prevent any route from being used that could form cycles.

We shall experiment with and analyze various interconnection mechanisms. The danger of a deadlock being used as an avenue for a loss-of-service attack may argue against the use of **Y**-to-**X** transformations in a commercial setting. Alternatively, the inclusion of channel deadlock recovery via 'progress' detection hardware may

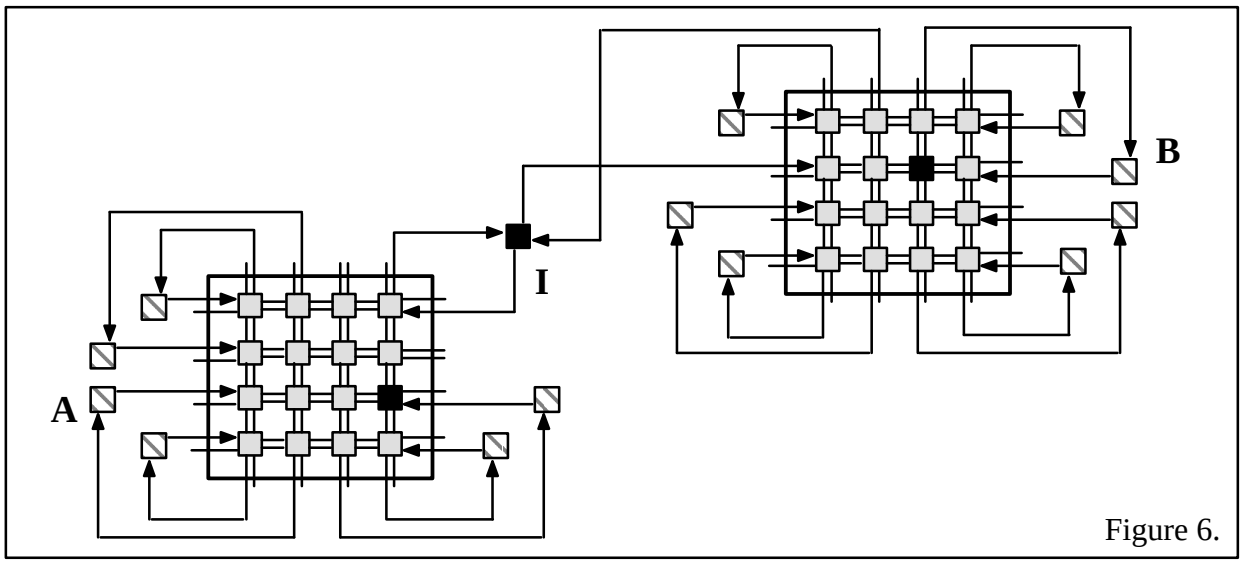


Figure 6.

be practical. This could also be done in software, although response would be much slower. These are topics for future research.

3f. Address Consultant - AC

The ATOMIC LAN is unusual. Hosts do not have a network address in the usual sense. ATOMIC interfaces have no network address since they are addressed relative to a sender's position. Since Mosaic is a point-to-point technology and does not implement broadcasting, the mechanisms for determining the route between source and destination cannot rely upon broadcast. A facility such as the ARP protocol in the Ethernet will not work, although the ATOMIC layer could implement a broadcast facility. This raises the important issue of how to map hosts to source routes.

Assume that at least one host process, called an Address Consultant (AC) resides somewhere on the network. This AC learns the LAN's topology and the location of each host on it. This topology information can be determined dynamically by each AC. In combination with a limited amount of network layer support added to the Mosaic nodes and host interface drivers, it becomes possible for each host to interrogate an AC for the source route to a particular destination. AC software is already in use in the prototype LAN.

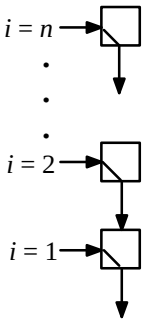
In the current prototype LAN, an AC daemon runs on each host that is connected to ATOMIC. As ACs encounter each other, the lower ranking AC in each encounter goes into a dormant state, only processing address resolution requests from the local host. If the highest ranking AC's host should crash, or the network should become partitioned, the dormant ACs remap the network, establishing new highest ranking AC(s) as necessary.

Returning to Figure 4, the sets of shortest paths between **A⇒B** and **B⇒A** have four elements. The set for **C⇒D** has two elements. The multiplicity of paths between nodes can be used by the AC for the purposes of bandwidth reservation, service guarantees or failure recovery.

The dynamic programmability of nodes allows an AC to command an interface to transmit packets at a controlled rate. It can also dynamically redefine each host's routing table. Traffic loads can be monitored and a new route assigned to avoid congestion, an old route redirected to decrease congestion, or a set of routes redirected to make available a new path of sufficient capacity. There are many possibilities to be investigated.

3g. ATOMIC vs. Switch-Based LANs

On the surface, an ATOMIC mesh router may resemble a switch. It bears some similarity to a space division crossbar switch with a self-routing property [13]. But unlike traditional switch elements, each Mosaic-C node contains its own CPU and program store. Each node can buffer and perform a function on the messages that it receives. A Mosaic-C node router is symmetric in two dimensions and independent from the processor. It may transmit or receive in any of four external directions simultaneously and may do that while node computation continues without interruption.



Assume that all messages are of equal length. If a crossbar operates synchronously, it assigns a fixed priority to each of its $i = 1,,n$ inputs. The input channel closest to the output has the highest priority. The first-come-first-served arbiter in Mosaic routers ensures that when contention occurs messages are forwarded in their order of arrival hop-by-hop. This assigns a geometrically decreasing share of the output channel to each input i, with input i's share equal to $1/2^i$. Under fully loaded conditions, the arbiter in the router ensures local but not global fairness. However, no input will be permanently blocked as it can be when priorities are fixed. This example is simplified. A crossbar element has two inputs, while Mosaic-C routers have five inputs.

Mesh Routing Flexibility

A dual-connected mesh can behave much like a crossbar with no internal buffering. This produces head-of-line (HOL) blocking, where the horizontal mesh input channels vie for access to vertical output channels. With no use made of mesh internal buffering, if k packets contend for the same output channels, $k-1$ input channels will remain blocked and so $k-1$ output channels remain idle. There may exist packets destined for one of the idle output channels that are queued behind blocked packets.

Mesh flexibility can be used in conjunction with the AC to alter gross mesh routing behavior. By making use of the storage within the mesh, HOL blocking can be avoided. This occurs when the AC assigns composite routes that store all input packets at the nodes on vertical output channels from where they are forwarded.

This can be done dynamically in response to changing traffic characteristics. It can be done for some, but not all input channels. It can be done for some, but not all output channels. Flow controls can also be programmed into the mesh. The geometrically decreasing share of an output channel discussed above results in unfairness under heavily loaded conditions. Nodes can be programmed to ensure that channel access is more fairly distributed.

Some Comparisons

In Table 1 the performance of an ATOMIC switch is compared to both the Autonet switch and Nectar HUB [15][12]. The advantage of distributed routing used in an 8×8 mesh is pointed out by comparing mesh performance to that of an Autonet switch. A mesh contains 64 routers, each independent, while an Autonet switch

Table 1.
Switching Performance for 80 Byte Packets

	pkts/sec	latency	channel bandwidth	aggregate switch bandwidth
Autonet	2.0 M	480 ns	100 Mb/s	1.3 Gb/s
Nectar	2.5 M	700 ns	100 Mb/s	1.6 Gb/s
ATOMIC	12.5 M	200 ns	500 Mb/s	16.0 Gb/s

contains one router. An 8×8 dual-connected mesh is externally similar to a 16×16 Nectar HUB crossbar.

ATM-Based LANs

Asynchronous Transfer Mode (ATM) has been suggested as an implementation technology for local as well as wide-area networks [17][23]. ATM messages are of fixed length. Each is 53 bytes long. with five bytes reserved for header, leaving a 48 byte payload. Several teams are creating prototype ATM host interfaces [16] [18][19]. The Autonet follow-on, AN2, will have ATM switches [20].

ATOMIC sends variable length packets and it uses the distributed computational and routing capability of a mesh. That sets ATOMIC well apart from ATM-based LANs. The fragmentation and reassembly required when ATM carries higher layer traffic are not required in ATOMIC. That is one reason why ATOMIC host interfaces are small, inexpensive and fast. However, nodes could be programmed to implement the AAL (ATM Adaptation Layer) for IP and associate source routes with circuit identifiers if that was desired.

One additional point should be stressed. The low cost of nodes coupled with their programmability makes it practical to utilize them in workstation architecture [21]. The network is then extended *into* the workstation. This allows internal devices to have their own independent Gb/s interconnect in addition to direct network access. It has also been suggested that ATM could be used for this, although switching would be external [22][23].

Questions of Mesh Geometry and Size

Although strict communications symmetry is a concern for those who design telecommunications switches, asymmetric designs might be better suited to the needs of a LAN. In most LANs today there exist key hosts, such as file servers, bridges and gateways, that produce and consume the majority of network traffic. It may be possible to take advantage of this to construct mesh geometries that provide *favored* attachment positions to some hosts but not others, whereby traffic to/from favored hosts can proceed without ATOMIC store-and-forward routing.

If all remote nodes are dual-connected to a mesh, the mesh must take on an $n \times n$ geometry, since each incoming **X**-channel must be matched by its own outgoing **Y**-channel. If remote nodes are not dual-connected to a mesh, the geometry need not be square.

The size of a mesh has a bearing on the number of hosts that can be attached to it and properly serviced. A large ATOMIC LAN is constructed by reserving channels for interconnection to other meshes. How large should meshes be so that a large LAN composed of interconnected meshes operates efficiently, with little inter-mesh blockage or congestion? Can mesh geometry be exploited to provide favored channels for mesh-to-mesh traffic?

Each of the issues touched upon above are interesting. Each can be investigated by performing extensive simulations and will be subjects for future research.

3h. Optical Cable Interface Cost

This brings us to a major obstacle: the absence of reasonably priced Gb/s cable transceivers. The Fibre Channel and SONET are representative of standards for high-rate transmission of data over long distances, whereas in LAN application what is needed is high-rate transmission over relatively short distances. Prototype fiber-optic LAN links will be expensive. Although volume manufacture will reduce price, the Fibre Channel may be inappropriate for cost effective installation in point-to-point based LANs. Fiber Channel has a complex 300 page standards specification and is not designed primarily for LAN application [14]. It transmits data in a strictly framed and encoded format for distances of up to 10 km.

FDDI provides a demonstration of what happens when host interface costs are too high. For Gb/s LANs to become practical, we judge that the cost per host should not exceed $2,000 in current U.S. dollars. The $2,000 target may be difficult to reach. Although an ATOMIC host interface is very inexpensive, as is its share of the mesh interconnect, the current cost for fiber-optic cable and transceiver electronics pushes the total cost well over the target figure and will grossly dominate total interface cost.

By way of contrast, interfaces that use ribbon cables with SLACK repeaters are inexpensive. If distances between nodes are typically several meters or less, e.g. within a host, the issue of optical interface cost is unimportant. Ribbon cables suffice and are much less costly. Transmission via bit-serial copper media, either shielded twisted-pair or coaxial, is impractical at gigabit rates for the distances typically found in LANs [24].

Some fiber-optic researchers believe that substantial cost reductions could be made when the application domain for gigabit fiber-optic transceivers is limited to asynchronous point-to-point communication at distances no greater than 150 meters. We feel that this is an area where more industry effort should be focused.

4. Preliminary Testbed Experiments

In late October 1991 the first ATOMIC LAN testbed was created at ISI. The prototype consisted initially of two Sun-3 model workstations. Each workstation had a VME bus to which an interface board was attached. A BSD UNIX device driver and network interface driver were written, providing complete TCP/IP compatibility. They are attached via an Ethernet gateway to the Internet.

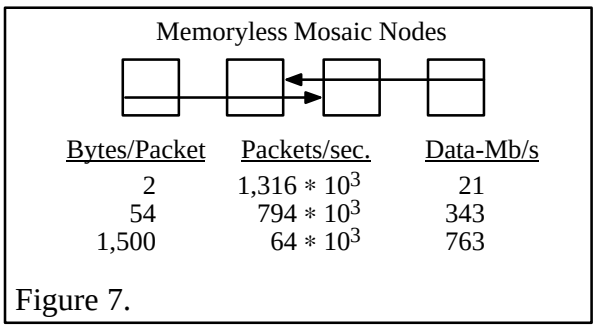
4a. Intra-Host and Inter-Host Transfer Rates

Empirical measurements on Sun-3 workstations show performance bounds that reflect the Sun's own software and VME bus limitations. Recently, substantial efforts have improved BSD UNIX kernel memory management and packet processing, but that still does not allow the testing of ATOMIC interfaces at anywhere near their capacity. Therefore, traffic generation and testing software were placed directly into the nodes themselves on the network side of the VME bus.

A packet size of 1,500 bytes is typical of the maximum size used for TCP packets carried over an Ethernet, while 54 bytes is the approximate length of a 53-byte ATM (Asynchronous Transfer Mode) cell. It is also about typical of Telnet interactive packets. These two sizes were used for performance characterization in addition to the physically minimal packet size of two bytes, that allows accurate determination of per-packet overhead.[3]

The full-duplex data capacity of channels between Memoryless Mosaic nodes is shown in Figure 7. Link-layer overhead was removed to accurately portray channel data carrying capacity. Packets were sent from the source node's shared memory, over channels and back into the shared memory of the destination node.

Each node was provided with 128 KBytes of 25 MHz SRAM organized as 16-bit words. This limits simplex channel bandwidth in these particular interfaces to 400 Mb/s, because data can neither be sourced nor sinked faster than that rate. The channel can operate at 500 Mb/s if 35 MHz SRAM is used.



Bytes/Packet	Packets/sec.	Data-Mb/s
2	$1{,}316 * 10^3$	21
54	$794 * 10^3$	343
1,500	$64 * 10^3$	763

Figure 7.

The per-packet interrupt service overhead limits each node in an interface to approximately 650,000 pkt/s or 1.5 μs/pkt. Memory bandwidth limits performance for longer packets. For 1,500 byte packets, 95% of the 400 Mb/s limit was achieved. Since a 32-bit VME bus is limited to bandwidth of no more than 320 Mb/s, it is clear that ATOMIC can transport data between subsystems and peripherals at rates at least comparable to that of buses.

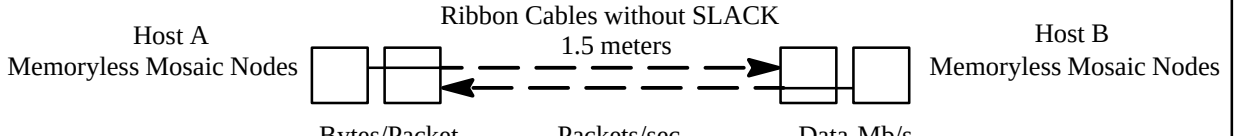
4b. Host Interface Performance

Prototype ATOMIC host interfaces contain two Memoryless Mosaic nodes. Each node operates independently, so that performance for duplex communication is limited to 800 Mb/s. Figure 8 demonstrates the rates achievable for a duplex connection between hosts. As before, per-packet overhead limits performance for small packets, but now cable induced delay limits channel bandwidth for long packets. With the incorporation of SLACK function, channel bandwidth for long as well as short cables should remain near that shown in Figure 7.

Near-Term Node Enhancement

The SRAM in current host interfaces limits network channel performance to well below the rate at which a node router can operate. A channel can only carry data as fast as it is sourced or sinked and both are limited to 400 Mb/s. Using 16-bit words, to saturate one simplex channel requires 40 MHz SRAMs. Memoryless Mosaic chips that operate at 38 MHz have already been fabricated.

[3] Measurements taken on Ethernet LANs populated primarily by Sun Microsystems workstations show that NFS (Network File System) and ND (Network Disk) protocols represent approximately 80% of traffic. Packets less than 50 bytes in length represent 30% of traffic. Approximately 10% of traffic was NFS remote procedure call, with packets of length between 124 and 168 bytes [25]. As LANs rely more upon request/response protocols this class may become dominant as suggested by recent measurements. [26].



Bytes/Packet	Packets/sec.	Data-Mb/s
2	$1{,}316 * 10^3$	21
54	$794 * 10^3$	343
1,500	$47 * 10^3$	566

Figure 8.

Thus a 50% improvement in statistics presented above is available by using the more recent chips.

Mosaic technology is designed using scalable design rules. As CMOS fabrication moves from 1.2 μ to 0.8 μ feature sizes, both clock rates and router speed can be expected to improve by at least 50%. Rates of 960 Mb/s are attainable without widening the 8-bit data path that Mosaic channels use today. Widening the channel to 16-bits allows that rate to double. Sixteen bit channels are already used by the Intel Touchstone MRC (Mesh Router Chip, itself based upon the Mosaic router).

4c. Host Limitations

Creating a gigabit workstation architecture is a challenge. Workstation, protocol, and operating system architectures are currently inadequate for gigabit communication. This is recognized and has been the subject of extensive study [5][17][29][30][31][32].

To pick an example, most buses operate well under gigabit rates. It is unrealistic to expect VME-based designs, that are limited to 32-bits at 10 Mxfer/s (transfers per second), to be capable of handling 0.5 Gb/s. A 64-bit Futurebus+ at 25 Mxfer/s is capable of Gb/s rates, but the cost of interfacing to it is very high.

We feel that that problem can be inexpensively solved by incorporating nodes into the workstation's major peripheral subsystems [21]. This brings multiple interconnected Gb/s channels into the architecture inexpensively, provides direct network access to subsystems and allows parallel formatting and protocol processing.

At gigabit rates unnecessary data copying must be avoided. Packet copying often occurs at the interface between the operating system and the network. This problem is solved by allowing the network device to address its host's memory. The processor in a node is capable of manipulating shared data structures while the DMA packet interface could read from or store into them.

Memoryless Mosaic nodes, that are currently limited to 128 KBytes, can address a larger segmented memory space. By making the dual-ported SRAM larger, the buffers associated with network IO could be treated as a part of the operating system buffer pool. This design is similar in spirit to a Nectar CAB [33].

Between the application and the operating system kernel, it is common for a packet data copy to take place. No consensus has been reached on solving this problem. Breaking down the boundary between the application and the protocol stack is one likely solution [34].

5. Closing Comments

Message-based multicomputer nodes have become practical to use outside their originally intended domain. They provide an effective and very general mechanism for the distribution and routing of data at Gb/s rates. Mosaic multicomputer nodes incorporate routers and use point-to-point communications channels. This provides them a linear scaling property not shared by one-at-a-time communications technologies, such as buses, Ethernet, FDDI and DQDB.

Mosaic nodes can be used to construct Gb/s LANs. Nodes can be used to create programmable routers and host interfaces. The buffering and processing power within each node provides design flexibility not normally associated with communication subsystems. This can be exploited to dynamically alter the behavior of an entire communications system or a portion of it.

The ATOMIC LAN is an operating example of a Gb/s testbed constructed from Mosaic message-based multicomputer nodes. In addition to its purpose as a demonstration vehicle, the presence of a gigabit LAN provides a laboratory in which to test proposed solutions to problems that arise with greatly increased network bandwidth and to develop new applications and devices for the emerging gigabit domain.

6. References

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

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

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

[4] Seitz, C. L.
Multicomputers
Chapter 5 in *Developments in Concurreny and Communication*
Ed. Hoare, C. A. R.
Addison-Wesley Publishing Co., 1990.

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

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

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

[8] 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.

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

[10] Mead, C., Conway, L.
Introduction to VLSI Systems
Addison–Wesley, 1980.

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

[12] 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.

[13] 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.

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

[15] 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.

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

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

[18] 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.

[19] 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.

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

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

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

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

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

[25] 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.

[26] 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.

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

[28] 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*.

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

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

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

[32] 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.*

[33] 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.

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