ATOMIC:
A Low-Cost, Very-High-Speed LAN

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

Abstract

ATOMIC is an inexpensive O(gigabit) speed LAN built by USC/ISI. It is based upon Mosaic technology developed for fine-grain, message-passing, massively parallel computation. Each Mosaic processor is capable of routing variable length packets, while providing added value through simultaneous computing and buffering. ATOMIC adds a general routing capability to the native Mosaic wormhole routing through store-and-forward. ATOMIC scales linearly, with a small interface cost. Each ATOMIC channel has a data carrying capacity of 500Mb/s. A prototype ATOMIC LAN has been constructed along with host interfaces and software that provides full TCP/IP compatibility. Using ATOMIC, 1,500 byte packets have been exchanged between hosts at an aggregate transfer rate of more than 1Gb/s. Other tests have demonstrated throughput of 5.25 million packets per second over a single channel. This paper describes the architecture and performance of ATOMIC.

Keywords: local area network, gigabit, source routing, high-speed, performance, Mosaic.

1.0 Overview

ATOMIC is an O(gigabit) speed, low cost, switch-based LAN built by USC/ISI. It relies on the repetitive application of Caltech's Mosaic chip (see Section 3.1), that serves as a fast and smart switching element. It is capable of routing variable length IP packets while providing added value through simultaneous computing and buffering. ATOMIC scales easily by adding point to point channels and has a low interface cost.

At present, a prototype of ATOMIC is operational including an IP interface for the BSD UNIX[2] environment. The developmental Host Interface (HI) boards use four memoryless Mosaic chips to create four network interfaces. Each interface uses 25MHz SRAM memory chips that are accessible to the host Sun-3 workstation via the VME bus. The use of 25Mhz, rather than 32Mhz, mem-

---

1. An earlier version of this paper appeared as ISI Technical Report ISI/RR-92-291. DARPA supports ATOMIC through Ft. Huachuca contract No. DABT63-91-0001 with USC/ISI. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

2. UNIX is a registered trademark of ATT.

ory restricts ATOMIC to 80% of its potential performance (as verified by other tests). Using these slow HI boards we were still able to obtain the following performance: a single interface acting as a source can send 381 megabits per second (Mb/s) if 1,500 byte packets are used. Using longer packets allows us to approach 400Mb/s, the limit of the 25Mhz memory. A packet rate as high as 5.25 million packets per second (Mpkt/s) has been achieved over a single channel. The bit error rate of ATOMIC channels is still unknown. Transfers of test patterns both intra-board and inter-host of over 1,000 Terabits ($1{,}000 \times 10^{12}$) resulted in neither bit errors nor lost packets.

ATOMIC is a switch-based local area network and is constructed using HIs and switches ("concentrators"), each of which may be connected to HIs or to other switches. Each switch is a perfect crossbar and every port is bidirectional with two independent channels, in and out. HIs have two such ports. If traffic analysis so suggests, multiple ports may be used between switches. Multiple hosts may be connected to one another without employing switches by using any strongly-connected topology (e.g., a ring). Parallel link connections and loops are allowed.

1.1 ATOMIC Attributes

ATOMIC possesses attributes not commonly seen in current LANs.

- Hosts do not have absolute addresses. Packets are source routed, relative to senders' positions. At least one host process in an ATOMIC LAN is an Address Consultant (AC). It "knows" the LAN's topology and can provide a source route to all the hosts on that LAN by mapping IP addresses to source routes. The network topology can be determined dynamically by the AC.
- ATOMIC consists of multiple interconnected clusters of hosts. This is unlike Ethernet and FDDI in theory, but similar to them in practice as Ethernets are often interconnected via gateways and routers.
- In general, there may be many alternate routes between a source and a destination. The rich routing topology may be exploited by an AC to provide bandwidth guarantees or to minimize switch congestion for high bandwidth flows, such as for video.
- Aggregate performance of the entire network is limited only by its configuration, since ATOMIC traffic flows do not interfere with each other unless they share links. This is unlike bus and ring LANs, such as Ethernet and FDDI, where all the traffic flows compete with each other for the same total bandwidth (of 10 and 100Mb/s respectively).
- Each Mosaic processor is a powerful general purpose computer, thus allowing the network itself to perform complex functions such as encryption or protocol conversion.

ATOMIC is an early example of a Very-High-Speed LAN (VHSLAN) that is characterized by transfer rates in excess of a gigabit per second. ATOMIC has a low interface cost and is capable of hosting a broad range of digital video, multimedia and large-scale, high data-rate distributed applications.

2.0 Introduction

Workstations and associated applications have begun to overload traditional local area networks. Ethernet and token ring have proven to be extremely capable networking technologies for interconnecting relatively slow machines using file transfers, electronic mail and remote procedure

calls. 100 MIPs workstations are just over the horizon and a single such processor could overload the capacity of existing LANs. These faster workstations, coupled with high bandwidth applications such as multimedia editors, desktop video conferencing and virtual reality, will be hamstrung if interconnected using current LANs. FDDI only presents a near term solution with its 100Mb/s shared bandwidth limitation.

ATOMIC is an effort to create an extensible LAN technology to address the local area networking needs of the future. Rather than settle for an incremental improvement in current designs, ATOMIC leverages recent advances in parallel computing technology to achieve several orders of magnitude performance improvement over current local area networks.

We believe ATOMIC is unique in its architecture, performance capability and potential for low-cost deployment. The remainder of the paper discusses the design and performance of the prototype ATOMIC network that has been operational at USC/Information Sciences Institute since October 1991. We postpone a discussion of related work until Section 10.0.

3.0 Architecture Overview

3.1 Mosaic

The ATOMIC project uses Caltech's Mosaic family of multicomputer components to construct a VHSLAN (see Figure 1). Each Mosaic chip contains a general purpose processor, RAM, ROM, a

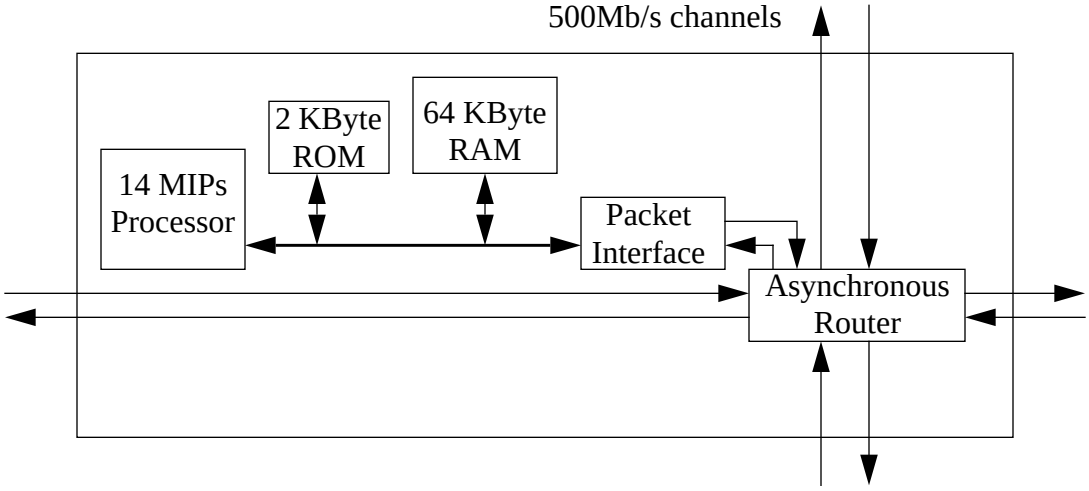


FIGURE 1. Mosaic Processor Chip

DMA channel interface and self-timed routing hardware that supports a two-dimensional source-routing topology. Eight simplex channels are supported at a nominal rate of 500Mb/s each. A full-duplex (FDX) point-to-point host link constructed from a pair of these channels provides 1 Gigabit per second (Gb/s) of data transfer capacity. There is also a memoryless version of the Mosaic chip that supports only 5 channels.

Mosaic chips route variable-length messages via a limited form of two-dimensional hop-by-hop source routing. Each message contains a source route prefix: a delta-X (ΔX)-byte followed by a

delta-Y (ΔY)-byte. Each byte can take on a value from -127 to +127 hops, with the sign controlling the West/East direction for the ΔX-byte or the South/North direction for the ΔY-byte. As a packet passes through Mosaic nodes, its leading prefix is decremented until it reaches zero. X-direction routing, if any, occurs before all Y-direction routing. This restriction avoids deadlock [10].

The Mosaic chip is the building block for both the HI boards and the switches. Other chip types are needed only for the long distance transmission necessary for construction of a practical LAN.

3.2 ATOMIC

The Mosaic chips provide an excellent base from which to create a LAN, but alone, they do not provide a sufficiently general network. Mosaic was designed for Caltech's 128-by-128 mesh super computer. In a rectangular mesh, the limited X-Y routing is sufficient for each node to reach any other. In an ATOMIC LAN, the interconnection geometry will not be that regular. It will consist of smaller meshes attached to one another and to hosts. Often, a single X to Y transition will be insufficient. A more general routing mechanism is necessary and is described in Section 5.2.

Host interfaces and cables also must be developed to create a LAN. Finally, an address resolution mechanism for converting between IP addresses (host names) and source routes needs to be in place. Therefore, ATOMIC requires several enhancements to Mosaic to construct a viable LAN.

3.2.1 Interfaces

The first requirement for building a functioning LAN is the addition of host interface hardware and software. Caltech has designed and built the Program Development/Host Interface Board (HI) to prototype Mosaic software. This board is capable of acting as an interface for the ATOMIC LAN. Section 4.1 describes the board in detail. To seamlessly integrate the hardware with the operating system, a BSD UNIX interface and a device driver were written for use in Sun-3 workstations (see Section 5.1). ATOMIC currently supports the IP protocol and therefore all the communication protocols above it, such as UDP, TCP, ICMP, TELNET, FTP and SMTP.

3.2.2 Extended Routing

We have created a network-layer protocol (ATOMIC) that runs in the Mosaic processors to provide more general routing by using a store-and-forward capability (see Section 5.2). ATOMIC source routes consist of multiple pairs of X-Y Mosaic routes such as (X1,Y1)(X2,Y2). Mosaic's native wormhole routing is used for each pair, and ATOMIC forwarding between the pairs.

In addition to allowing more general network topologies, ATOMIC routing can exploit multiple routes between hosts.

3.2.3 Switches

To provide an interconnection fabric between hosts, 8-by-8 meshes of Mosaic nodes will be interconnected in a multi-star configuration. Each 8-by-8 mesh is a single board approximately 8 inches square. These meshes will be inexpensive (approximate cost $5000), because they are being mass-produced for Caltech's Mosaic supercomputer. Section 4.2 describes the mesh and LAN configuration in more detail.

3.2.4 Cables

Long cables are necessary for connecting hosts to switches. Gigabit-speed fiber optic interfaces practical for use in low-cost LANs have not yet been developed. Section 9.0 discusses the issues involved in creating cables for ATOMIC.

3.2.5 Address Resolution

There are no absolute addresses in the ATOMIC network. All packets are source routed from a particular sender to a particular receiver. In order for a sender to know the route along which to send a message to a specific destination (known to it only by its IP address), we have created an "Address Consultant" (AC) user process to map the network and to provide routes between hosts. As the AC maps the network and discovers hosts, it provides each host with a host→AC route. When host A needs to send a packet to host B, it first sends a request to the AC. The AC returns an A→B source route to A. Then host A is able to send packets directly to host B without the intervention of the AC. These routes are cached locally at each host. Section 6.0 provides more details on the operation of the AC.

4.0 Network Hardware

4.1 Prototype Host Interface Board

As mentioned above, Caltech created the HI board that attaches to a VME bus to prototype Mosaic software. The ATOMIC project utilizes HI boards for its initial host interfaces to demonstrate the capabilities of the LAN, its underlying point-to-point Mosaic technology, and new architectural and communication software innovations.

A prototype ATOMIC host interface is small, composed of four 32KByte memory chips, the memoryless Mosaic chip, plus clock and bus interface logic chips. Each memoryless Mosaic chip contains a 14 MIPs processor and support for five Mosaic channels. The HI board provides each Mosaic chip with external, shared memory that is available to both the Mosaic chip and the VME bus. The memory chips impose a performance ceiling whenever a chip is either the source or the destination for a packet.

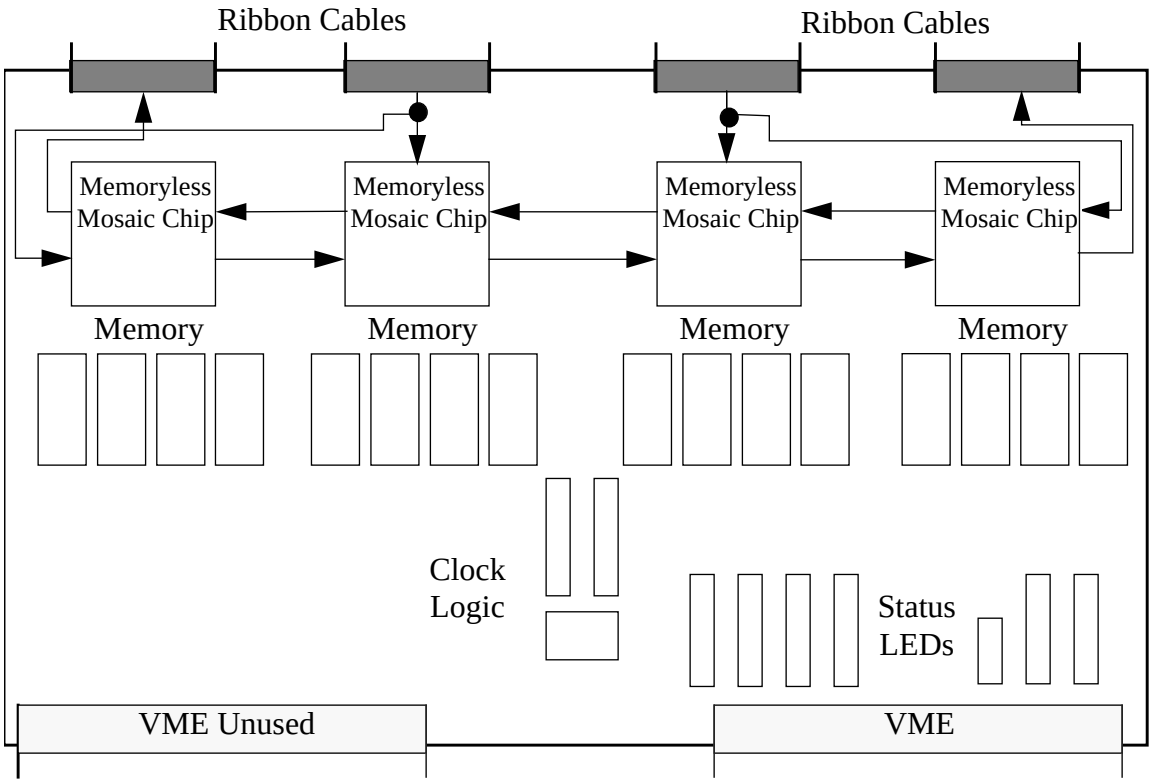


FIGURE 2. Program Development/Host Interface Board

Each HI board contains four complete ATOMIC host interfaces and 4 external channels, 2 in and 2 out. Each host requires at least one interface, though two are preferable, since one may be dedicated for network input and one for network output, doubling potential throughput and providing parallel processing and management of network traffic. The HI board is depicted in Figure 2.

The HI board demonstrates another attribute of this networking technology. Interfaces may be directly connected to one another, as are the four on an HI board. Unless Mosaic-to-Mosaic distance exceeds 2 feet, no external logic is required to interconnect ATOMIC interfaces. Beyond this distance additional logic is needed in order to achieve full channel performance. Without such logic the performance degrades as reported in Section 8.0. It is practical to have multiple interfaces within a workstation, individually associated with high-speed peripherals or processors. This makes it possible for data to be routed to and from these interfaces at gigabit rates independently of buses or backplanes [15].

4.2 Mesh Routers

Arrays of Mosaic chips can be interconnected with one another in a two dimensional mesh. In an ATOMIC LAN, meshes of Mosaic chips function as crossbar switches. A typical mesh board will contain 64 Mosaic chips organized in an 8-by-8 matrix (for illustration 4-by-4 mesh boards are shown in Figure 3). 8-by-8 meshes are being mass produced at low cost using Tape Automated Bonding on Multi-Chip Module packaging technology for Caltech's Mosaic computer of 128-by-128 nodes. An *n*-by-*n* mesh has 4*n* full-duplex Mosaic channel pairs available at its edges. Each channel pair may be used to connect one or a chain of hosts to the mesh. They may also be used to

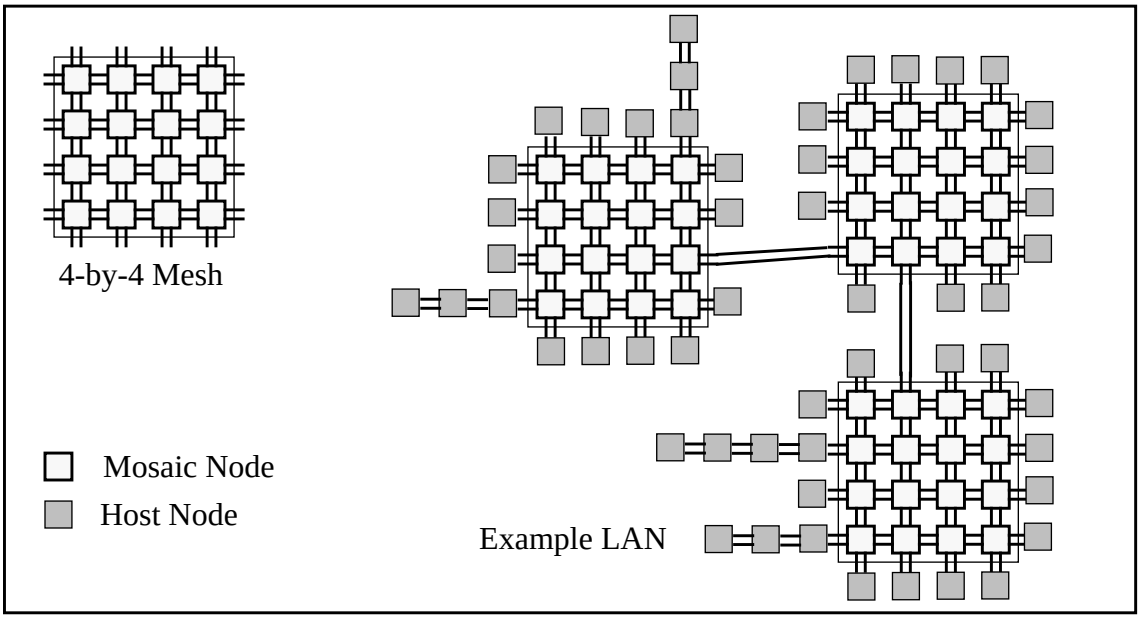


FIGURE 3. Mesh Board and LAN Configuration

connect to other switches. At present, we are using a 3x3 version while the 8x8 meshes are being debugged by Caltech.

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*-by-*n* switch, this is 2*n* channels in either direction. Assume that each Mosaic chip in a mesh continuously sends packets to other chips in the mesh with destination addresses chosen randomly. Simulation studies performed at Caltech have shown that under those conditions the achievable throughput across a bisection is approximately 50% of its theoretical maximum capacity. Since Mosaic channels provide 0.5Gb/s of capacity, a bisection provides 0.5*n*Gb/s of capacity in either direction [29].

An *n*-by-*n* ATOMIC switch with hosts attached at its edges will normally deal with much more regular traffic routing patterns than were assumed for the above simulations. An *n*-by-*n* ATOMIC switch uses both X and Y directions to transmit packets. An 8-by-8 mesh should therefore have a maximum theoretic capacity of 16Gb/s, and 50% of that would be 8Gb/s.

It is possible to use connections of host to crossbar such that the crossbars are perfect, meaning that conflict will only occur if two sources try to send to the same destination. This is unlike more limited switching networks (e.g. Butterfly). An 8-by-8 mesh can support 16 hosts in “crossbar” mode or 32 hosts with blocking.

5.0 Network Software

5.1 UNIX/Sun OS Modifications

The current prototype of ATOMIC is implemented for Sun-3 workstations running a BSD UNIX operating system. The ATOMIC LAN driver is implemented in the BSD kernel and interfaces to the standard BSD socket mechanism. This provides IP level access to the ATOMIC network,

allowing higher level protocols, such as UDP, TCP and ICMP, and higher level network services, such as telnet, file transfer and electronic mail, to be run transparently.

Autoconfiguration:

Autoconfiguration is a BSD mechanism by which a subset of all possible devices, the devices that are actually present, are configured at the time that a workstation is booted. During the booting process, the shared memory mapping, the mapping of Sun memory to PD/HI memory including the VME location, the priority of interrupts, and the name of the interrupt routine is established. Standard kernel entry points are specified in the driver structure. Configuration of the ATOMIC interface associates an Internet address and a network mask with the interface. This allows us to use standard BSD "routing" routines to send IP packets out through the appropriate interface. The IP address of the ATOMIC interface is then associated internally with the ATOMIC loopback address (0,0). A packet that is passed to the router addressed to (0,0) will be returned to the sending processor with the first two bytes stripped off.

Interrupt Linkage:

Since the Host Interface board and network are so much faster than the Sun, flow control is a major issue. In order to maximize the number of packets received by the user process, while minimizing the number of packets discarded by the Host Interface and at the IP input queue, the queue sizes and the rate at which the HI processor interrupts the processor on the Sun have to be carefully tuned. BSD internal queue sizes (e.g., between the host interface and IP) have in the past been optimized for Ethernet interfaces that are much slower than the ATOMIC interface. Our near-term solution is to adjust buffer sizes to maximize performance. We expect in the future to implement some sort of flow control mechanism.

Address Resolution:

In the "address resolution" function an Internet address is mapped to a corresponding link level address. For Ethernet, the link level address consists of six octets that are unique. For ATOMIC, the link level address is a variable length source route that describes the location of a node relative to the source. In the BSD implementation of IP over the Ethernet, the Address Resolution Protocol (ARP) is implemented entirely within the kernel. For ATOMIC, the expectation is that the Address Consultant (AC) will be comparatively complex and may need to be changed more often than the operating system. In addition, it may be desirable to have different versions of the AC running on different hosts or even on the same host. Placing the AC functionality within the kernel would severely restrict the above possibilities. Therefore, the AC is implemented as a user process. This creates a different set of complications.

Raw Link Level Addressing:

The AC determines the network topology by attempting to loop packets through the network on various source routed paths. To do this, the AC must be able to specify the ATOMIC link layer packet header (source route) in order to communicate with Mosaic nodes on the network, within a switch or within a host interface board, that may not be connected to an Internet host. In the mapping process the AC receives packets previously sent, as they complete the loop. More than one user program that sends packets with the link layer header specified (e.g. an AC or a network con-

trol program) may be run on the same host, at the same time. Therefore, there must also be a mechanism for distinguishing these “loop” packets that are destined for different processes in the same host.

In a standard BSD system, the lowest level of “raw” addressing previously available has been at the IP level. To implement user specification of the link layer header, we devised an “AF_ATOM” address family that is analogous to the “AF_INET” and is used to open a BSD socket. To make use of this address family the user program opens a User Datagram Protocol (UDP) socket specifying the “AF_ATOM” address family. On each send operation the user program supplies the link layer header at the beginning of the packet. As the packet is processed by the kernel, the packet supplied by the user is encapsulated in a UDP header and an IP header, and the ATOMIC link layer header is copied onto the front of the packet. Once the packet has been sent by the ATOMIC device driver in the kernel, the packet makes its way through the network and the source route is gradually stripped off. Eventually the packet arrives back at the source. When the packet is received, the addressing information in the IP header and the UDP header are used by the kernel to direct the packet to the appropriate user process to be read on an “AF_INET” UDP socket. The operation of the AC routing protocol is diagrammed in Figure 4.

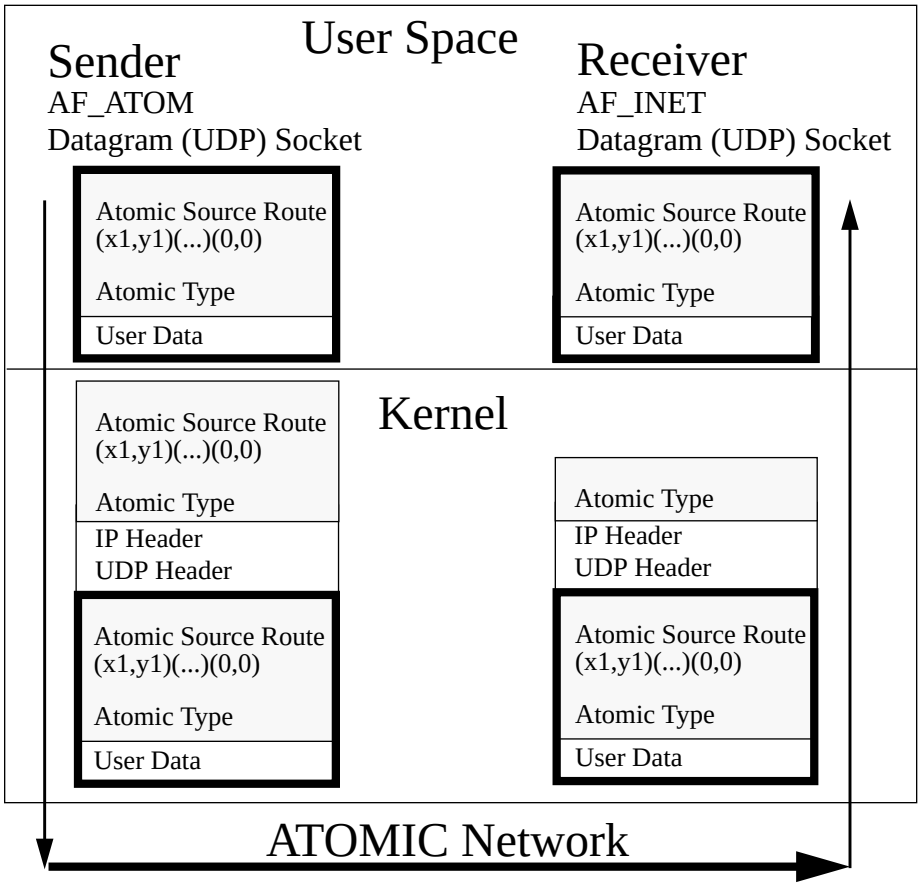


FIGURE 4. AC source routed packet encapsulation.

Input/Output Control Functions:

Communication between a user process and the ATOMIC driver in the kernel is accomplished via “IOCTL” functions. Three IOCTLs (analogous to the Ethernet address loading functions used by ARP) are used to read, to write, and to delete ATOMIC address table entries in the kernel. Additional IOCTLs are used to control the Mosaic nodes on the host interface board including loading programs and data into the Mosaic processors, reading counters and status fields and executing control functions including resetting the HI and enabling/disabling interrupts.

5.2 ATOMIC Network Layer

The ATOMIC network layer extends the native Mosaic wormhole routing and allows more complicated source routes by providing a store-and-forward capability. ATOMIC routes are created by repeated application of Mosaic routes. In Figure 5 suppose host A needed a route to host B. Mosaic routing with only a single X to Y transition would be insufficient to connect A and B. By catenating multiple Mosaic routes together we can create a path (several paths) from A to B. One example path is the highlighted one. We denote this path as (+6,+1)(+4,+3)(+1,0)(0,0).The path from host B to host A travelling along the reverse links would be denoted as: (-1,-3)(-4,-1) (-6,0)(0,0).

ATOMIC network layer software runs in each of the Mosaic nodes of the network. A packet generated at A and destined for B would be prepended with the ATOMIC address listed above. The packet would be injected into the network using Mosaic routing and would only appear at the Mosaic processor labeled T1 after travelling (+6,+1). The ATOMIC network layer software would recognize that the packet needed forwarding because the leading bytes would be (+4,+3). After travelling its next Mosaic route it would appear at the Mosaic processor labeled T2. This

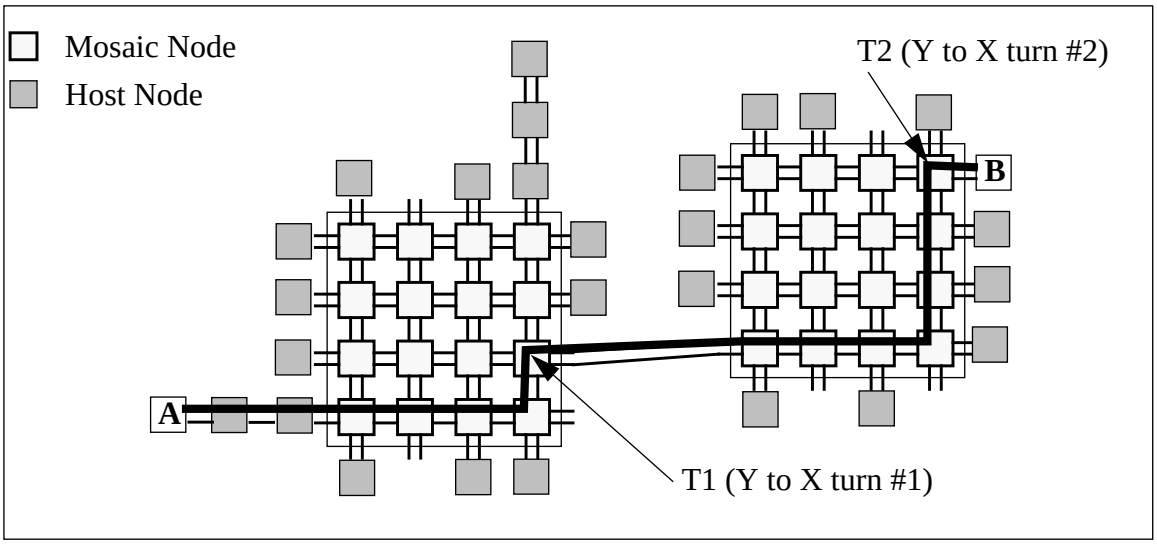


FIGURE 5. Example Network.

processor would forward the packet (+1,0) where it would arrive at Mosaic processor B. This processor would deliver the packet to the host upon detection of the (0,0) at the head of the packet. Note that at each Y to X turn the entire packet is stored in the Mosaic processor before being forwarded along its next path. Native Mosaic routing uses wormhole routing to avoid store-and for-

ward delay. ATOMIC uses native Mosaic routing when it can, but resorts to store-and-forward for complex routes.

6.0 Address Consultant

The Address Consultant is responsible for providing routes between hosts. In traditional LANs, such as Ethernet, each host is associated with a specific (fixed) address, and host IP addresses (or names) are translated into unique hardware addresses. In ATOMIC all packets must be source routed. In order for a host to convert a host name or IP address into a source route, it needs to "know" the topology of the network. It would be inefficient for every Mosaic node or even every host to maintain the topology of the entire LAN. We have chosen instead to designate one (or more) hosts to perform this topology discovery as a user process known as the Address Consultant or AC.

The AC performs two basic functions. The first is to map (and remap) the network so as to maintain a consistent picture of the LAN. Its second function is to use this map to provide routes between hosts upon request. An added benefit of the AC is its ability to balance flows throughout the network. When a source host requests a path to a sink host, the source may also provide some information to the AC about the type of connection to be opened. If the connection were a video stream, the AC would know to return a source route that avoided other connections and to avoid creating new, conflicting paths.

To increase fault tolerance, any host may become an AC if it cannot find one in the network. In fact, every host may be an AC simultaneously without affecting the correct operation of the network. In a large network, it may make sense for multiple ACs to be running in different parts of the network so that requests from hosts need not travel large distances to get to an AC.

6.1 AC Mapping

The AC maps the network by discovering its own neighbors and then recursively discovering neighbors of neighbors. A node is "discovered" if an ATOMIC message returns from it. For example, in a network with bidirectional links, the AC would check for a neighbor on its right by sending a message to (+1,0)(-1,0) (right, then left). If a Mosaic processor exists to the AC's right, then the message will return. The AC then probes left, up and down. If any nodes are discovered, the procedure is recursively repeated from each of the discovered nodes, thus implementing a breadth-first search over the network. The mapping operation requires that all the Mosaic nodes in the network be prepared to forward ATOMIC packets, but the nodes do not need to perform any further processing. The AC, in effect, sends packets to itself by looping through potentially existing Mosaic nodes. If a packet returns, the AC knows that Mosaic nodes exist along the path. If a packet doesn't return, the AC discovers the absence of a node. On hypothetical LAN topologies, it appears that 20 messages per Mosaic node are sufficient to completely map the network. The exact number is topology dependent, and we have not yet derived a tight bound on the number of messages required.

6.1.1 Network Configurations

At present, ATOMIC is running on a network with four hosts. The hosts are connected via a 3-by-3 switch in a *symmetric* and *consistent* configuration. Symmetric implies that each channel has a

matching channel in the reverse direction. Consistent describes a network where channels do not change direction of travel: East output is connected to West input and North output is connected to South input, etc. (see Figure 6).

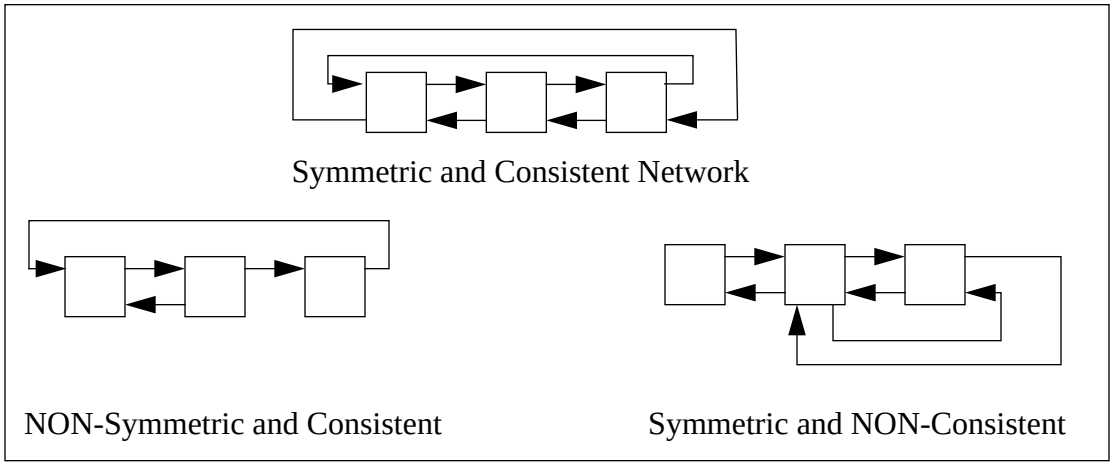


FIGURE 6. Network Configurations.

In a symmetric and consistent network, the AC's network mapping is straightforward. Probe (loop) messages can be constructed by reversing the outgoing path to create a return path. For example, (+2,0)(-2,0) sends a message right two hops, then left two hops. In a symmetric and consistent network, with nodes at (+1,0) and (+2,0) from the AC, the message will return. In a non-symmetric or non-consistent network, no such guarantee is possible and a more complicated algorithm, using "modified flooding", is required. Non-consistent connections are useful to avoid store-and-forward delay, while still allowing general topologies. A connection going out on a Y channel and in on an X channel creates the "Y-to-X" turn that native Mosaic routing lacks. Non-symmetric network connections are useful to improve the performance of the LAN, but they add other mapping complications.

We have implemented an AC for mapping any topology including the non-symmetric and non-consistent networks. The algorithm for non-symmetric nets needs the Mosaic processors to flood mapping messages in a controlled manner through the network and therefore requires more complex code in the network Mosaic nodes.

7.0 Performance

The Sun-3/160 processors, the VME bus, and UNIX socket processing are too slow to determine ATOMIC's performance. In fact, we have not yet been able to see a performance difference at the user level between ATOMIC and Ethernet. Therefore, traffic generation and monitoring code were installed in the memory that is shared by the Sun and the Mosaic processors, thus avoiding the Sun performance limitations. Results reported below are ATOMIC network measurements in the absence of UNIX software overhead.

When 1,500 byte packets (typical FTP packet size) are transferred across a Mosaic channel, a single interface limits the flow to 381Mb/s. Longer packets approach 400Mb/s which is the limit of the HI's 25MHz memory. Similarly, the flow of 54 byte packets (such as ATM) is limited to

171Mb/s. The limit for the shortest packets (4 bytes) is a rate in excess of 657 thousand packets per second (Kpkt/s). Note, these are interface (not Mosaic channel) limitations.

When two flows compete for the same Mosaic channel, the measured performance is 405Mb/s for 500 and 1,500 byte packets, 343Mb/s for 54 byte packets, and 1.3Mpkt/s for the shortest packets. The latter figure is still interface limited and is not even close to the channel capability. With six flows competing over the same Mosaic channel on a 15 foot cable, 3.9Mpkt/s for 2 byte packets was measured. The data are summarized in the tables below. Mosaic chips, in a different test setting, with faster memory chips, have demonstrated transfers in excess of 800Mb/s over a single channel.

Byte/pkt	Kpkt/s	Mb/s
4	657	21
54	396	171
1,500	31	381

TABLE 1. Single flow measured performance.

Byte/pkt	Kpkt/s	Mb/s	Flows
2	5,250	84	8
54	793	343	2
1,500	33	405	2

TABLE 2. Multiple flows competing for a single channel.

7.1 Errors

Mosaic channels use no error detection or correction information and were reported to be extremely reliable, but this was still a cause of some concern. Therefore, we ran a series of data transfer validation tests over the Mosaic channels to characterize their error rates. The Mosaic channel data were transmitted via on-board plated traces and between hosts by ribbon cables.

Checkerboard data patterns were sent from Mosaic nodes and verified upon reception. Typically, test sources sent packets over a multi-hop path. For short distance transmission via ribbon cables (up to 30’) and chip-to-chip on-board transmission, no bit errors were detected in the transmission and reception of more than 1,000 trillion bits ($1{,}000 \times 10^{12}$). Furthermore, no packets were lost. That suggests that neither CRC nor parity are needed for Mosaic channel transmission. This level of fidelity should not necessarily be expected for longer distance transmission over differing physical media.

8.0 Analysis Of Performance

There are several major bottlenecks that limit the performance of ATOMIC:

1. Memory speed
2. Processing (interrupt processing, etc.)
3. Communication links (Mosaic channels, cables)

4. Store-and-Forward delay

In fact, the HI memory chips limit both the storage and the processing time since CPU instructions are read from memory, too. (1) and (2) limit the performance of HIs, and (3) is a channel limit. Under different ATOMIC network architectures the effects of the bottlenecks may change.

In the ATOMIC HI, the time required to transfer packets includes a processing period (to set up pointers and to initiate the send and/or the receive processes), the time for each byte to be retrieved from (or stored in) memory, and the time to be transmitted over the channel. Over short distances and for short packets, this period is dominated by the per-packet processing time. For long packets it is dominated by the memory access time (for retrieval or storage). When transmitting over longer distances, without additional logic, the transmission time on the cable becomes dominant.

Mosaic channels use a byte-by-byte flow control of request-acknowledgment signals. Therefore, as cables get longer the propagation time of these signals may become the dominant bottleneck in packet transfer. Caltech is currently debugging “Slack” chips that employ FIFOs to eliminate this bottleneck. These chips buffer the incoming bytes and “fake” the acknowledgment signal. The Mosaic channels are then able to transfer data at the normal rate of 500Mb/s. The data and model presented below is for the system without Slack chips.

Performance of the network depends on the size of the packets, the length of the links, and the “nature” of the path in use. Through extensive measurement of the network using various sized cables we have developed the following model of performance. The time, T(L,D), to transfer a packet of L bytes over a cable of D feet (the length of the longest ribbon cable along the path) is

$$T(\mathbf{L},\mathbf{D}) \approx \mathrm{MAX}\left[(1.440 + 0.020 \times \mathbf{L}),\ (0.013 + 0.003 \times \mathbf{D}) \times \mathbf{L}\right] \mathrm{us}$$

For D < 2.3 feet, no degradation due to cable length is observed. For short transfers T(L)=(1.440 + 0.020×L) us/pkt was measured. This formula is based on two components. The first term (1.440) is the per packet interrupt processing. It is the fixed overhead per packet. The second term (0.020) is the per byte cost of transmission over short distances. For longer packets, a different behavior is observed. The Mosaic request/acknowledgment handshaking protocol requires that control signals propagate the length of the channel. For long cables (> 2.3 feet) this propagation delay becomes the dominant factor in performance. The 0.003 x D term is the time it takes to propagate a signal down and back a cable of length D. This is precisely 1.5 nanoseconds per foot. Each byte pays this cost plus some additional overhead (0.013).

See Figure 7 for measured data for various packet sizes and cable lengths. Figure 7a shows that the performance is linear for large packets. Figure 7b shows the non-linear effects for small packets. These measurements fit the above model within the repeatability of the measurement (≈1%).

The transfer rate (“bandwidth”) is L/T(L). The theoretical maximum data rate (with L=∞) with these 25MHz memory chips is 1/0.02us = 50MByte/s = 400Mb/s, which is the bandwidth of the memory chips on the HI. The maximum packet rate (with L=0) is 1/1.44us = 694Kpkt/s. These figures are for the HIs, not for the channels, which are faster, as proven by tests showing the channel performance (for two competing flows) to be above 405Mb/s (for medium and long packets) and above 5.25Mpkt/s (for the shortest packets).

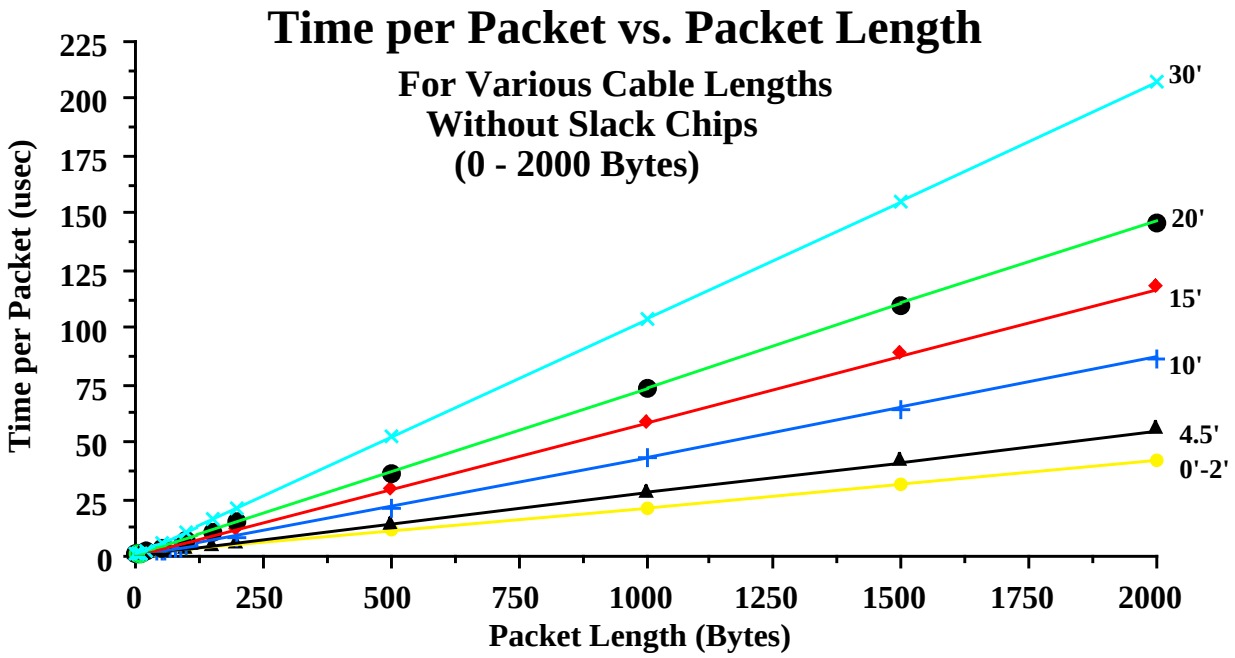


FIGURE 7a. Time per Packet (0-2,000 Bytes).

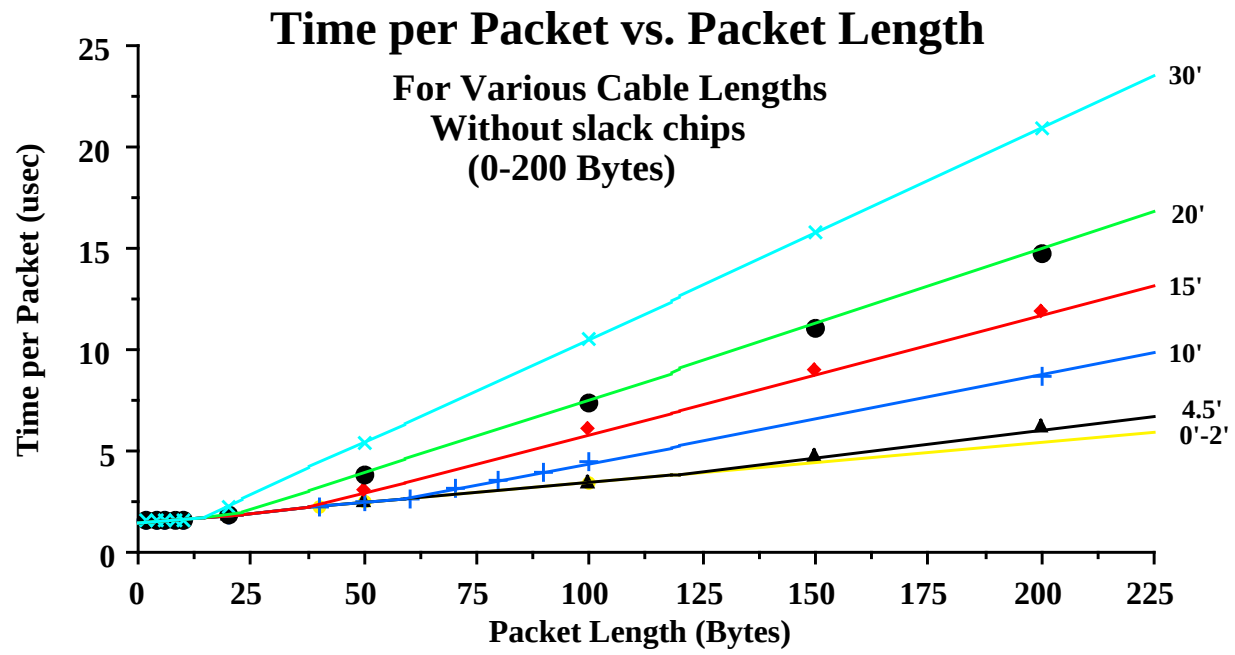


FIGURE 7b. (0-200 Bytes)

There are various pipelines in the system that operate much faster than the channels (at about 5ns/Byte). As expected, an increased level of overlap between these pipelines and the communication links improves the performance by absorbing delays.

Multi-computer designers prefer that communication channels be faster than the memory to prevent the memory from having to wait for communication. Communication system designers pre-

fer that memory be faster than the communication channels, to prevent the communication from having to wait for the memory. It is impossible to satisfy both. Memory that is slower than the communication channels (as Mosaic has) implies another bottleneck (4): store-and-forward may be performed only at half the rate allowed by memory speed (much slower than the channels) due to the additional memory accesses required in the store-and-forward node.

8.1 Improving Network Performance

The solutions to the above bottlenecks are: (1) faster memory chips, (2) faster processing, and (3) faster links and increased overlap of the communication links with pipelines. A solution to bottleneck (4), store-and-forward degradation, is to eliminate the need for it by designing the network topology such that only native Mosaic wormhole routing is needed (non-consistent connections).

Mosaic chips can operate with 32MHz memory chips that are readily available. Unfortunately, our present HIs use slower 25MHz memory chips. This limits both the transmission and reception of messages and the general processing associated with the communication. The simple replacement of these chips (and of some crystals) will improve the host interface performance of ATOMIC by approximately 25%.

The basic transfer operation of the Mosaic chips currently handles 8 bits in parallel. Extending it to handle 16 or 32 bits in parallel is primarily a matter of packaging, not of a high-risk redesign. This could improve the performance of ATOMIC by 100-200%.

Mosaic is designed in scalable 1.2um design rules. Re-fabricating it by a 0.8um process would improve the performance of ATOMIC by 50-100%.

Hence, no breakthroughs (or even small miracles) are needed to improve performance five to ten times.

These improvements apply to the ATOMIC network only, not host/UNIX overhead. To deliver gigabit rates to a workstation *application* requires major changes in the handling of network communication [5].

8.2 Better Interface Design

The ratio in performance of the ATOMIC prototype to the Ethernet is approximately 100:1. Until now most network interfaces were designed on the assumption that the network was much slower than the core communications resources within the workstation, its bus and memory. This is no longer true. The ATOMIC prototype can deliver or demand data much faster than typical workstation busses.

Current network device interfaces read packets from the network into device buffers and subsequently interrupt the workstation. One and possibly two copy operations occur, often with data conversion to a storage format more efficient for kernel procedures. The device driver software copies the packets, placing them onto kernel queues where they await distribution to consuming processes. The consumers may copy the packets again when retrieving them from kernel queues. Network output reverses that procedure [5].

For a workstation to achieve the performance that a gigabit LAN can deliver requires a new network interface design. The kernel⇔interface copying may be eliminated by allowing the inter-

face to share access to the kernel network queues. Packets may then be stored directly into those queues on input and retrieved from the queues on output [11].

This requires that the interface be sufficiently flexible that it can manipulate kernel data structures. The Mosaic CPU in ATOMIC provides such flexibility. The modifications that would allow elimination of kernel⇔interface data copying are straightforward in BSD UNIX. Elimination of the second copy operation between kernel and application processes is much more complex, but could be achieved by virtual memory manipulation.

9.0 Active Cables

Mosaic technology was designed to communicate over short distances. However, a practical LAN requires that it be possible for hosts to be separated from their switches by distances of hundreds of meters. Since Mosaic channels operate asynchronously, this raises some timing issues as well as requiring a practical method to transmit data between a switch and a host at a gigabit per second.

The asynchronous timing constraints are largely removed by a employing a FIFO to decouple the hop-by-hop and byte-by-byte flow control inherent in Mosaic channels. The FIFOs absorbs cable delays up to a certain amount without degrading the transfer rate by creating a few nanoseconds of slack per stage. The more stages the FIFO has, the more slack time the chip absorbs.

Prototype Mosaic chips have already transmitted data at a rate of 800Mb/s over simplex channels of eight data and three control signal lines. To achieve these data rates requires over a gigabit per second of cable bandwidth. In the near future that figure can be expected to rise substantially as circuit technology improves. For a practical LAN, we would need cables of a few hundred meters. These figures strongly suggest the use of serial fiber-optic cable technology. The creation of inexpensive, gigabit fiber-optic transceivers that can operate inside an air-cooled chassis in a typical office is an area of active research.

At this time, the authors are not aware of cost-effective fiber-optic transceivers well-suited for asynchronous channels. As FDDI has demonstrated, the lack of cost-effective interfaces is an important obstacle to widespread deployment of fiber-based LANs. One is currently able to find inexpensive, slow, short fiber connections and expensive, fast, long fiber connections (e.g. long-haul phone lines). What ATOMIC needs is low cost, fast cables, and we believe that this can be achieved for short distances (hundreds of meters).

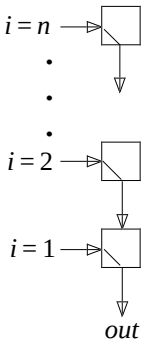
10.0 Previous Work

Having presented the details of the ATOMIC LAN it is now possible to compare it to previous work in this area.

10.1 ATOMIC vs. Switch-Based LANs

On the surface, an ATOMIC 8x8 mesh router may resemble a switch. It bears some similarity to a space division crossbar switch with a self-routing property [3][33]. 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 an 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.

10.2 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 with 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.

In the table below, the expected performance of a 64-processor ATOMIC mesh using Mosaic-C and the upcoming Mosaic-T[30] is compared to both the Autonet switch[26] and Nectar HUB[1]. For each switch we list the maximum achievable packets per second through the switch, the latency across the switch, the channel bandwidth and the aggregate switch bandwidth. The advantage of distributed routing used in an 64-processor mesh is pointed out by comparing mesh performance to that of an Autonet switch. A mesh contains 64 Mosaic chips, each with a router, each independent, while an Autonet switch contains one router. A 64-processor dual-connected mesh is exter-

nally similar to a 16 x 16 Nectar HUB crossbar.

Network/switch (# crossbar ports)	max pkts/sec	latency	channel bandwidth	aggregate switch bandwidth
Autonet (12x12)	2M	2000ns	100Mb/s	1.2Gb/s
Nectar (16x16)	14M	700ns	100Mb/s	1.6Gb/s
ATOMIC (6x6) *[Mosaic-C]*	31M	125ns	500Mb/s	3Gb/s
ATOMIC (16x16) *[Mosaic-C]*	80M	375ns	500Mb/s	8Gb/s
ATOMIC (16x16) *[Mosaic-T]*	320M	100ns	1600Mb/s	50Gb/s

10.3 ATM-Based LANs

Asynchronous Transfer Mode (ATM) has been suggested as an implementation technology for local as well as wide-area networks [22][6]. 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 [11][12][8]. The Autonet follow-on, AN2, will have ATM switches [27].

ATOMIC sends variable length packets and it uses the distributed computational and routing capability of a mesh. This 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 desired. ATOMIC is more general than all of the ATM work, which is specifically designed to support only ATM. ATOMIC supports IP, but with a software change could support ATM or any network protocol.

One additional point should be stressed. The low cost of nodes coupled with their programmability makes it practical to utilize them in workstation architecture [15]. 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 [18][6].

Some of the details of the Autonet project [26] at DEC Systems Research Center is similar to ATOMIC [7]. The Autonet LAN is also a switch-based network, and it uses point-to-point links of 100Mb/s over coaxial cable. One difference between the two projects lies in packet addressing. Autonet uses a scheme described as somewhere between source routing and routing by unique identifier. ATOMIC uses strict source routing inherited from the Mosaic chips[2][29][32]. Another difference is that Autonet's switches are responsible for reconfiguration and route selection in a distributed manner, while ATOMIC opted for a centralized address consultant process to perform this function. This difference is mainly due to the design of the Mosaic chips and the source routing used in the network. Our "switches" are actually distributed processing networks.

Packets may be source routed through a mesh without ever appearing at a Mosaic processor in the mesh. Native Mosaic routing is performed entirely by the routers on each chip in the mesh without interrupting the CPU. All the "switching decisions" can be made external to the mesh by the host generating a source routed packet. But, the fact that our meshes and host interfaces have general purpose processors associated with each router makes ATOMIC far more general and flexible than Autonet or any other similar network.

11.0 Summary

ATOMIC is an operating very-high-speed local area network. It has already demonstrated over one gigabit per second of aggregate throughput between two hosts using 1,500 byte packets, and more than 5.25Mpkt/s over a single channel with the smallest packets (2 bytes).

ATOMIC was built by using hardware designed for message-based multi-processing and adding software layers to manage inter-host communication in the IP style. Its high cost-effectiveness makes it a very attractive high-speed LAN technology. The technology that is used for our initial prototyping effort could be made faster by five to ten times without the need for any further technological breakthroughs.

12.0 Acknowledgments

Chuck Seitz of Caltech developed the Mosaic technology. Both he and Wen-King Su (of Caltech) taught us how to use Mosaic and provided us with much needed hardware and software utilities.

DARPA supports ATOMIC through the Directorate of Contracting, Ft. Huachuca contract No. DABT63-91-0001.

We thank them all for their help.

13.0 References

[1] Arnould, E., Bitz, F., Cooper, E., Samsom, R., Steenkiste, P. "The Design of Nectar: A Network Backplane for Heterogenous Multicomputers", in *Proceedings, ASPLOS-III*, pp. 205-216, April 1989.

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

[3] Cheriton, D. R. "Sirpent™: A High-Performance Internetworking Approach", *in Proceedings of Sigcomm-89*, pp. 158-169.

[4] Clark, D., Jacobson, V., Romkey, J., Salwen, H. "Analysis of TCP Processing Overhead", *IEEE Communications*, 27(6):23-29, June 1989.

[5] Clark, D. D., Tennenhouse, D. L. "Architectural Considerations for a New Generation of Protocols", *Proceedings of SIGCOMM-90,* pp. 200-208, August 1990.

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

[7] Cohen, D., Finn, G., Felderman, R., DeSchon, A., "The ATOMIC LAN", presentation at the IEEE Workshop on High Performance Communication Subsystems (HPCS92), February 1992.

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

[9] Cooper, E., Menzilcioglu, O., Sansom, R., Bitz, F. "Host Interface Design for ATM LANs", *Proceedings of the 16th Conference on Local Computer Networks*, pp. 247-258, October 1991.

[10] Dally, W. J., Seitz, C. L. "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks", *IEEE Transactions on Computer*s, Vol. C-36, No. 5, May 1987.

[11] Davie, B. S. "A Host-Network Interface Architecture for ATM", *Proceedings of SIGCOMM-9*1, pp. 307-315, August 1991.

[12] 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.

[13] 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.

[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] Finn, Gregory G. "An Integration of Network Communication with Workstation Architecture", *ACM Computer Communication Review*, Oct 1991.

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

[17] 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.

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

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

[20] Kanakia, H., Cheriton, D. "The VMP Network Adapter Board (NAB): High Performance Network Communication for Multiprocessors", *Proceedings of SIGCOMM*-88, pp. 175-187, August 1988.

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

[22] Leslie, I., McAuley, D. “Fairisle: An ATM Network for the Local Area”, *Proceedings of SIGCOMM-9*1, pp. 327-336, August 1991.

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

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

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

[26] Schroeder, Michael D., et. al. “Autonet: A High-speed, Self-Configuring Local Area Network Using Point-to-Point Links”, *IEEE Journal on Selected Areas in Communications,* Vol. 9, No. 8, October 1991.

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

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

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

[30] Seitz, C.L., Seizovic, J., Su, W. “The Design of the Caltech Mosaic C. Multicomputer”. Submitted to the *Symposium On Integrated Systems*, Seattle, Wash, October 1992.

[31] Seitz, C.L., Su, W. “A Family of Routing and Communication Chips Based on the Mosaic”, Submitted to the *Symposium On Integrated Systems*, Seattle, Wash, October 1992.

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

[33] 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.

[34] Traw, S.B.C, Smith, J.M. “A High-Performance Host Interface for ATM Networks”, *Proceedings of SIGCOMM-9*1, pp. 317-325, August 1991.

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

[36] 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.*