Pulsar: Non-blocking Packet Switching with Shift-register Rings*

Gary J. Murakami, Roy H. Campbell, and Michael Faiman

University of Illinois at Urbana-Champaign
Department of Computer Science
1304 W. Springfield Avenue
Urbana, Illinois 61801–2987 USA

Abstract

This paper discusses the design of a switch for high-speed computer networking at gigabit rates. We present the *Pulsar* switch, a non-blocking design based on a high-spin-rate, port-dedicated, word-parallel, shift-register ring. Several design alternatives address the problem of Head-Of-Line blocking. In contrast to Batcher-Banyan switches, access to the ring is asynchronous which facilitates low delay and arbitrary packet length. The switch can support ATM cells simultaneously with packets sized for applications such as single characters, memory words, disk blocks, memory pages, or video images. *Pulsar* can be used as a high-throughput computer backplane replacement. The design can be implemented with existing high-speed circuit technology.

1 Introduction

High-speed networks are needed to support integrated broadband services which includes the transmission and switching of data, voice, and video. This presents a major challenge since a wide range of data rates and packet sizes from individual characters to video images are transported through a single network fabric. A circuit switched solution provides guaranteed bandwidth and predictable delay, but limited resources must be allocated inflexibly and inefficiently. In contrast, packet switching does not necessarily guarantee bandwidth or delay, but it provides a high degree of multiplexing with flexible and efficient resource allocation on demand.

*This work was supported in part by AT&T & NSF

Improvements in fiber-optic transmission technology are scaling up the speed and distance of computer networks. Delay due to blocking, routing, queueing, and protocol processing must be minimized to reduce buffering requirements because of the high product of bandwidth and propagation delay. High speeds, long distances, and variable rate traffic emphasize the need for high-throughput packet switching. Slow, complex, traditional software switching methods must be replaced.

Our view of a packet switch is a single box or circuit board which switches packets between a collection of point-to-point links. Transmission and propagation concerns are separated from the switching function. Our approach is to use simple shift-register rings for packet switching. The simple design enables speed-up for non-blocking throughput so that the switch is no longer the bottleneck in the network.

We have developed a high-speed packet switch design named "*Pulsar* ". The design endeavors to minimize complexity while providing non-blocking throughput and self-switching logic. High-speed circuitry implements a fast, slot-dedicated, shift-register ring. A non-blocking switch with 16 or more ports and gigabit per second transmission links can be realized using current technology [1]. The switch design can also function as a high-throughput system interconnect backplane which replaces the system bus in computer systems. This paper presents the basic *Pulsar* switch design, major design alternatives, some simulation results, and a discussion of the merits of the design.

2 Switch Architectures

This overview compares the features of major switch architectures and provides a frame of reference for the *Pulsar* switch. Since high-throughput is a major con-

[1] *Pulsar* capitalizes on high-speed electronic technology and does not depend on further improvements in optical switch components.

cern, the selected architectures are informally categorized with respect to blocking.

2.1 Blocking

There are three types of blocking for packet switching. *Internal blocking* is a loss of throughput through the interior of a switch. This is typically due to contention for critical resources or insufficient internal bandwidth. A necessary requirement for an internally non-blocking switch is that the bandwidth internal to the switch must be greater than or equal to the aggregate bandwidth of the links. *Output blocking* occurs when packets conflict for the same output port. A conflict resolution mechanism is needed to determine which packet proceeds to the output port while the other packets are deferred. *Head-of-line blocking* occurs at an input queue when the packet at the head of the queue blocks packets behind it. Suppose packets are processed strictly in first-come-first-served order. If the head packet cannot be delivered due to output blocking, it blocks other packets behind it even if they are destined to output ports that are not busy.

2.2 Internally Blocking Switches

Traditional packet switches have internal blocking since a shared resource is used for switching. A packet being switched occupies the shared resource and blocks other packets from being switched even if there is no contention for the output port. Internal bandwidth is $O(1)$ with respect to port speed.

Buses: The bus architecture is used for both computer systems and computer networks, for example, the VME bus and Ethernet, respectively. The bus structure is a broadcast medium with full connectivity. Computer system buses use parallel lines for high bandwidth, and the physical backplane configuration enables pluggable modules for CPUs, memory, and I/O devices.

However, the bus is a shared resource which limits throughput. In computer systems, CPUs and I/O contend for the bus and block each other. In order to reduce contention, multiple buses are often arranged in a segmented or hierarchical structure. For example, several graphics workstations have separate "pixel" buses for graphics processors and display memory. However, a transfer of an image from disk to display memory still blocks the pixel bus, the system bus, and the bridge between the buses. Multiprocessors based on a shared-bus architecture are limited to around 32 nodes [5].

Token Rings: Token rings are used primarily for computer networks (e.g., FDDI, IEEE 802.4) but can also be used as a multiprocessor interconnect structure [7] [3]. The ring is fully connected and can provide a broadcast medium. Token logic is a simple mechanism for arbitrating access to the medium. Some ring networks permit spatial reuse for improved utilization and throughput [2]. However in most implementations, the ring is a shared resource which limits throughput. Token rings typically have higher delay than computer buses due to ring transit time.

2.3 Low-blocking and Non-blocking Switches

In internally non-blocking switches, the switch has sufficient capacity to support switching to all output ports simultaneously. However, careful design is needed to overcome output blocking and head-of-line blocking.

Crossbars: The crossbar is a matrix switch used in telephone systems, computer networks, and in computer systems [17]. Crossbar switches have been implemented optically [15] and electrically in silicon [19]. The crossbar is non-blocking and can support broadcast capability. A major drawback to the crossbar is the requirement for switch setup via external control for each packet or set of synchronous bounded-size packets. A coordination mechanism must be used to provide fairness for crossbar setup. Scaling of a crossbar requires N^2 components.

Shuffle Networks: Unlike crossbar switches, shuffle networks are self-switching: the destination port address for a packet is used to switch through a binary sorting network bit by bit. The crosspoint complexity of shuffle networks is $O(N \log N)$. Several projects researching optical interconnection use variations of the perfect-shuffle connection pattern [11] [22].

Batcher-Banyan switches use a Batcher sorting network and an associated mechanism to resolve output conflicts before delivery to the Banyan network, however, the switch still suffers from some internal blocking. Copy networks have been added to Batcher-Banyan switches to provide multicast capability [14].

A major problem for Batcher-Banyan networks is resolving contention for output ports. The Starlight [2] [8] 32 switch adds a purge-skew-concentrate stage before the Banyan network. Packets which have lost the contention are recirculated and reentered into the Batcher network. Packets can be lost due to blocking within the reentry network, and packets may be delivered out of sequence. An alternative to the Starlight contention resolution is the three phase algorithm: [9] (I) send and

[2]Starlight is a registered trademark of AT&T.

resolve request, (II) acknowledge winning port, and (III) send packet.

2.4 Other Topologies

Hypergraphs: Hypergraphs are often used to interconnect multiprocessors like the Intel iPSC/2. The multiplicity of links provides a higher aggregate bandwidth for for communication. As a result, hypercube-based multiprocessors can be scaled higher than bus-based multiprocessors. The longest path between any two nodes is $O(\log_d N)$ where d is the degree of a node.

Mesh networks: Mesh networks are another interesting regular structure, and they can be implemented in VLSI [25]. The Manhattan Street Network [16] is a planar network fabric that uses deflection routing. If possible, packets are routed toward their destination. If the network is congested, packets are not queued, instead they are temporarily misrouted on a free link. So queues are eliminated in the network fabric, and buffer resources are static. However, the Manhattan Street Network suffers from less predictable delays and out-of-sequence packet delivery.

Many different different topologies can be used for constructing a larger switching fabric from smaller switch components. Note that most of the common topologies suffer from internal blocking.

3 Design

The basis for the *Pulsar* switch design is a high-spin-rate port-dedicated word-parallel shift-register ring. The architecture separates switching and transmission; a central switch moves data between local ports which have fiber transmissions lines or trunks to other switches [3]

3.1 Previous Work

The new design is an outgrowth of two hitherto unrelated results of our earlier research. The first is the design of a word-parallel shift-register ring as the communication mechanism for a shared-memory multiprocessor [7]. The second is a general technique for hardware conversion between an optical bit stream and electronic data words [18].

[3] Switching, transmission, distribution, and processing are intermixed in some network designs. For example, Ethernet functions as both a broadcast/filter switch and a coaxial-cable transmission/distribution medium. Some applications such as file transfer can tolerate the impact of transmission delay on switching, but other applications such as multiprocessor interconnection are adversely affected by delay.

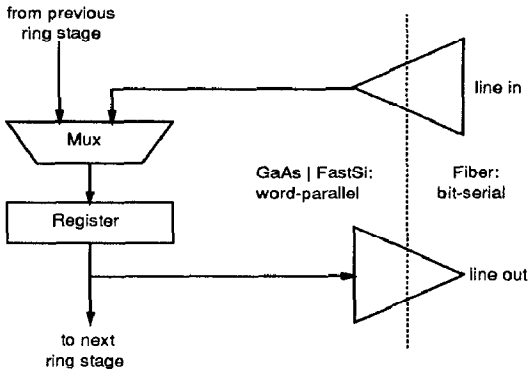

Figure 1: One Stage of the Shift-Register Ring

3.2 Basic Switch Design

The switch exploits currently developing high-speed circuit technology, e.g., gallium-arsenide (GaAs) or fast silicon (FastSi) technology, in order to effect a smooth bandwidth transition between serial optical bit rates and clock rates of parallel digital circuits. The core of the switch design is a word-parallel shift-register ring.

A segment of the data paths in one stage of the switch is shown in Figure 1. For a switch with p ports – each port is one input plus one output line – p of these stages are cascaded in a ring. Incoming optical data at a *serial* bit rate of b bits per second are converted to w-bit words, as shown on the right of the figure, double-buffered, and clocked into the ring at a *port* rate of b/w words per second, assuming sustained input. Data to be removed from the ring are simply gated from the output of the appropriate register and converted from parallel to serial format, as shown in the figure. All ports are capable of simultaneous activity.

Unlike other ring designs, the *Pulsar* ring is clocked at a much faster rate than the port rate to achieve non-blocking throughput. The shift-registers composing the *ring* are driven with a common clock frequency of pb/w MHz. Thus, the ring constitutes a set of circulating slots, with the entire set making one complete revolution in the time between successive word insertions from any one port. One practical set of parameters is:

serial bit rate	b	1 gigabit/sec
number of ports	p	16 ports
word size	w	64 bits
port clock rate	b/w	15.625 MHz
ring clock rate	pb/w	250 MHz

The speed of GaAs or FastSi technology is exploited for implementing the 1 GHz serial-to-parallel converters and the 250 MHz shift-register ring. The design minimizes complexity for the high-speed switch circuitry.

The port logic at 15.625 MHz is slow enough to use existing Si memory and processor technology to implement routing and queueing.

Shift-register slots are dedicated to a port [4]. This allows the ports to run at a lower speed while the ring circulates at higher speed for non-blocking throughput. Output-port dedicated slots requires input queueing and token logic; input-port dedicated slots requires output queueing and filter logic. Packets of arbitrary size can be switched contiguously through the ring. A packet is switched word by word as the appropriate dedicated slot cycles by the port. The design alternatives are examined in further detail in the next section.

The basic design has several important characteristics:

- *internally non-blocking*: Full simultaneous transfer through all ports is possible given evenly distributed traffic [5].
- *self-switching*: Packets are self-switching; external per-packet switch setup is not required.
- *extremely low delay*: A word is switched through the ring in one 64 nsec. port clock cycle which is attractive for multiprocessor interconnection. For networking applications, switching of a packet can begin asynchronously as soon as it arrives [6].
- *elegance*: The design is simple and regular and facilities implementation in high-speed circuit technology.
- *arbitrary packet size*: Packets of arbitrary size can be switched through the ring *contiguously*.
- *easy scaling*: Appropriate parameters can be chosen for linear scaling [7] (e.g., number of ports, word width) and to match selected technology (e.g. Si, FastSi, or GaAs), speed, and packaging constraints.
- *multiprocessor interconnection*: The same switch design can be used for both computer networks and computer processor-memory systems as a non-blocking system bus replacement.

[4] This is somewhat similar to TDM ring networks [21] but slots are one word wide, implemented with shift registers, and only consume one clock cycle.

[5] Loop-back paths are also possible.

[6] Cut-through switching can be implemented to minimize queueing delay.

[7] Due to physical limitations, a single switch is limited in size (e.g., board size limitations). The ring rate may also be limiting: practical sizes for a single switch range from 8 ports to maybe 128 ports. A switch with more ports will have to be constructed using higher level networks of switches.

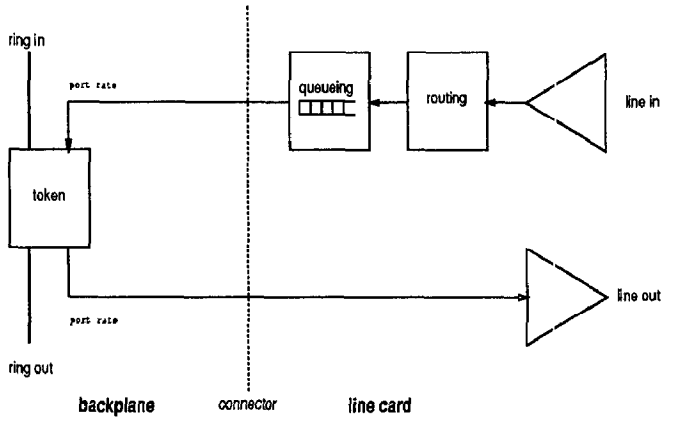

Figure 2: Block Diagram for the Single-Head Queue Alternative

4 Design Alternatives

The primary feature of the major design alternatives is the placement of the queues and the corresponding slot dedication and switching logic We have developed three major alternatives: a *single-head* input queue with token logic, multiple output queues with *filter* logic, and a *multiple-head* input queue.

4.1 Single-head

A *single-head* input queue per port is the most straightforward design alternative with N queues for an N port switch. Figure 2 shows a block diagram for this design alternative and has the following basic features: (1) queueing occurs at the input port, (2) a slot is dedicated to each output port, and (3) token logic is used for ring access [8]. Ring access via token logic is packet-per-port fair [9]. This basic design alternative demonstrates Head-of-Line blocking which lowers throughput.

The following sequence lists the major events for switching a packet.

Input port packet arrival A packet arrives at the input port. The routing module determines the output port for the packet. The input port has a single queue with space for one or more packets [10].

Wait for output port token: The input port waits for an empty slot for the output port.

Input port transmission The input port transmits the packet word-by-word through the output slot. At the end of the packet, the token is put back on the ring.

[8] Additional token bits are needed for each port-bit-slice chip.

[9] Input queueing is not word/byte/bit fair if packet size varies.

[10] Both FCFS and FQ (Fair Queueing) queue disciplines are under investigation.

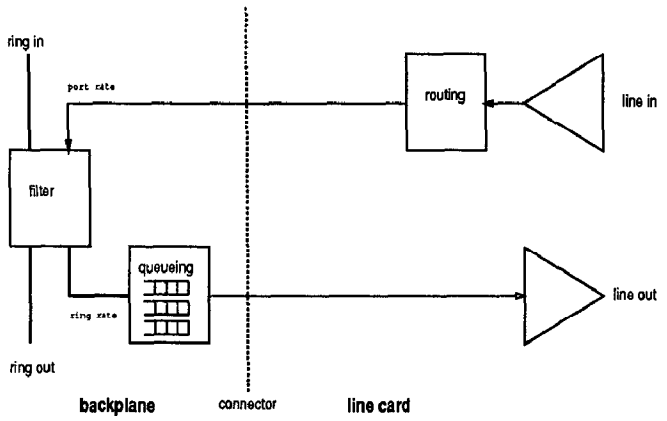

Figure 3: Block Diagram for the Filter Alternative

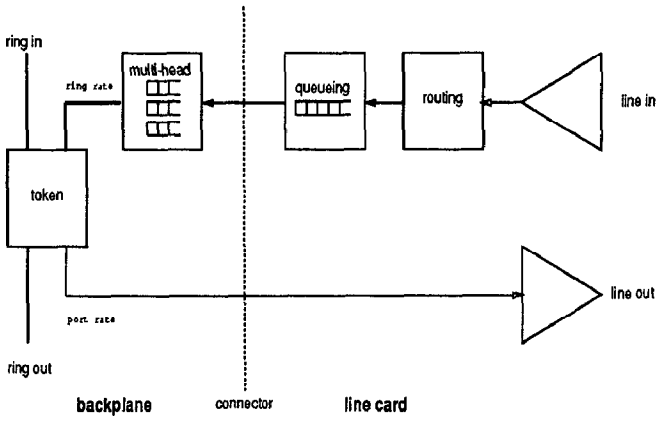

Figure 4: Block Diagram for the Multi-Head Queue Alternative

Output port reception The output port accepts data from its slot and transmits it directly on the fiber.

The *single-head* input queue design suffers Head-Of-Line (HOL) blocking. If an input port has a packet destined to an output port that is busy with a packet from a different input port, then packets that arrive at the input port are blocked even if they are destined to an output port that is not busy. Simulation results and analysis show that HOL blocking lowers throughput to 59% $(2-\sqrt{2})$ [9] [6] for the saturated-switch worst-case.

4.2 Filter

A multiple output queue design eliminates HOL blocking but has M*N queues for an N port switch with M queues per port. Figure 3 shows a block diagram for this design alternative and has the following basic features: (1) queueing occurs at the output port, (2) slots are dedicated to input ports, and (3) filter logic is used for ring exit [11]. Proper buffer selection can provide fairness [12]. The major drawback is that high-speed output-port demultiplexing and queueing logic and buffer memory is needed since the ring output runs at M times port rate.

The following sequence lists the major events.

Input port packet arrival/transmission A packet arrives at the input port. The routing module determines the output port, the packet is directly injected into the slot dedicated to the input port.

Output port filtering and buffering The output port filters, demultiplexes, and buffers data separately to one of the M queues to preserve contiguity.

Output port buffer selection The output port selects the buffer to transmit on the fiber via round robin or Fair Queueing discipline.

The *filter* alternative does not suffer from HOL blocking. Additional address bits are required for each word and chip. Data can arrive simultaneously from several or all input ports. High-speed demultiplexing logic is needed at the output port. Sufficient memory bandwidth is required to absorb the data. So separate per input-port queues must be maintained.

With input queueing, the queue memory and queue discipline can be implemented completely with slower port-rate circuitry. However output queueing requires that queue memory and queue-discipline logic be closedly coupled with faster ring-rate circuitry. Data can be lost without warning due to the queueing discipline which is less acceptable for multiprocessor interconnection applications. These problems are major drawbacks to the *filter* alternative.

4.3 Multi-head

The next alternative is a *multi-head* input queue with N M-head queues for an N port switch. Figure 4 shows a block diagram for this design alternative and has the following basic features: (1) queueing occurs at the input port, (2) slots are dedicated to output ports, and (3) token logic is used for ring access [13]. Ring access via token logic is packet-per-port fair [14]. High-speed queue head and input-port multiplexing logic is needed,

[11] Address bits are needed to specify the destination port

[12] Output queue combined with Fair Queueing provides approximate word/byte/bit fairness. The Input queue alternatives are packet fair which is unfair if packet size varies.

[13] Additional token bits are needed for each port-bit-slice chip

[14] Input queueing is not word/byte/bit fair if packet size varies.

but the amount of high-bandwidth buffer memory is reduced.

The following sequence lists the major events.

Input port packet arrival A packet arrives at the input port. The routing module determines the destination port for the packet and queues the packet in the large slow-speed main queue on the line card. If the high-speed queue head memory on the backplane is not full, the packet is transferred and inserted into the queue head for the appropriate destination.

Wait for output port token As each output designated slot is presented to the input port, token logic is run using the corresponding queue head. In effect, the input port is simultaneously waiting for the output port tokens for each of the multiple queue heads.

Input port transmission The input port queue head transmits the packet word-by-word through the output-dedicated slot. At the end of the packet, the token is put back on the ring. Multiple queues from the same input port can be transmitting simultaneously.

Output port reception The output port accepts data from its slot and transmits it directly on the fiber.

With a *multi-head* input queue, HOL blocking is eased as M increases to N. To implement the multi-head part of the queue, high speed multiplexing logic is needed at the input port along with enough queue memory to overcome HOL blocking. It is likely that this may fit on one chip and possibly on the same chip as the shift registers.

4.4 Other Alternatives

We previously discussed and discarded several other design alternatives. One design alternative that merits further evaluation is outlined in Figure 5. As packets arrive at an input port, they are separated into individual queues for each destination. A request message from the queue for the corresponding output port is sent from the line card to the selection manager on the backplane. When contention for an appropriate token succeeds, contention for other requests is suspended. A signal specifying the "selected" port is sent back to the queue, and the the winning packet is completely transmitted [15]. Then contention cycling resumes. This design lacks fairness and can suffer from indefinite postponement. For example, suppose that an output port is continuously busy receiving data from one or more input ports. An input port with many pending packets to separate destinations is likely to miss the token for the busy output port as it contends for and transmits to other output ports. However, the unfairness can be bounded by metering the requests submitted by the input port. The major benefit of this design is that HOL blocking is reduced without the need for high-speed memory.

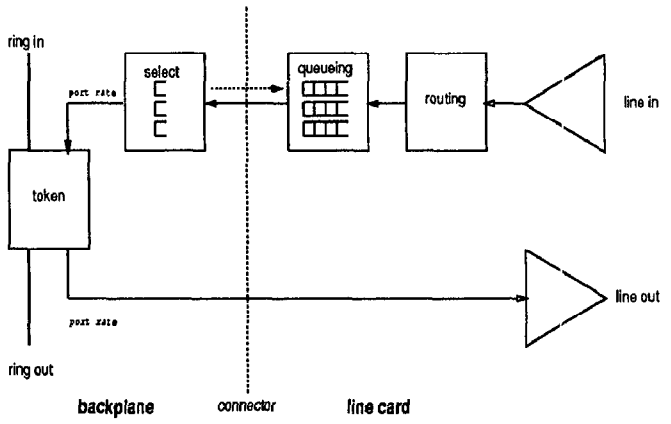

Figure 5: Block Diagram for the Select Alternative

Perhaps the most promising design is a modification of the filter alternative. We select an output buffer size large enough to ease HOL blocking but small enough to fit chip packaging constraints. A slow-speed input queue is added to buffer packets temporarily when output-port blocking occurs. This design is the exact dual to the multi-head queue. It has both large slow-speed input queues and small high-bandwidth output queues. We intend to study the effects of various output queue sizes along with comparison to other designs.

5 Implementation Issues

The following issues discuss implementation details and support feasibility of constructing the switch.

5.1 Cut-through

To minimize delay, packet cut-through switching should be considered. For example, a simple queue implementation would wait to receive a whole packet before passing it on. If the queue output is free, a cut-through implementation would present the packet to the queue output while it is being received with a delay of only one or two word delays instead of a packet delay [16].

[15] To avoid a word delay at the start of the packet, a register per port is needed to save the first word of a packet which is transmitted at request time. This pipelines the packet transmission when contention succeeds for a request.

[16] Cut-through may not be immediate due to routing delay which cannot be bypassed.

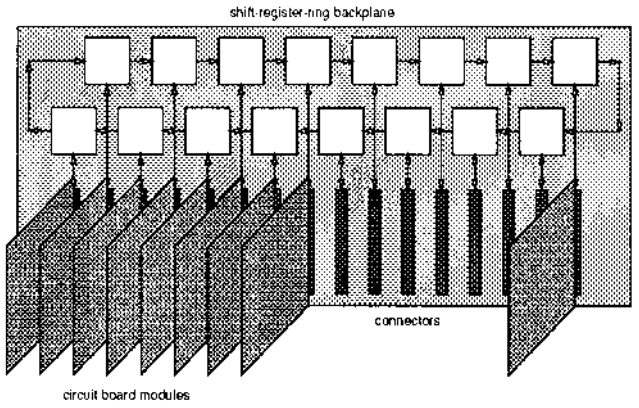

Figure 6: *Pulsar* Backplane Physical Configuration

5.2 Physical Packaging

The *Pulsar* design has two speeds of circuitry: high-speed 250 MHz ring rate and low-speed 15.625 MHz port word parallel clock rate. The high-speed ring-rate circuitry requires GaAs or FastSi technology. The ring-rate circuitry would be laid out on one PC board with differential drivers high-speed lines and minimal distances between chips. The lower-speed port rate is compatible with bus connector technology and lower-speed Si technology. So for the single-head and multi-head queue designs, the queueing discipline can be implemented with standard Si technology on a separate PC board module. The ring has a regular structure that facilitates simple layout and wiring. It can be physically configured as a backplane with connectors for circuit-board modules. Figure 6 illustrates a backplane physical configuration.

5.3 Chip Packaging

We currently favor a port-bit-slice chip packaging for the ring. The ring is divided into chips via port and bit slice. Current chip packages of 256 pins [17] are adequate for a 32-bit slice. So a 64-bit word 16-port switch would require 32 chips. The port-bit-slice approach permits the construction of rings with an arbitrary number of ports and a word size that is a multiple of the slice size.

The example ring rate of 250 MHz can be implemented in FastSi as well as GaAs technology. Faster clock rates can be chosen, but the 250 MHz rate can be implemented on PC boards with balanced lines without resorting to micro-coax.

Port interface logic for the multi-head queue alternative can be implemented on a separate chip or integrated into the ring chip [18].

A common chip or chip-set subset for both computer network switching and computer system interconnection is an attractive idea. However the differing applications may require sufficiently different circuitry support which may not be feasible with a common chip.

5.4 Network Switch

Routing and queueing mechanisms are needed to complete the network switch design.

5.4.1 Routing

Routing is determining the output port for a packet based on information in the packet header. The resulting local destination port identifier is used to switch the packet. There are three standard routing methods: source routing, destination routing, and virtual circuit routing. Source routing requires either headers of arbitrary length in order to span a network of arbitrary width, or else a path length limit, which places restrictions on network width. For every packet at each switch, destination routing requires a relatively expensive destination lookup in a database to determine the output port. [19] Virtual circuit routing isolates expensive destination symbolic name or address translation to a circuit setup phase, so subsequent packets follow the existing virtual circuit with reduced address translation cost. We favor virtual circuit routing since it facilitates high-speed switching and queueing, and also network administration, billing, authentication, security, and management. However, the *Pulsar* switch design is compatible with all three routing methods. The routing module could support more than one routing method; it is possible that all three methods could be supported simultaneously.

5.4.2 Queueing

The queueing discipline determines how buffer resources and corresponding bandwidth are utilized. It provides a protection mechanism for determining which packets to discard under congestion. We currently favor the implementation of Fair Queueing [4] which approximates bit-by-bit round robin queueing. Another possibility is Hierarchical Round Robin (HRR) [12] [20] which pro-

[17] We are contemplating the construction of a scaled-down prototype built around PALs or gate arrays. ASICs will be considered for further implementation.

[18] Note that this may place size restrictions on the ring

[19] Destination address translation has been too expensive for high-speed routing even using sophisticated software searching techniques. We are investigating VM (Virtual Memory) mapping which capitalizes on MMU (Memory Management Unit) hardware to manage a sparse cache of destination addresses.

[20] HRR assumes fixed length packets. FQ provides RR fairness for variable length packets. Mechanisms from both HRR and FQ are needed for supporting hierarchical priority and variable length

vides dedicated bandwidth and traffic priorities. As mentioned previously, the single-head and multi-head queue alternatives are only packet fair per port. This is fine for fixed packet sizes or ATM cells but unfair for varying packet sizes. However, the output-queue filter alternative can be word fair regardless of varying packet sizes.

5.5 System Interconnect

The bus is a bottleneck in many computer systems and their applications. Multiprocessors are limited in scale due to bus and memory bandwidth limitations. High-speed graphical workstations or "viewstations" have a pixel bus to provide an additional segment of bandwidth, but throughput is still limited for common activities such as moving images between a frame buffer and disk where both pixel bus and system bus bandwidth is consumed.

The *Pulsar* switch can be used to replace the system bus for interconnecting computer system components. This provides non-blocking throughput and removes the bus bottleneck. For multiprocessors, multiple memory boards plugged into the ring provide higher aggregate memory bandwidth for improved scaling. For viewstations, CPU processing and image transfer can occur simultaneously. In contrast to Batcher-Banyan switches, *Pulsar* provides asynchronous access, arbitrary length packet transfer, and very low delay. This facilitates its use of *Pulsar* as a system backplane. Memory fetches by a CPU use short packets. Disk blocks or image transfers use large packets.

For system interconnection, physical addresses and simple buffers suffice for routing and queueing. Modules submit requests and wait for responses. This reduces the need for sophisticated queueing. Round trip request-response delay must be fast enough to minimize processor wait states. With token logic, the token is released for fairness, so a continuous conversation between two ports is limited to every other request. Filter logic does not suffer from this restriction. While system interconnection is simpler than packet switching, there are several design issues that need to be considered. However, the *Pulsar* switch appears to be extremely attractive as a non-blocking backplane for system interconnection.

5.6 Snooping Caches and Broadcast Capability

Snooping caches are very popular for shared-memory multiprocessor designs. In all three *Pulsar* design alternatives, each word injected into the ring passes by each port in the ring. So it is possible to implement snooping caches and broadcast capability. If the snooping cache is placed on the backplane side rather that the module side, processor-to-cache bandwidth is limited to the port rate. If the snooping cache is place completely on the module side, it looses synchronization with the ring since cache updates can only be fed to it at port rate. Another alternative is to use a mechanism like Network Virtual Memory which provides coherency at the page level. This solves the bandwidth mismatch problem but has coarser granularity. The caching capability must be weighed against the added complexity.

6 Simulation and Analysis

The crossbar switch is a simple example of a non-blocking switch and is a useful model for understanding simultaneous traffic through a switch. Data flow in input queueing alternatives of *Pulsar* is logically similar to a crossbar with switch setup that cyclically and non-exhaustively serves each port one packet at a time.

Simulations have been run for the single-head and multi-head queue design alternatives [21]. For these simulations, a fixed packet size of 4104 (8 header + 4096 data) bytes has been chosen to match Network Virtual Memory service with 4K pages [1]. For comparison, some simulations have been run with a fixed packet size of 64 bytes (8 header + 56 data) bytes corresponding to encapsulated ATM cells [22] Packet destinations are uniformly distributed, and simple FCFS queueing is used.

6.1 Single-Head: Saturated

Simulation of the single-head queue design alternative shows throughput loss due to HOL-blocking. Switch input rate is the rate of data flow into the switch which is equal to the sum of the input rates for the input lines. Occupancy is the input loading ratio: the current switch input rate divided by the maximum switch input rate. Switch throughput is the rate of data flow through the switch which is equal to the sum of the output rates for the output lines. Efficiency is the switch throughput divided by the switch input rate.

A fully saturated switch has all input ports loaded with no inter-packet gap between packets being received. For the single-head queue design with infinite length queues, simulation results show that efficiency is 59%.

The simulation results agree with the theoretical analysis for head-of-line blocking which shows that through-

packets.

[21] Further simulation is in progress for the filter design alternative.

[22] The ATM cell has 5 byte header and a payload of 48 data bytes.

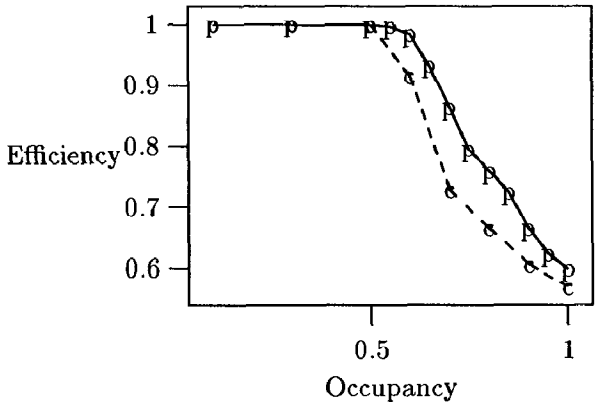

Figure 7: Single-Head: ATM Cell, Page Size Packet

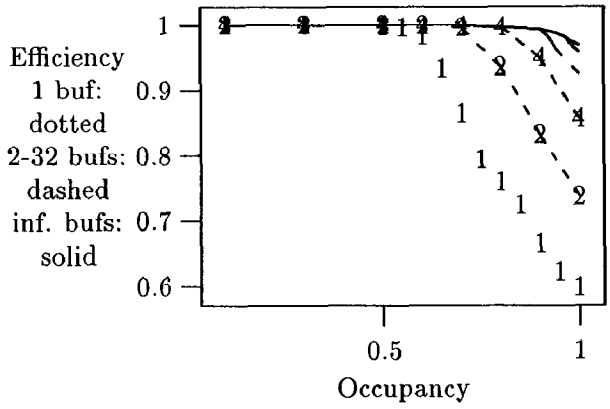

Figure 8: Multi-Head: Selected Head Buffer Limits

put efficiency is limited to $(2-\sqrt{2})$ [9]. The HOL blocking problem is common to various switch designs with input queueing.

6.2 Single-Head: Occupancy Versus Efficiency

Figure 7 plots efficiency for a range of occupancy values. The dashed line shows the results for large page size packets. As expected, efficiency begins to drop as occupancy increase above 50%. The drop in efficiency is approximately linear.

6.3 Single-Head: Packet Size

Figure 7 shows efficiency curves for packet sizes with 64 data bytes and 4096 data bytes [23]. In the simulation, the smaller packet size has slightly lower efficiency than the larger packet size. Note that the difference in efficiency is small relative to the large difference in packet size.

6.4 Multi-Head: Head Buffers

In the single-head design alternative, each port presents only one packet at a time for injection into the ring. Each port contends for only one destination-dedicated slot at a time. In the multi-head queue design, this limitation is overcome with additional high-speed ring rate circuitry. For each port, the "multi-head" is really a set of simple FCFS queues, one for every destination port. As a slot is presented to the port, the corresponding queue is used to contend for or transmit into the slot. So each port can contend for every destination-dedicated slot. Words from multiple packets can be injected into the ring in a single ring revolution.

The logic for the multi-head queue mechanism runs at the high-speed ring rate, so the queues must be simple and buffer memory is limited, since it must fit onto the single "ring" chip. Note that the main queue is low speed, implemented with multiple chips, and located on the separate PC board module. The main queue feeds into the ring chip in FCFS order. As a packet arrives in the ring chip, a buffer for the packet is allocated and enqueued into the proper destination queue. Each non-empty queue contends to transmit a packet into the corresponding slot. If all buffers are allocated, the multi-head is busy and additional packets are not fed to the ring chip.

Figure 8 shows efficiency for head buffer limits of 1, 2, 4, 8, 16, and 32. Additional buffers improve efficiency with diminishing returns, and the measurements for 32 buffers and infinite buffers are essentially identical. For a 90% occupied switch, 8 buffers are sufficient for 99% non-blocking efficiency. Note that the maximum throughput at saturation improves as more buffers are used.

7 Directions

Further research on the *Pulsar* switch and the design alternatives is planned, along with research into related issues. The basic approach is a combination of simulation and analysis. The various design alternatives will be simulated for measurements such as throughput and

[23] Each packet in Pulsar has an additional 8 byte (1 word) overhead for the packet route address.

delay. There is a substantial amount of literature which includes the analysis of token rings based on polling models [6] [23] [10] [20] [13] [24]. An analytical model for *Pulsar* will be developed.

8 Comparison to Other Network Architectures

The following table presents a preliminary an cursory comparison of crossbar, Batcher-Banyan, and Pulsar switch architectures.

	clock	complexity	thruput	comments
crossbar	b/w	$O(p^2)$	59%	sched.
Batcher -Banyan	b/w	$O(p \log p)$	70%	synch.
Pulsar				asynch.
singleH	pb/w	$O(p)$	59%	
filter	pb/w	$O(p)$	100%	p^2 Qs
multiH	pb/w	$O(p)$	97%	p^2 Qs
select	pb/w	$O(p)$		p^2 Qs

While *Pulsar* has less complexity which facilitates simple layout and wiring, it also has a higher clock rate. The high clock rate is proportional to the number of ports and sets a bound on scaling. The crossbar suffers from Head-Of-Line blocking. Batcher-Banyan networks have mechanisms to ease HOL blocking. In the Starlight switch, packets that loose contention are saved in a shared buffer and recirculated at higher priority for the next phase. In the Sunshine switch, multiple contention rounds provide a window into the head of the queue. The design alternatives for *Pulsar* use various mechanisms to address HOL blocking.

The crossbar requires a scheduling mechanism to set up the switch. In contrast, both Batcher-Banyan and *Pulsar* are self-switching. Batcher-Banyan switches are synchronous: packets must be submitted to the switch at the same time since contention resolution is simultaneous for contending packets. *Pulsar* is asynchronous: this permits low-delay switching for arbitrary packet sizes which can be efficiently matched to applications including memory word fetches, ATM cells, disk block, pages for network virtual memory, and video images [24].

The comparison between the Batcher-Banyan and *Pulsar* designs can be summarized with respect to the intended application. If the non-blocking switching of ATM cells is the only intended service, then the Batcher-Banyan design is a mature solution. However, if system interconnection or arbitrary packet sizes are important, then *Pulsar* is a simple solution that provides flexibility for increases in both speed and packet size.

[24] Larger packet sizes improve efficiency but reduce multiplexing

9 Conclusions

Pulsar is a non-blocking switch design based on a high-spin-rate, port-dedicated, word-parallel, shift-register ring. Several design alternatives address the problem of Head-Of-Line blocking. Preliminary simulation and analysis show that the basic simple-head queue design suffers from head-of-line blocking. The filter alternative has merit for fairness and robustness but requires high-speed demultiplexing and queueing circuitry with a large buffer memory to avoid packet losses. The multi-head queue alternative achieves non-blocking throughput with a modest amount of high-speed circuitry for the multi-head queue. The select alternative eases head-of-line blocking with a minimum of high-speed circuitry but is unfair.

Since access to the ring is asynchronous, packets of arbitrary size can be switched with minimal delay. *Pulsar* can be used as a high-throughput system backplane which is especially promising for multiprocessors and "viewstations". We will continue our research and will investigate the possibilities for implementation of the design.

References

[1] R. Campbell, V. Russo, and G. Johnston. A Class Hierarchical, Object-Oriented Approach to Virtual Memory Management in Multiprocessor Operating Systems. Technical Report UIUCDCS-R-88-1459, Department of Computer Science, University of Illinois at Urbana-Champaign, 1988.

[2] I. Cidon and Y. Ofek. Metaring - A Full-Duplex Ring with Fairness and Spatial Reuse. Technical Report RC 14961(#66845), IBM Research Division, September 22 1989.

[3] G. S. Delp, A. S. Sethi, and D. J. Farber. An Analysis of Memnet: An Experiment in High-Speed Shared-Memory Local Networking. *Computer Communications Review*, 18.4:165–174, August 1988. SIGCOMM "88 Conference Proceedings.

[4] A. Demers, S. Keshav, and S. Schenker. Analysis and Simulation of a Fair Queueing Algorithm. *Computer Communications Review*, 19.4:1–12, September 1989. SIGCOMM "89 Conference Proceedings.

[5] E. F. Gehringer, J. Abullarde, and M. H. Gulyn. A Survey of Commercial Parallel Processors. *Computer Architecture News*, 16.4:75–107, September 1988.

[6] M. G. Hluchyj. Queueing in High-Performance Packet Switching. *IEEE Journal on Selected Areas in Communications*, 6.9:1587–1597, December 1988.

[7] K. H. Horton. *Multicomputer Interconnection Using Word-Parallel Shift-Register Ring Networks.* Ph.d. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1984. also Technical Report No. UIUCDCS–R–84–1164.

[8] A. Huang and S. Knauer. Starlite: A Wideband Digital Awitch. In *Proceedings of the IEEE Globecom "84*, pages 121–125, 1984.

[9] J. Y. Hui and E. Arthurs. A Broadband Packet Switch for Integrated Packet Transport. *IEEE Journal on Selected Areas in Communications*, 5.8:1264–1273, October 1987.

[10] A. E. Kamal and V. C. Hamacher. Approximate Analysis of Non-exhaustive Multiserver Polling Systems with Applications to Local Area Networks. *Computer Networks and ISDN Systems*, 17, 1989.

[11] M. J. Karol. Optical Interconnection Using ShuffleNet Multihop Networks In Multi-Connected Ring Topologies. *Computer Communications Review*, 18.4:25–34, August 1988. SIGCOMM "88 Conference Proceedings.

[12] S. Keshav. Congestion Control in High Speed Networks, February 26 1990. presentation on HRR at the XUNET Student Meeting.

[13] G. Kimura and Y. Takahashi. Diffusion Approximation for a Token Ring System with Nonexhaustive Service. *IEEE Journal on Selected Areas in Communications*, 4.6:794–801, September 1986.

[14] T. T. Lee. Nonblocking Copy Networks for Multicast Packet Switching. *IEEE Journal on Selected Areas in Communications*, 6.9:1455–1467, December 1988.

[15] R. I. MacDonald. Terminology for Photonic Matrix Switches. *IEEE Journal on Selected Areas in Communications*, 6.7:1141–1151, August 1988.

[16] N. F. Maxemchuck. Routing in the Manhattan Street Network. Technical Report 11382-850905-01TM, AT&T Bell Laboratories, Murray Hill, NJ, September 1985.

[17] K. Murakami, A. Fukuda, T. Sueyoshi, and S. Tomita. An Overview of the Kyushu University Reconfigurable Parallel Processor. *Computer Architecture News*, 16.4:130–137, September 1988.

[18] Y. Ofek. Design and Analysis of a Fiber-Optic Hypergraph Network. Technical Report UIUCDCS–R–87–1343, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1987. also PhD Thesis.

[19] I. Page and J. Niehaus. The Flex Architecture, A High Speed Graphics Processor. *Computer Architecture News*, 16.4:117–129, September 1988.

[20] J. W. M. Pang and R. W. Donaldson. Approximate Delay Analysis and Results for Asymmetric Token-Passing and Polling Networks. *IEEE Journal on Selected Areas in Communications*, 4.6:783–793, September 1986.

[21] J. W. Reedy. The TDM Ring — A Fiber Optic Transport System for Campus or Metropolitan Networks. *IEEE Journal on Selected Areas in Communications*, 4.9:1474–1483, December 1986.

[22] J. Sauer. An Optoelectronic Multi-Gb/s Packet Switching Network, February 1989. Seminar notes.

[23] H. Takagi. Queuing Analysis of Polling Models. *ACM Computing Surveys*, 20.1:5–28, March 1988.

[24] T. Takine, Y. Takahashi, and T. Hasegawa. Performance Analysis of a Polling System with Single Buffers and Its Application to Interconnected Networks. *IEEE Journal on Selected Areas in Communications*, 4.6:802–812, September 1986.

[25] S. K. Tewksbury and L. A. Hornak. Communication Network Issues and High-Density Interconnects in Large-Scale Distributed Computing Systems. *IEEE Journal on Selected Areas in Communications*, 6.3:587–609, April 1988.