Chapter 6

Isochronet Switch Design and Implementation

6.1 Introduction

6.1.1 The Problem

The problem addressed in this chapter is to design and to implement an electronic Isochronet switch (Isoswitch) prototype. The switch must operate at giga bit per second rates and use only simple off-the-shelf components.

6.1.2 Main Results

Electronic switch design and implementation. An electronic Isoswitch has been designed and implemented. It has four input and four output ports each operating at 1 Gb/s. The design addressed the following challenges:

- Scalability with respect to number of channels and link speeds;
- Use of simple off-the-shelf components;
- Ease of integration with existing hardware;
- Configuration overheads do not disturb or slow on-going transmissions.

Interface card to a SPARC 1 machine. An interface card between the Isoswitch and a SPARC 1 machine has been built. The card has a nominal throughput of 22 Mb/s. It signals the beginning of bands to the SPARC processor and can thus be used to implement LTM and SPS.

Optical switch design. This chapter also proposes a preliminary design of an all-optical switch implementation. The implementation has a few advantages:

- No conversion of en-route signals to the electronic domain, which makes possible implementations potentially at hundreds of terabits per second.
- Fewer wavelengths are necessary than in pure WDM.
- RDMA- can be implemented guaranteeing no loss for priority traffic.

6.1.3 Chapter Organization

The reminder of this chapter is organized as follows. Section 6.2 discusses the main issues in building high-speed switches. Section 6.3 describes the electronic Isochronet switch design and implementation. Section 6.4 describes the design and implementation of an interface card between the switch and an attached workstation. Section 6.5 overviews the efficiency, complexity, and scalability of the switch. Section 6.6 evaluates the compatibility of the switch with current network devices. Section 6.7 identifies the main novel services the switch offers. Section 6.8 describes a potential all-optical Isochronet switch design. Section 6.9 overview future work. Finally, Section 6.10 summarizes.

6.2 High-speed Switching

Judging by the pace at which current optical transmission link technology is evolving, future HSNs will switch gigabits or even terabits of information per second. Besides being efficient, an adequate switching mechanism for HSNs must maximize bandwidth use while providing the necessary QoS that integrated data, voice, and video applications need.

Switching efficiency. Current switching techniques rely on the fact that processing efficiency significantly prevails over transmission efficiency. Unfortunately, the scenario is likely to change when HSN link speeds reach hundreds of gigabits or even terabits per second because it may become difficult to process frame headers or synchronize sources and destinations at these speeds.

QoS guarantees. HSN switches must be able to support a variety of QoS needs in terms of maximum end-to-end delay, jitter, loss, and bandwidth allocation. These parameters must be tunable by sources.

Scalability in speed. Link speeds have been increasing very rapidly in recent years and are expected to continue the trend with advances in optical transmissions. It is thus advisable not to base the switch design on peculiarity of specific transmission rates.

Scalability in the number of ports. Scalability in the number of ports has always been an issue in switch design. Network systems are continuously growing and switches must be scaled accordingly.

Implementation complexity and cost. The provision of multimedia services to a large market will significantly increase the complexity and size of HSNs. If switches are complex, their maintenance and cost will affect the number of end users supported and cost of

services provided. It is thus desirable to employ simple components (including switches) in the network.

Interface complexity and cost. The engineering of the switch must not complicate the engineering of its interface. The simpler the interface, the more services can be directly connected to the switch at cheap cost. Additionally, current services may be easier to integrated if the interface is kept simple.

Compatibility with current network devices. Current network devices will not cease to exist in future integrated HSNs. Slower speed technology will coexist with highs-speed technology, especially during transition phases. The switching technology must be compatible with existing devices to assure wide acceptance.

Resource multiplexing. HSN must be able to multiplex resources when QoS can be traded for cheaper services.

Minimal re-configuration overhead. Switch re-confirmation must be performed without disturbing on-going transmissions.

Isochronets are a strong candidate architecture to accommodate these features. There is no frame processing in the Isoswitch, which has a few advantages:

- The switching function is a simple mapping from incoming links into outgoing links, based on the current enabled trees and contention resolution policy among contending links within the same tree. These functions are realized completely off-line with transmissions. This makes the design efficient in terms of internal delay, scaleable with respect to link speeds, and simple to implement.

- Multiple frame structures can be supported without adaptation because headers are not processed during switching. This makes the Isoswitch easy to integrate with existing network devices and protocols.
- The interconnection fabric between input and output ports is not controlled by any frame header field, and thus any of the current modular interconnection technologies [Adams et al. 87, Broomell and Heath 83, Feng 81] can be used to implement the fabric, controlled by the slow periodic changes in tree configurations. The use of such technologies makes it possible for Isoswitches to scale with the number of ports.

Switch control functions can be accomplished through one simple mechanism: band allocation. Because of this:

- The interface to the switch has to provide two simple functions: transfer of frames to and from the network, and provision of signals to attached nodes informing when new trees become available. This simplifies the interface implementation.
- The necessary configuration information must include only current tree configuration together with priority among contending sources. All this information can be computed off-line (using a general purpose machine) and downloaded into the switch periodically, which then functions based on the new configuration. The configuration overhead is thus reduced to downloading simple configuration tables.

Additionally, the band allocation mechanism can be tuned to satisfy a variety of QoS demands. On one hand, strict guaranteed QoS in terms of delay, jitter, and loss can be

provided by priority bands. On the other hand, resource multiplexing can be provided by contention bands. By tuning priority and contention bands, QoS can be compromised for multiplexing in many different ways.

6.3 Isochronet Electronic Switch Design

This section describes the electronic RDMA+ Isoswitch design. Switching in the Isoswitch consists of two functions. The first is to configure the nodes periodically to form respective routing trees. The second is to select one of the contending sources when contention happens. The following sections take a bottom-up view to the switch design.

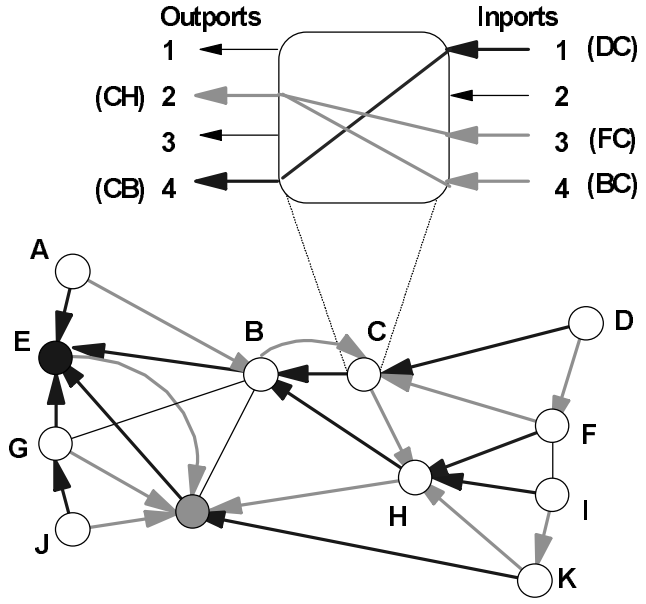


Figure 6.1: Tree allocation and node configuration

6.3.1 Configuration Table

A typical tree configuration is depicted in Figure 6.1. One of the nodes in the tree is shown with its input and output port connections necessary to accomplish the desired tree

configuration for two non-interfering routing trees (depicted using different shades of gray). The connections can be described by maintaining a data structure at each output port that lists all the input ports connected to it. In addition, the data structure should identify which of the connected inputs (if any) has priority. This information is captured in the Configuration Tables (CTs) at each node, described in Figure 6.2.

For the reminder of this section, it is assumed that each node has n input ports (inports) and m output ports (outports). Each inport is identified by a number in the range [1,n]. Similarly, outport identifiers are in the range [1,m].

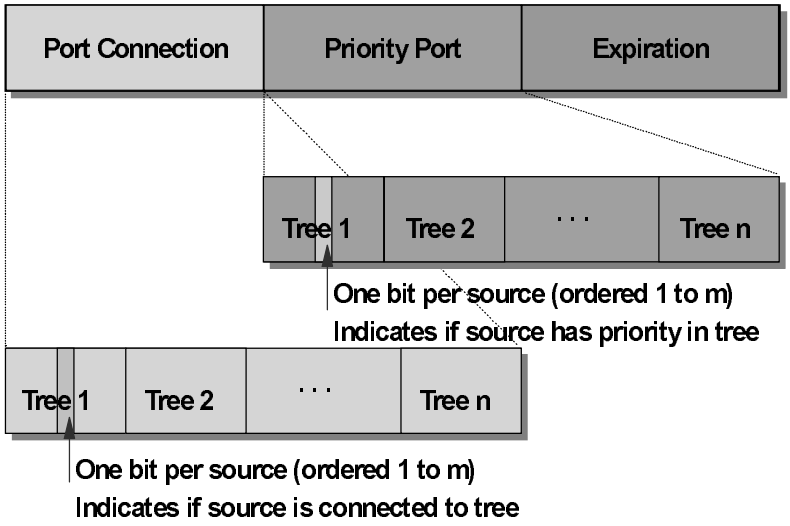


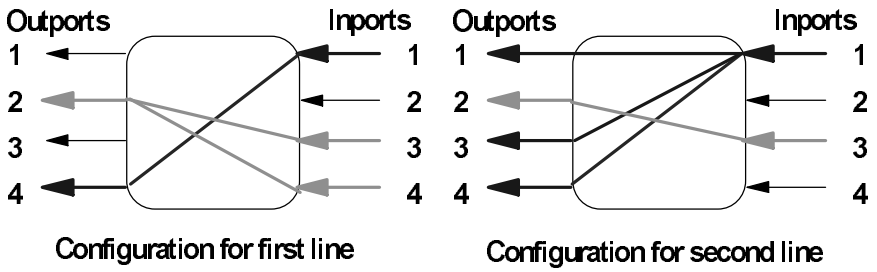
Figure 6.2: Configuration Table structure

The CT contains one line for each possible band in the cycle. Each line contains three fields: Port Connection, Priority Port, and Expiration. The Port Connection field describes the configurations of the trees that transverse the node. It contains a sequence of m binary words (one per outport) of size n bit (one per inport). Bit i ($1 \le i \le n$) in word j ($1 \le j \le m$) is 1 if and only if input i is connected to output j.

The priority ports field is again a sequence of *m* *n*-bit words. Bit *i* in word *j* is 1 if and only if *i* has priority to *j* during the current band. Notice that only one source can have priority to a particular output port within a band. This means that only one of the *n* bits can be 1 for each word.

The expiration field is a binary number that indicates how many clock ticks the current band should last.

Figure 6.3 illustrates the CT at a particular network node.



Port Connection	Priority Port	Expiration
…	…	…
0000 0011 0000 1000	0000 0010 0000 0000	100011110001
1000 0010 1000 1000	0000 0000 0000 0000	000111101100
…	…	…

Figure 6.3: Configuration Table in a particular network node

The CT is implemented as a RAM memory.

6.3.2 Arbitration Logic

The CT provides all information that is necessary to decide which input is granted access to the respective output at any given time. The Arbitration Logic (AL) is responsible for reaching such decisions. It is implemented as a combinational circuit.

Algorithm 6.1 is an abstraction of the AL. In Step 1, *Grant* is initialized, assuming that no input will be granted access to any output port. In Step 2, each output is evaluated to decide which input will be connected to it. Step 2.1 checks to see if there is a busy input that has priority to the output. If so, that input is granted access to the output. In Step 2.2, if no input has priority, there are two possibilities. In the first (Steps 2.2.1 and 2.2.2), there are many busy input ports connected to the output port being analyzed. One of them is picked randomly and granted access. In the second, no busy input is connected to the output and consequently the output is kept idle. This algorithm is repeated periodically to resolve input to output connectivity.

Algorithm 6.1: *Let n and m be the number of input and output ports, respectively. Let Con*[*i,j*] *be 1 if and only if input i is connected to output j (1 ≤ i ≤ n, 1 ≤ j ≤ m). Let Pri*[*i,j*] *be 1 if and only if input i has priority to output j. Let Busy*[*i*] *be 1 if and only if input i is busy. Let Grant*[*i,j*] *be 1 if and only if input i is granted access to output j. This algorithm computes Grant*[*i,j*].

1. For all $1 \le i \le n$ and $1 \le j \le m$, make *Grant*[*i,j*] = 0.
2. For all $1 \le j \le m$ do:
    - 2.1. If *Pri*[*i,j*] = 1 and *Busy*[*i*] = 1 for some *i* then *Grant*[*i,j*] = 1.
    - 2.2. If *Pri*[*i,j*] = 0 for all *i* then
        - 2.2.1. Let *K* be the set of all *i* such that *Busy*[*i*] = 1 and *Con*[*i,j*] = 1.
        - 2.2.2. If *K* is not empty, choose a random *k* in *K* and make *Grant*[*k,j*] = 1.

6.3.3 Switch Engine

The switch consists of the following modules: Line Cards, Switching Fabric, and Control Unit. The Control Unit receive configuration data from a Host machine directly connected to it. These components are depicted in Figure 6.4.

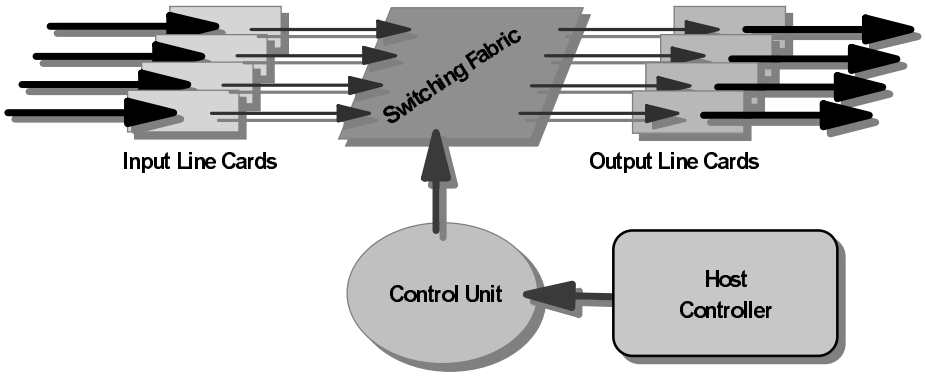


Figure 6.4: Switch organization

The Host machine computes band allocation information off-line with respect to data transmissions. Sporadically, it stores the new configurations in the CT inside the Control Unit. The Control Unit contains the AL that, based on the information in the CT, configures the Switching Fabric to provide the desired input and output connectivity. The Input Line Card receives bit-serial data from the network trunks and converts them into bit-parallel words that are routed to the proper output ports by the Switching Fabric. Similarly, the Output Line Cards converts bit-parallel words into bit-serial streams to be transmitted through the outgoing network trunks.

The Host machines in the network are connected by a traditional network such as a point-to-point modem connection, an Ethernet, or an Internet, depending on how far apart they are. They use these slow speed channels to exchange configuration and control information. This information is used to allocate and synchronize bands. The separation of

the high-speed links from the control channels makes the network more reliable and easy to control. For example, loss of band synchronization is easier to detect and communicate using a separate control channel.

The following presents a more detailed look at each component.

Input Line Cards. Line cards are responsible for media conversion (electronic/optical), signal (parallel/serial) generation, and buffering. Figure 6.5 depicts the Input Line Cards. The Isoswitch uses 40-bit words internally.

Figure 6.5: Input Card

Input queue buffering is performed after the serial to parallel conversion. In Isochronets, input or output queueing achieve the same performance. This is not usually the case in normal switching technologies due to the "Head of Line" (HoL) blocking effect [Murakami 91]. HoL blocking happens, for example, in PS technology with input queueing of packets. When the HoL is addressed to an output port that is busy, it must wait until the output becomes idle. Other packets that are behind the HoL might be destined to idle output ports, but are effectively blocked by the HoL, diminishing the potential throughput of the switch. The solution to this problem is to employ output queueing in which input packets are sorted according to their destination and placed in an output queue associated with the corresponding output port. Output queueing implementation is complex because it involves switching at n (where n is the number of input ports) times the rate of the ports. In Isochronets, the HoL effect does not exist because all packets

queued at a given input port are being routed through the same routing tree and thus seek the same output port. Because of that, it can employ the less complicated input buffering solution without incurring loss in switch throughput.

Switching Fabric. The Switching Fabric provides the connectivity between the input and Output Line Cards. The structure used in the Isoswitch is depicted in Figure 6.6. It connects each output port to all the input ports. The building blocks for the Switching Fabric are multiplexers, one per output port. Each multiplexer is connected to all input lines. By using the multiplexer selection lines, the appropriate input/output connectivity is achieved based on the current switch configuration supplied by the AL. The implemented Isoswitch has four input and four output ports.

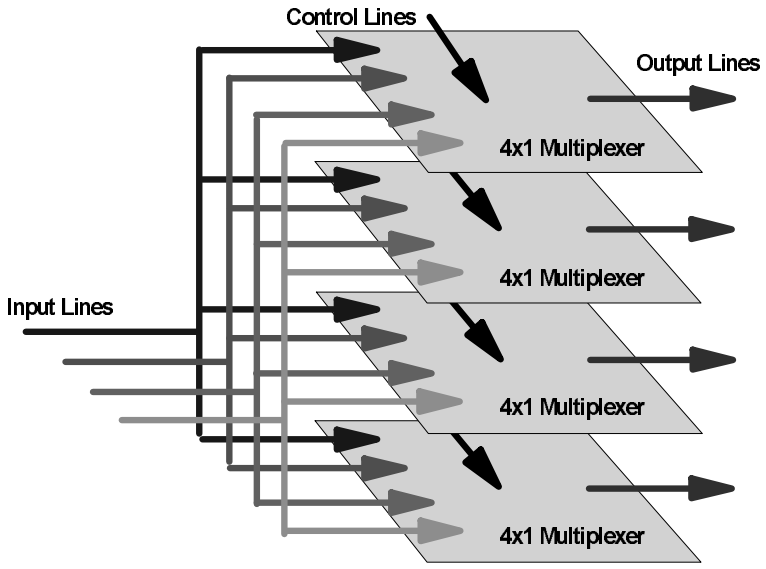


Figure 6.6: Switching Fabric

The Switching Fabric architecture is independent of Isochronets operations, and can be implemented using any available interconnection architecture. In particular, architec-

tures that do not necessarily dependent on the contents of internal frame headers for switching decisions are best suitable for RDMA.

Multi-stage interconnection networks [Adams et al. 87, Broomell and Heath 83, Feng 81] are one such classes of architectures. They reduce the internal complexity of the fabric at the cost of extra stages (and thus added end-to-end delay). Such interconnection can replace the one in Figure 6.6, and can be controlled directly from the Control Unit. The AL decides how to control each stage in the interconnection based on the contents of the CT and the busy lines.

In the proposed design, a complete configuration is used to simplify the logic, since the number of ports is small and consequently the internal fabric complexity is small.

Control Unit. Configuration and arbitration are the main functions of the Control Unit, depicted in Figure 6.7. The CT (discussed in Section 6.3.1) is implemented in the Configuration Memory (CM), explained in more details later. The AL (discussed in Section 6.3.2) is implemented as a combinational circuit.

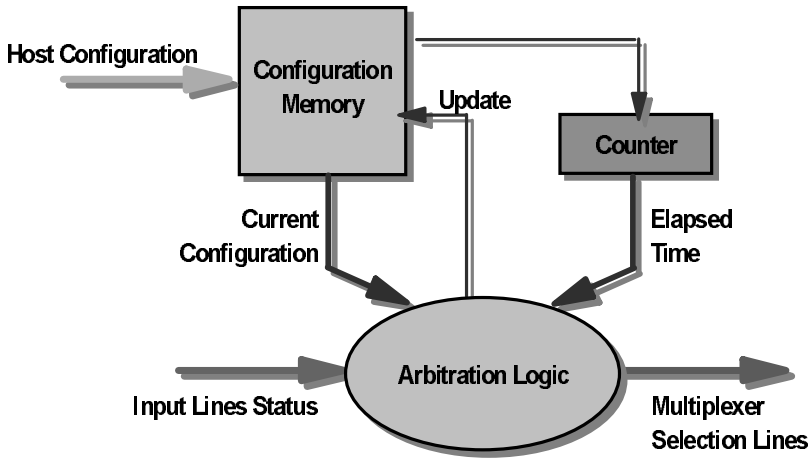


Figure 6.7: Control Unit

New configurations are loaded from the Host machine into the CM. It works similarly to a main memory in a computer. The CM contains switch configurations that are fetched in sequence, much like instructions are accessed in computers. When a new configuration is fetched, the duration for the configuration in the Expiration field is loaded in the Counter register. At each clock tick, the counter is decremented to reflect the elapsed time. When the Counter is 0, it is time to fetch a new configuration from the CM. The sequence of all configurations form the Isochronet cycle that is repeated forever.

During each configuration, the Port Connection and Priority Port fields in the current CT entry along with input line status information on which lines are busy are supplied to the AL that decides the configuration of the multiplexers in the Switching Fabric. The correct input and output connectivity is achieved by selecting the proper multiplexer control.

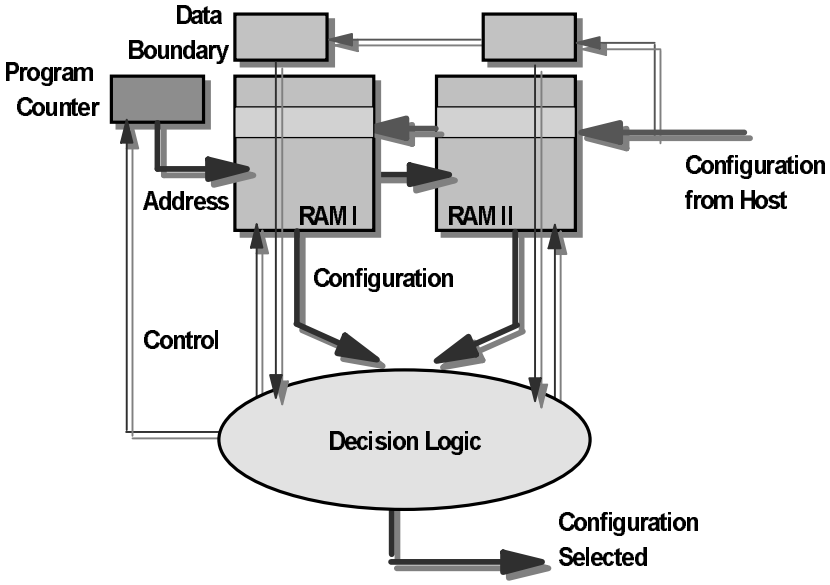


Figure 6.8: Configuration Memory

Figure 6.8 illustrates the CM. Two tandem memories (RAMs) are used to store CTs. Only one of the RAMs is being used by the AL at any given time. When a new CT needs

to be loaded from the Host machine, the other RAM is used to store it. In this manner, a new CT may be loaded while the switch is still operating with the old CT. When the current cycle is finished, the switch may operate using the new CT stored in the other RAM.

The Program Counter points always to the address of the current configuration. The RAM containing the current CT is addressed by the Program Counter and supplies the configuration to the Decision Logic that selects the valid RAM. The Program Counter is incremented when the current entry of the CT expires, to point to the next CT. The Data Boundary register contains the address of the last valid configuration. The Program Counter is reset when it reaches this address to restart the cycle.

Meanwhile, the Host can directly address the other RAM to store the next CT concurrently. It also stores the boundary information in the respective Data Boundary register. Once finished, it signals the Decision Logic so that the latter will begin using the new CT after the current Isochronet cycle is finished.

An interesting characteristic of the current implementation is that it uses a special kind of RAM. The RAM is asynchronous and always outputs the current word pointed by the Program Counter. When the Program Counter changes, the time to fetch the next word and stabilize it in the RAM output lines is less than 40 ns in the current implementation. The Program Counter can change and the new configuration used by the AL in the same clock tick. In other words, the change in configuration does not take any overhead because it can occur in parallel within the delay of the clock tick used for arbitration. Other components, like the AL delay, define the size of the clock tick rather than the time to fetch words in the RAM.

Output Line Cards. Figure 6.9 depicts the Output Line Card. Besides performing the inverse function of the Input Line Card, it has a delay element before the parallel to serial bit conversion module. The main function of the Delay Module is to make the whole link propagation delay a multiple of the cycle period. In this manner, the cycles at each node begin a the same time and band synchronization becomes simple. More details on the protocols that use this feature can be found in Chapter 4.

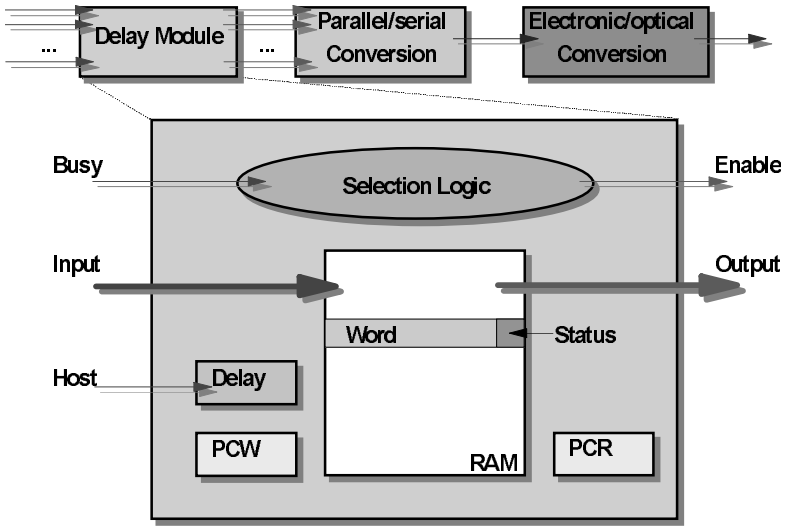


Figure 6.9: Output Card

The main function of the Delay Module is to delay transmissions according to a Host specified amount, between 0 and the cycle period. The Host monitors each link end-to-end propagation delay and then decides which value is to be set in the delay module. Such delay (in clock ticks) is placed in the Delay register. The dual port RAM can be written and read concurrently. The Program Counter Write (PCW) is used to write words to the RAM, while the Program Counter Read (PCR) is used to read words from the RAM. At each clock tick, a word is written into the RAM. If the input was busy, the word written is valid and the status bit is marked with 1. If the input was idle, the word is not valid and its

status is marked with 0. Words are then fetched using PCR. If the word fetched from PCR has status 1, it is transmitted. If it has status 0, it is not transmitted. To achieve the delay effect, PCW is initially equal to the contents of Delay while PCR is equal to 0. In this way, the number of words stored in Delay elapse before the first transmission occurs. This assures the required delay.

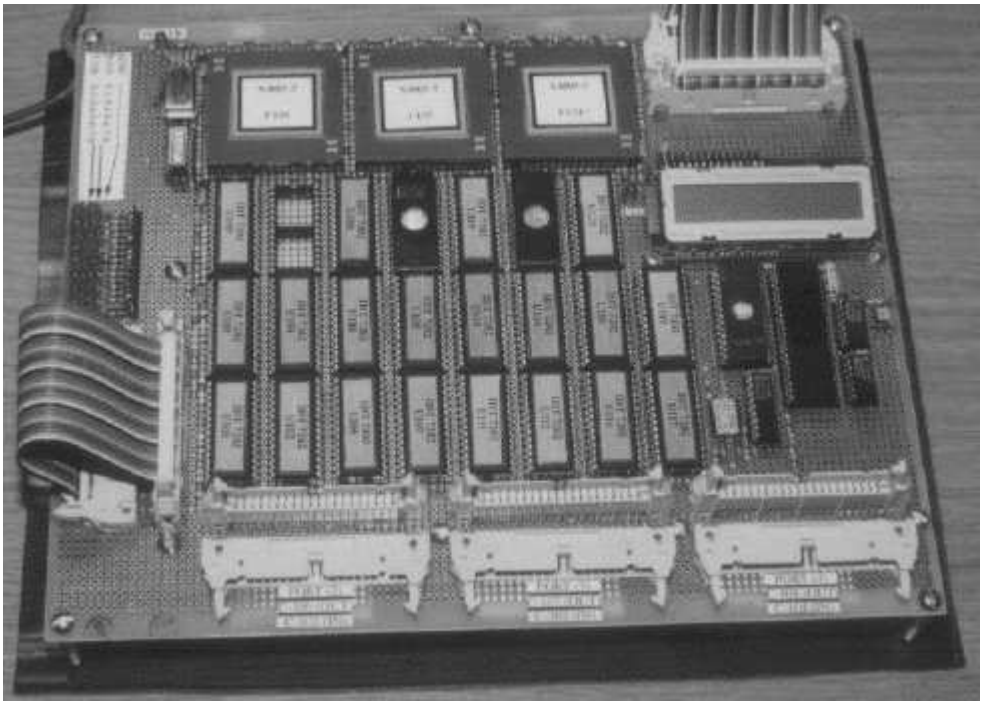


Figure 6.10: Isoswitch implementation

6.3.4 Operational Characteristics

The Isoswitch was implemented using the 4005H family Xilinx Logic Cell Arrays (LCAs). A picture of the prototype is shown in Figure 6.10. LCAs are relatively slow, but were chosen due to the ease of changing the implemented design. This feature is essential for a prototype design and implementation. To get a feel of how slow the adopted 4005H LCAs operate, the implementation of a single D flip-flop [Roth 92] in the chip can achieve

a throughput of only 38.5 MHz and the minimal latency through a single combinational gate is 5 ns. Nevertheless, even using such slow components, it was possible to accomplish the goal of implementing a 1 Gb/s per port Isoswitch due to the simplicity in Isochronet operations.

The Isoswitch Control Unit operates at 3.125 MHz, that is, it reaches more than 3 million switching decisions per second. Internally, the data bus is 40-bit wide. In the absence of contention or queueing delays, it takes 320 ns from the time a frame arrives at the switch until it is selected for switching. At each clock tick, 8 40-bit data words are transferred through the switch. The nominal rate per output port is thus 3.125×8×40 Mb/s, or 1 Gb/s. Additionally, each data word takes 40 ns to cross the Switching Fabric from the input to the output port.

6.4 Isochronet Interfaces

The Isoswitch interface must provide, as basic functionality, signaling of network status and basic means for data transfer. The status signaling embody information such as current enabled destinations, priority sources, etc. This is a typical example of a loosely-synchronous network interface discussed in Chapter 5.

Figure 6.11 depicts the interface card to the Isoswitch. Buffers are placed in the interface for data transmission and reception. The objective of the Transmission Buffer is to gather data from the slower Host machine so that it can be sent at full speed through the switch. The Reception Buffer is used to store data received from the high-speed switch until the Host machine can access it.

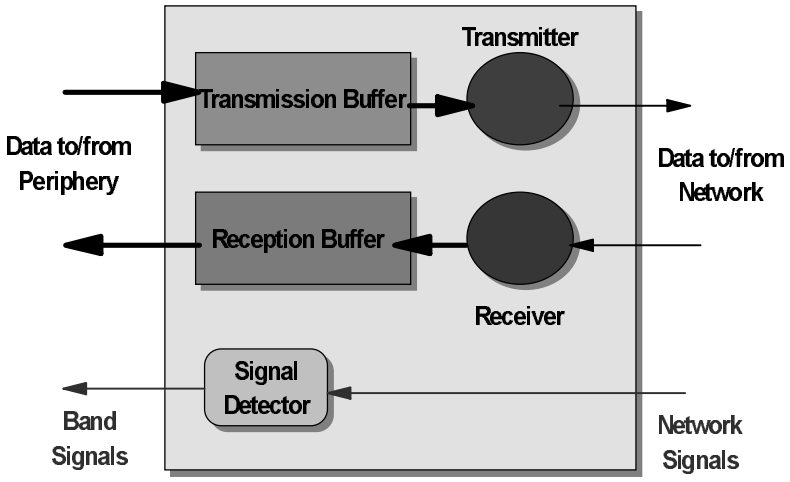


Figure 6.11: Isochronet interface to an end machine

Figure 6.12: Interconnection between the Isoswitch and the Host machine

The card receives signaling information from the switch when the following events occur: cycle begins, bands begin, and data reception. These signals are forwarded as interrupts to the Host machine containing the interface card. A status register in the card can be read by the Host to retrieve the details of what event spawned the interrupt. Alterna-

tively, the status register may be used to check the same events without enabling the interrupts. In this mode, the Host can poll the status registers to see when events occur. Finally, a control register in the card can be used by the Host to enable or disable some of the events or interrupts.

This line card was implemented as an interface between the switch and Sun SPARC [Frank and Lyle 90] machines. A picture showing the connection of the switch to the SPARC containing the card is shown in Figure 6.12. The card can send data to the switch at the peak 1 Gb/s rate once the Transmission Buffers contains at least 8 words. Similarly, data is received from the switch at 1 Gb/s. Nevertheless, the speed at which the SPARC can fill the card buffers is dependent on the protocol used by its bus (the SBus). Measures of the SBus transfer rate between processor registers and the implemented card showed a maximum throughput of 22 Mb/s.

6.5 Efficiency, Complexity, and Scalability

This section evaluates the switch efficiency and complexity, and how these values scale with the number of ports and the link speeds. These measures are evaluated for the Control Unit logic and the Switching Fabric logic.

Control Unit Logic. The Isoswitch switching functions is performed by the Control Unit. In order to find out the effect of increasing the number of switch ports it thus suffices to evaluate the AL complexity and latency in Algorithm 6.1. Step 1 is not really performed in the implemented hardware because the flip-flops that implement *Grant* have default value 0. The operations in Step 2 are performed in parallel, by m similar combinational logic circuits. Each of the circuits has as input a portion of *Pri* of size n plus *Busy*

that is of size n as well (see steps 2.1 and 2.2). Thus, each of the m parallel circuits has complexity $O(n)$. The overall AL complexity is thus $O(nm)$. As for latency, since m parallel combinational circuits are processing n entries, and since each instruction has linear latency, the total latency is $O(n)$. Notice that both complexity and latency are also optimal. Firstly, the switch has n input ports (and thus arbitration needs to at least read all n input status). Secondly, it has m output ports (and thus must decide m problems of size n).

The whole Control Unit (including the tandem memory components) is implemented in a single LCA. It occupies 92% of the LCA logic and 45% of its pins.

It is important to notice that the Isoswitch AL complexity and latency depend only on the values for n and m. They do not depend on other factors normally found in other switching architectures for HSNs such as processing per frame header. Processing and link transmission per frame affect traditional switch arbitration complexity as follows. If the link speed in b (in bits per second), frames have size f (in bits) and each frame takes c (in seconds) to have their headers processed, the switch needs computational power to process each input port at $(b/f)c$ frames per second. For example, an ATM switch with 1 Gb/s lines needs to be able to process about 2.36 million cells in one second or one frame every 424 ns per port. Since in the Isoswitch $c=0$, the processing component does not affect the switch complexity.

Nonetheless, the link speed does affect the Control Unit logic speed in the Isoswitch. This dependency may be overcome due to the simplicity in Isoswitch operations, which makes it possible to run it at very high clock rates. The AL must make decisions at a rate of b/f per outport. Since each Isoswitch outport is processed independently, each of these

circuits must have a maximum latency of *f/b*. For example, at 1 Gb/s and 53 bit frames, the latency must be 424 ns. Notice that such constraint is not a problem, since only simple arbitration is to be performed (no processing dependent on frame headers). For example, the arbitration latency for the implemented Isoswitch is 320 ns.

The number of ports *n* also affects the control logic speeds. When the number of ports is increased, the Control Unit needs to increase its arbitration decisions linearly. In reality both the constraints on link speed and on number of ports can be overcome by the following techniques applicable in RDMA.

In the first technique, each periphery source must send always *w* consecutive frames. Since arbitration is now performed for the whole block of *w* frames, its latency may be *wf/b* (that is, *w* times bigger than originally). Such technique cannot be used in any switching technique that needs to process frames because the switching decision for each of the *w* individual frames may be different. Furthermore, this technique does not affect the frame size (that is, it is different of creating a new network frame size). For example, if 8 ATM cells are always sent through the Isoswitch, the tolerated latency is 2.56 μs using this technique.

In the second technique, the internal data bus in the switch is increased. By increasing the bus size *d* times, the Control Unit can operate *d* times slower. Such technique also relies heavily on the fact that no frame-dependent processing occurs in the Isoswitch.

Switching Fabric Logic. The implementation complexity is reduced in Isochronets since no processing occurs in the fabric. The complexity in terms of the number of input to output lines is independent of the RDMA technique. Care must be taken to avoid a large

number of connections in the fabric. Many existing solutions to reduce the number of lines can be adopted in the Isoswitch. For example, one may employ multi-stage interconnection networks, as discussed in Section 6.3.3.

The Switching Fabric implemented contains *m* tandem multiplexers with *n* inputs each (see 6.3.3). Thus, its complexity is *O(nm)* and its latency is *O(n)*. The Switching Fabric implementation occupies only 43% of the LCA chip logic, while is uses 92% of the pins. It is implemented in two LCAs. The total number of LCAs used in the switch is thus 3, 1 for the Control Unit and 2 for the Switching Fabric.

Another factor that increases the Isoswitch scalability is its modular design. Multiple switches may be interconnected to build an *Isohub*, that is, a node consisting of multiple Isoswitches, with higher number of ports. The interconnection of the multiple Isoswitches can be accomplished in many ways.

Finally, as mentioned in Section 6.3.3, the re-configuration overhead is absent in the Isoswitch due to the use of dual memory modules.

6.6 Compatibility with Current Network Devices

The end-to-end delay in interconnected HSNs will heavily depend on how efficiently component networks can operate. For example, when an Ethernet network is connected to an ATM network, a router must be placed at the interface between both networks. The router has to fragment (or assemble) Ethernet frames into (or from) ATM cells, allocate and free ATM virtual connections (circuit or path), prioritize multiplexed Ethernet traffic to deliver necessary QoS, etc. Such functions are not simple and it may very well be the

future bottleneck in interconnected HSNs. It is thus fundamental for HSN switches to minimize such routing functions.

Isoswitches can significantly simplify inter-operability among networks because they do not rely on any particular frame structure. Consequently, frames may be forwarded through the switch without adaptation. Other protocol operations can be supported through the Isoswitch directly, with no changes.

6.7 Switching Services

This section analyzes the Isoswitch as a black box and identifies the fundamental services that it offers. The emphasis is on services that can be provided in addition to traditional switching services.

The first new service is the provision of synchronization signals. The Isoswitch issues synchronization signals that identify the beginning of bands and cycles. These signals may be used by periphery nodes to synchronize their local clocks. As a result, global synchronization can be achieved by local exchange between network switches and attached periphery nodes.

Synchronization signals can additionally be used in the protocol stacks at the end nodes to provide synchronous services. A whole new synchronous protocol stack, detailed in Chapter 5, can be implemented using this feature.

These signals are novel when compared to traditional STM synchronization in many respects. First, the necessary accuracy of Isoswitch signals is much lower. For example, an 8 bit slot in a STM frame at 2.4 Gb/s transmission rate lasts 3.3 ns. Typical bands last between a few hundreds of nanoseconds up to a few scores of microseconds. Second, no

global network-wide synchronization is necessary among the periphery nodes. Third, the signals embody routing and QoS information that may be used by periphery nodes to schedule their activities. For example, these signals can be forwarded to applications that can schedule their activities to the signals.

The second new service is direct frame forwarding. That is, the Isoswitch does not rely on any particular frame structure to operate. On the contrary, any protocol frame may be directly forwarded through the switch without adaptation thus significantly simplifying work at periphery nodes.

The third new service is guaranteed (as opposed to statistical) QoS provision. Isoswitches can provide guarantees through priority bands. Once signaled at the beginning of a priority band, sources can transmit to the respective destinations and be assured that no loss or delay due to contention will occur in the switch.

The interface is simple, as explained in Section 6.4. The main functionality is frame forwarding, reception, and signal forwarding. All these functions can be accomplished with simple circuitry.

6.8 All-optical Implementation

This section describes the all-optical RDMA- Isochronet design. An all-optical realization of Isochronets must avoid buffering at intermediate switches. The design proposed in this section uses Wavelength Division Multiplexing (WDM) to allocate one wavelength for each band. Figure 6.13 illustrates how wavelengths are used in each band when there is only one tree per band. The figure displays three such trees atop the network topology. Notice that each tree is allocated to a different band because they all share link (F,E) in the

same direction. The idea is to use a different wavelength for each tree. In this case, three wavelengths cross link (F,E), one for each tree. Contention among trees is thus resolved by WDM.

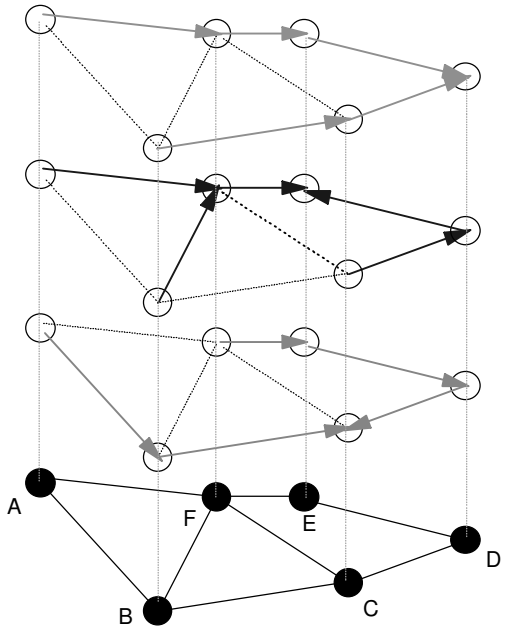


Figure 6.13: WDM allocation to routing trees

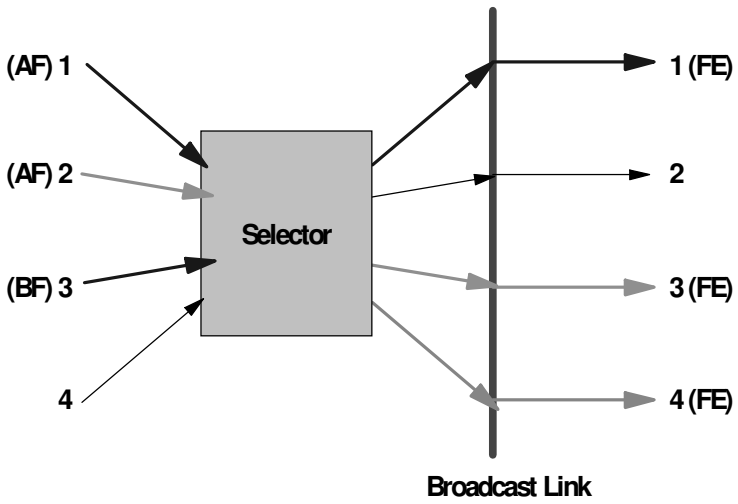


Figure 6.14: All-optical switch implementation: one tree per band

The architecture for Node E in Figure 6.13 is shown in Figure 6.14, where each input and output port is labeled with the corresponding link. Each wavelength is depicted using a different shade of gray. Incoming wavelengths are first fed into a Selector box (explained later) and then multiplexed through a single optical Broadcast Link (the interconnection fabric) connecting all source and destination links. At each output link, a slowly-tunable receiver picks the wavelength of the trees sharing the link. The receiver is directly connected to a slowly-tunable transmitter that regenerates the wavelength in its output link.

Contention in the all-optical implementation is resolved by discarding all but one of the contending frames. Contention occurs when two or more of the input links are sending frames using the same wavelength through the Broadcast Link. In this circumstance, one of the sources must be enabled while the others must be stopped, otherwise all data may be lost. This functionality is achieved through the Selection box, the only electronic component in this architecture, depicted in Figure 6.15. Its function is to detect incoming signals from the links and immediately grant access to one of them, shutting the others.

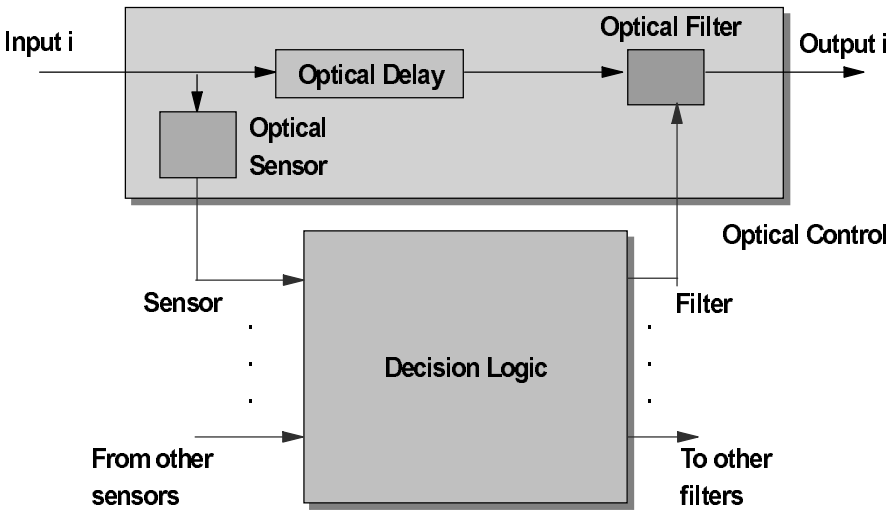


Figure 6.15: Selection box

Each input line is extended inside the selection box to increase the delay from the input to the output ports. This extension is the Optical Delay in the figure. At each entrance, an optical sensor detects incoming light. The idea is to only allow one input per wavelength into the Broadcast Link. The Decision Logic decides which of the inputs should be allowed to proceed and which should be shut. The decision is reached using the AL with inputs from the CT and sensors. The decision is input to the filter at the exit of the line, which passes or block light according to AL instructions. The size of the extension inside the selection box must be big enough to allow the AL reach its decision before the optical data reaches the end of the extension.

Priority bands are implemented as in the electronic Isochronet implementation, using time division inside the selection box. The CT contains the priority data information and the AL uses it when deciding which sources should proceed in each band. Priority bands exist during portions of a cycle, whereas contention bands are always “opened”.

Multiple trees per band are implemented by extending the architecture in Figure 6.14 into multiple Broadcast Links. The idea is to propagate different routing trees in the same Broadcast Link using separate wavelengths, but still be able to reuse wavelengths in separate Broadcast Links. In this manner, the total number of wavelengths necessary is reduced. Notice that trees in the same band do not share any link in the same direction, so they will not interfere at any link if they use the same wavelength.

The implementation of the multiple Broadcast Link scheme works as follows. Each input link is connected to all the Broadcast Links, but optical filters are placed at the interface between each input link and each Broadcast Link to pass or block wavelengths. The

filters are set so that, after the filtering phase, no two trees using the same wavelength are broadcast through the same Broadcast Link. After this phase, operations within each Broadcast Link work as explained for the single Broadcast Link case.

The optical Isochronet implementation has many advantages when compared with traditional WDM. First, a small number of wavelengths (at most n, where n is the number of switches in the network) are needed. Second, no pre-allocation of wavelength or frame processing is necessary for communication. Most recent schemes (see [Ramaswami 93] for a survey of such schemes) either need to provide a special control channel for the reservation of wavelength prior to communication or need to process frames. The reservation schemes suffer drawbacks such as round-trip allocation delay, necessity for rapidly-tunable receivers/transmitters, and dedicated bandwidth. The schemes that switch frames incur the added delay for media conversion and frame processing. Third, the implementation described can use slowly tunable transmitters and receivers since they only need to tune when nodes or links fail, which occurs at much slower rates than the speed of incoming frames.

It is important to notice that all bands are opened all the time, avoiding synchronization of bands.

Frame loss may occur in RDMA- during contention bands. It would be interesting for an all-optical implementation to implement slots in each link to decrease the probability of frame loss. In the following, the frame-loss probability for the slotted implementation when arrivals are Poisson is analyzed. Also, an extension to the basic scheme to reduce the frame-loss probability is suggested.

Let us assume that the input rate to the system is between 0 and 1 (it is simple to extend the following analysis for generic input rates). Let $\frac{\lambda}{n}$ be the input rate ($0 \le \lambda \le 1$) of each input link to a particular switch, and n be the number of input links to the switch. The probability of no transmission from a source link during a slot is $1-\frac{\lambda}{n}$. Thus, the average successful transmission rate during a slot is $1-\left(1-\frac{1}{\lambda}\right)^n$ (that is, if at least one source transmits). As $n \to \infty$, the successful transmission rate becomes $1-e^{-\lambda}$. The expected success probability is $\frac{\left(1-e^{-\lambda}\right)}{\lambda}$. Finally, the expected loss probability is $1-\frac{\left(1-e^{-\lambda}\right)}{\lambda}$. When $\lambda \to 1$ (loaded system), the expected loss is e^{-1} (less than 37%).

It will be shown in Chapter 7 that the loss rate is bound to 50% when the source stream transmissions are not slotted.

It is possible to improve the performance of this scheme. Multiple copies of the same frame may be sent, thus decreasing the loss probability for the frame. For example, assuming that losses are uniformly distributed among the contending frames, each frame can be repeated m times, resulting in a maximum loss probability of e^{-m} per frame (that is, all m frame copies would have to be lost to lose the frame and, in addition, the input load in the system including the copies would be 1). Thus, m may be computed from the maximum loss rate r that can be tolerated in the system: $r \le e^{-m}$ so that $m \ge -\ln r$. For ex-

ample, $m = 4$ insures less than 2% loss rate when the system is heavily loaded and the number of input sources is big.

Notice that when m copies are generated, the maximum input load to the system before copying is at most $1/m$. Thus, $1 - (1/m)$ of the potential bandwidth is lost due to the copying process. For example, if $m = 4$, the loss in bandwidth is 75%. This may be reasonable in all-optical networks where the rates are in the order of scores of terabits per second.

To complete the design using the analysis above, a filter is placed at the traffic sources (before the traffic enters the network), which disturbs the input traffic frame inter-arrival times to the network and makes them exponentially distributed (thus generating a Poisson arrival process to the network). Each source sends m copies of the same frame, where m is computed from the tolerated loss rate.

One can observe that the same design used for the electronic design could have been used for the optical RDMA- design as well. That is, bandwidth could be time divided among trees in the optical design as well. The Selection box in Figure 6.15 would be necessary in the optical design to detect and resolve contention among frames. The Control Unit in Figure 6.7 could be used unchanged in the all-optical design. This design using time division is an alternative to the one proposed using wavelength division.

Both have advantages and disadvantages that should be evaluated according to the operational goals of the network. The time division version is cheaper to implement in terms of transmitters and receivers per node and interface cards; it only requires one receiver and one transmitter. Once a band is granted, the whole optical spectrum is available for

transmissions. The wavelength version does not require band synchronization. Additionally, the optical spectrum is so rich in bandwidth that it is unlikely that a single source will be able to generate alone such an amount of traffic. If this is the case, wavelength division may be more reasonable.

This chapter chose to elaborate on the wavelength version to show two different design possibilities.

6.9 Future Work

One of the most important extensions to this work is to pursue the implementation of an all-optical switch. The switch has the potential of running at terabit per second rates.

Another interesting experiment is to implement a few Isoswitches to test a full-size network. In a first stage the switches can be implemented as a hub, that is, all nodes in the same box. As a second stage, they could be implemented in a distributed fashion using the delay elements proposed in this chapter.

The study of efficient interfaces to the network is not complete. What is the ideal general purpose interface to Isochronets? The interface proposed in this work is simple and relies to the attached Host queueing and traffic scheduling responsibilities. One interesting extension would be to build a more complex interface where these responsibilities would be embedded in the interface card.

6.10 Summary

The fact that Isochronets reduces all network-layer functionality to the MAC layer significantly simplifies the implementation of its switch. Among other features in Iso-

chronets: (1) no frame header processing is required in the network, (2) there is no need for adaptation layers at the network interface, and (3) internetworking is simplified.

All these features guide the design of the Isoswitch. Because switching is independent of frame headers, both an electronic and an all-optical implementation of the Isoswitch are possible. This work developed both designs in details and described an electronic Isoswitch implementation.

The implemented Isoswitch has four input and output channels operating at 1 Gb/s. The design is modular and scaleable with respect to increases in the number of channels and transmission speeds. Inter-operability with other switching techniques is simplified. Finally, it offers novel services: (1) synchronous signaling, (2) no necessity for adaptation, and (3) guaranteed QoS provision.