11. Switching

When a node port wants to send frames to another node port, it places the destination address of the other port in the frame header and sends those frames into the fabric. The fabric receives, routes and delivers the frames based on the destination address. The fabric is responsible for selecting the path, or paths, used for the delivery of frames. The sending node port has no knowledge of paths, nor does it have any control over the actual path used.

The decision to do all frame routing based on the destination address in the frame header relieves the node port from having to maintain any kind of routing table. All the sending port needs to know is the address of the destination port.

Allowing the fabric to select the path used to route frames from the source node port to the destination node port allows a fabric to perform adaptive path selection based on traffic or congestion or select an alternate path in the event of a failure. The adaptive routing of frames is not visible to the sending or receiving ports except as differences in the paths may affect latency or other delivery characteristics.

Many designs can be used within a switch to route frames from one switch port to another. This chapter explores some techniques used by current Fibre Channel switches.

11.1 Switching Construct Characteristics

Within every Fibre Channel switch is a function that transports frames from an input port to an output port. The standards refer to this function as the switching construct.

The switching construct may be implemented in many different ways within Fibre Channel switches and there is no requirement that all switches within the fabric implement the same design. As long as all of the switches route frames appropriately the differences in implementation do not affect correct operation of the fabric.

Different switching construct implementations may exhibit different characteristics such as the aggregate bandwidth provided by the switching construct, the time required for a frame to be routed, the cost of implementing the switching function, whether traffic between one pair of ports blocks traffic between other ports, etc.

11.1.1 Circuit Switching vs. Frame Switching

Fibre Channel provides the capability to reserve bandwidth for the transmission of frames between ports. This capability is provided by Class-1 and is referred to as connection-oriented classes of service. When a connection-oriented class of service is used, a circuit is established between the ports prior to frame transmission. This ensures that the bandwidth is available for delivery of subsequent frames. The circuit lasts until the connection is removed.

Class-2 and Class-3 do not reserve bandwidth and are referred to as connectionless classes of service. In these classes of service, frames are routed on a frame-by-frame basis (this type of operation is frequently referred to as a frame-switching or packet switching service). No circuit is established and no resources are reserved for the delivery of frames.

Fibre Channel switches may support one or more classes of service. If a switch is to support Class-1, its switching construct must provide a circuit-switching capability. If it supports Class-2 or Class-3, its switching construct must provide a frame-switching capability. If the switch supports Class-1 and either Class-2 or Class-3, it must provide both; a circuit switching capability and a frame switching capability.

If a switch supports Class-1, the situation becomes somewhat more complicated. A Class-1 connection guarantees full link bandwidth to the connected node ports (in buffered Class-1 the amount of bandwidth guaranteed is determined by the link speed of the slower node port). This makes the ports busy to other traffic and limits communications to the two ports. In order to allow the ports outside communication during a Class-1 connection, an optional feature called intermix is defined. When supported, intermix allows connectionless frames (Class-2 or Class-3) to be sent or received while a Class-1 connection is in effect. The connectionless frames are extracted from, or intermixed onto, the Class-1 connection path subject to the availability of bandwidth.

Intermix impacts the switching construct design within the switch because the switch must now provide the ability to maintain the Class-1 connection and route the connectionless frames on the connection-oriented path. This may require two independent switching construct functions, one circuit-switching and the other frame-switching. Both capabilities must be available simultaneously because intermix frames are connectionless frames using the path of an established connection.

If a switch supports Class-1, it also must support this dual-mode capability in order to interject Class-F frames onto an existing Class-1 connection that is using an interswitch link. In this case, the traffic inserted into the Class-1 connection is the fabric's own internal traffic.

11.1.2 Blocking vs. Non-Blocking Switching

Within a switch, traffic between one pair of ports may or may not have an effect on traffic between other pairs of ports. If traffic between one pair of ports affects traffic between other pairs of ports, the switching construct is said to exhibit blocking characteristics (the traffic blocks other unrelated traffic). If traffic between one pair of ports does not prevent traffic between other port pairs, the switching construct is referred to as non-blocking. The duration of the blockage may be as short as a single frame in a connectionless class of service or an entire connection in a connection-oriented class of service.

The choice of switching construct design determines whether a given switch exhibits blocking characteristics. Providing a non-blocking design requires a switching construct function with sufficient bandwidth to support the worst-case traffic condition and as many internal routing paths as there are ports on the switch. For example, a 16-ported switch with 100 MB/sec. ports requires an aggregate bandwidth of 1,600 MB/sec and 16 internal routing paths to prevent blockage. Providing this level of capability may add to the cost of the switching function.

On the other hand, a switch implementation may choose to provide less bandwidth and fewer internal switching construct paths in order to save cost. An analysis might determine that typical traffic patterns use only 50% of the worst-case bandwidth and paths. In this case, the switch may implement a smaller and less expensive switching construct to reduce the cost of the switch. However, if the amount of traffic exceeds the capabilities of the switching construct, traffic will become blocked as the switching construct becomes congested.

When multiple switches are connected in a fabric, the design of the fabric itself can result in blocking behavior, even if the individual switches are non-blocking. This can occur if the interconnections between switches do not provide the necessary bandwidth or number of paths.

11.1.3 Internal Head-of-Line (HOL) Blocking

Head-of-line blocking within a switch occurs when a frame that can't currently be delivered blocks other frames that could be delivered. An example of an head-of-line blocking is shown in Figure 66.

In this example input A has a frame for output D followed by several frames for output C. However, the switch point associated with output port D is currently committed to the delivery of a series of frames from input B. As long as input B holds the switch point, input port A can't deliver the frame to output D and the frames to port C are blocked.

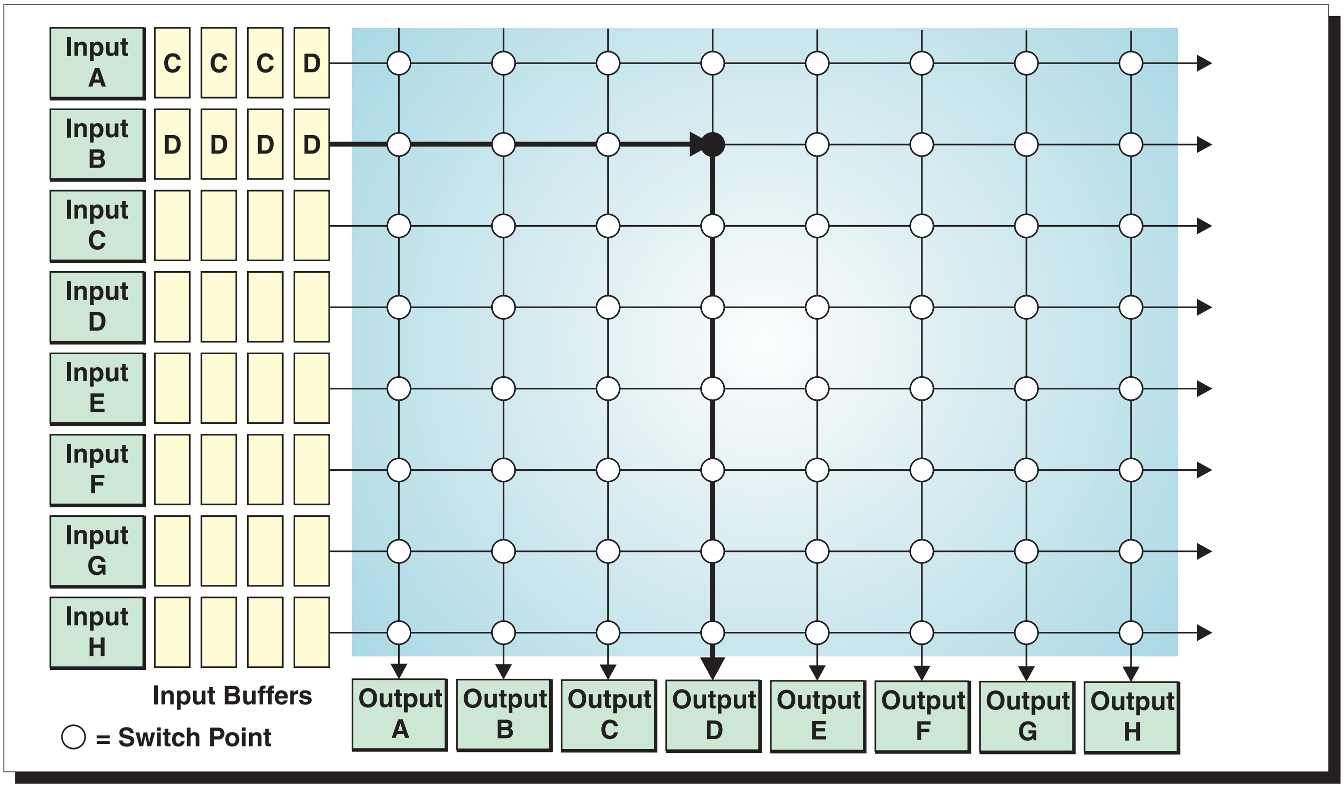

Figure 65. Head-of-Line (HOL) Blocking

This type of head-of-line blocking can be addressed by providing look-ahead capabilities in the receive buffer. If input A could look past the frame for D, it would see that there are frames for output C that can be delivered. Up to a point, the deeper the look-ahead, the less often frames will become blocked. Of course, providing this kind of look-ahead adds to the cost and complexity of the switch.

11.1.4 Cut-Through Routing

If a switch port waits until it has received a complete frame before attempting to forward the frame, it adds an entire frame's worth of latency to the frame delivery. Adding this much latency may not be acceptable because it could have an adverse effect on performance.

A switch port may begin forwarding a frame as soon as it has examined the destination address and gained access to the switching construct. This is why the destination address is placed in the first word of the frame header. When cut-through routing is used, the frame may begin emerging from the output port before it has been completely received by the input port.

11.1.5 Cut-Through Routing and Speed Matching

If the input port and output port are operating at the same speed, or the input port is operating at a faster link rate than the output port, frame transmission at the output port can begin as soon as the first word of the frame header has been received and the path established through the switching construct.

If the input port is operating at a slower rate than the output port then frame forwarding can't begin until a sufficient number of words have been accumulated by the input port. Otherwise, the output port could run out of words before the frame is complete and be forced to insert one or more Idles into the data stream corrupting the frame.

For example, if the input port is operating at 1 gigabit per second and the output port at 2 gigabits per second, frame delivery can't begin at the output port until at least half of the frame has been received by the input port. In the case of an output port operating at 4 gigabits per second, frame delivery can't begin at the output port until 75% of the frame has been received at the input port.

Store and forward schemes avoid these considerations because frame delivery by the output port does not begin until the entire frame has been received.

11.1.6 Space-Division Switching

Space division switching provides a separate switching resource to connect every pair of ports. Examples of space-division switching constructs are crossbar switches and, perhaps, shared memory schemes. Since there is a separate switching resource for each port pair, multiple ports may communicate simultaneously and frames can be forwarded as an integral unit.

11.1.7 Time-Division Switching

Time-division switching uses a common switching resource shared by the ports on a time basis. Examples of time-division switching constructs are shared-bus and ring-based designs. Since the switching resource is shared by multiple switch ports, information is generally not forwarded as complete frames, but in smaller units.

While a switch port has access to the common switching resource, other switch ports are excluded. To ensure that all the switch ports have an opportunity to access the common switching resource in a timely manner, the duration of any access is limited. By limiting the amount of time a given switch port can use the switching resource, all ports can multiplex their traffic across the switching resource.

To minimize the latency, the access interval should be as short as practical. If the access interval is the duration of an entire frame, a switch port might have to wait for every port in the switch to deliver an entire frame before it could access the switching resource. In a 16-port switch, a port might have to wait for 15 other ports to each send a complete frame before it could send its frame (on a 100 MB/sec link a 2k-byte frame takes about 20 microseconds).

One way to minimize latency, is to give each port access to the switching construct frequently enough to keep delays to an acceptable value. If each port has access to the switching resource once per word rather than once per frame, the latency is more acceptable (15 words instead of 15 frames worth of delay). If frames are multiplexed at word boundaries, a frame is delivered to the destination port one word at a time, not all at once. The output port receives words rapidly enough so it can begin frame transmission once the first word has been received (it is not necessary to wait for the entire frame).

11.2 Crossbar Switching

The crossbar switch is an efficient switching mechanism with a long history reaching back to the early days of the telephone system. In a crossbar switch, an xy switching matrix is used to route traffic from an input port to one or more output ports. The switching matrix is non-blocking and provides the ability to handle multiple concurrent transfers between different port pairs without interference. An example of an eight-ported crossbar switch is shown in Figure 66.

In a crossbar switch, any input can be connected to any output by activating the appropriate switch point. For example to route traffic from input A to output D, the switch point at [A,D] is activated. If a reverse path is desired, then the switch point at [D,A] can also be activated. Because multiple switch points can be activated concurrently, the total throughput of the crossbar switch is equal to the number of port pairs. The eight-port switch shown in Figure 66 is capable of eight concurrent transfers, or a total of 800 megabytes per second (in a 100 MB/sec. Fibre Channel environment).

11.2.1 Crossbar Input Buffering

Each input port requires at least one frame buffer to hold a received frame while the crossbar switch is set up to deliver that frame. As soon as the switch point is activated, the frame can be forwarded to the selected output port. It is not necessary to wait for the entire frame to be received before forwarding begins. It is simply necessary that the switch point be activated before forwarding occurs. To expedite the switching process, the destination address is contained in the first word of the frame header.

To improve the performance on the link between the N_Port and the switch input port, it is desirable to have more than one input buffer available. This allows multiple frames to be in transit between the N_Port and the switch input port improving link utilization.

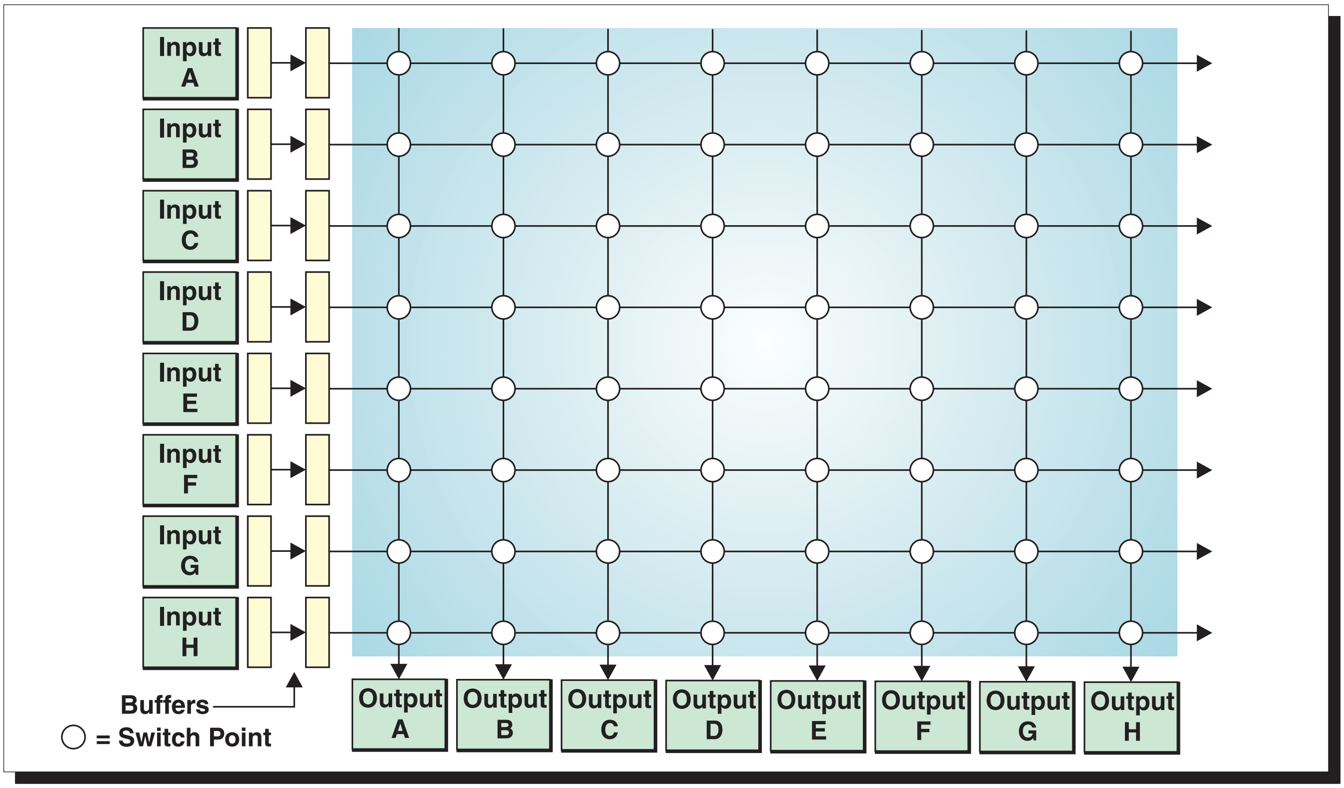

Figure 66. Eight-Ported Crossbar Switch

11.2.2 Crossbar Connection Modes

The crossbar switch can support both connection-oriented and connectionless modes of operation. An illustration of the crossbar switch with a dedicated connection between port A and port B and concurrent simultaneous connectionless traffic (indicated by dashed lines) from port D to port G, port F to port E and port G to port F is shown in Figure 67.

When a connection-oriented class of service is used, the appropriate switch points are activated when the connection is established and they remain activated until the connection is removed. In normal Class-1 operation, two switch points are activated to provide a bidirectional circuit between the two ports. Once the connection is established and the appropriate switch points activated, frames can be routed directly from the input port to the output port, bypassing the input buffers, if desired. For this reason, buffer-to-buffer flow control is not used after the connection is established.

When a connectionless class of service is used, the appropriate switch point is activated only for the duration of the frame or frames being delivered from one port to another. If a frame is being sent from input port F to output port B, the switch point [F,B] is activated for the duration of that frame. There is no need to concurrently activate the reverse path because that path will be activated when frames are received in the reverse direction.

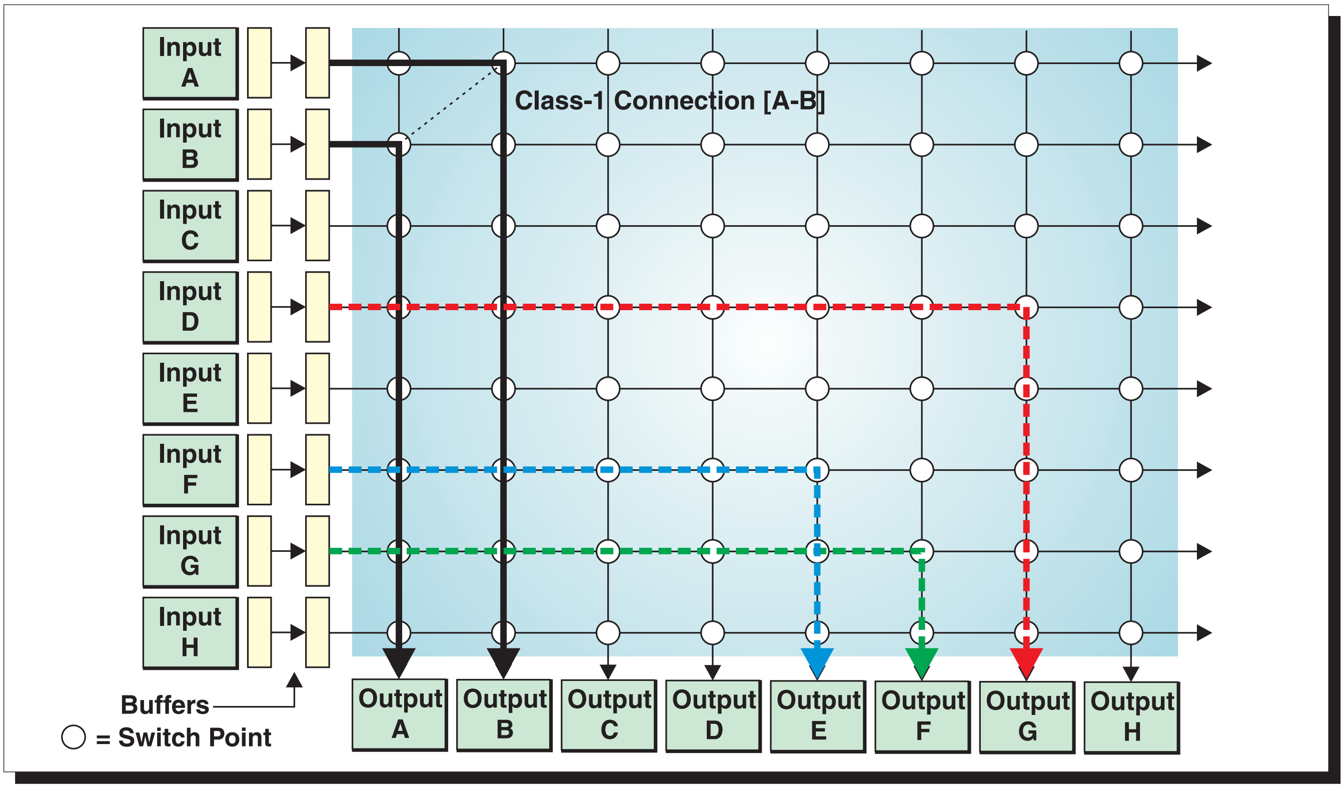

Figure 67. Routing Through a Crossbar Switch

11.2.3 Crossbar Broadcast and Multicast Support

The crossbar switch lends itself to performing broadcast and multicast operations. In this mode multiple switch points on the crossbar are activated simultaneously, allowing the frame or frames to be delivered to multiple output ports in a single operation.

An illustration of a multicast operation in a crossbar switch is shown in Figure 68 on page 150. In this example, port B is multicasting traffic to port A, port D, port E and port F in a single operation by activating the appropriate switch points. The switch points could be controlled by a bit mask with each bit corresponding to one of the switch points associated with a given input port. In this example, the multicast group could be represented as ‘10011100’b where a ‘1’ bit represents a member of the group and a ‘0’ bit denotes a non-member.

Broadcast is simply a special case of multicast where all of the switch points associated with a given input port are activated simultaneously, causing all frames placed on the bus associated with the input port to be sent to all output ports.

11.2.4 Crossbar Access Control (Zoning)

By disabling selected switch points, access from an input port to one or more output ports can be disabled. This would allow the switch to be configured so that a particular input port could only have access to a selected subset of the output ports. An example of this technique is shown in Figure 69 on page 151.

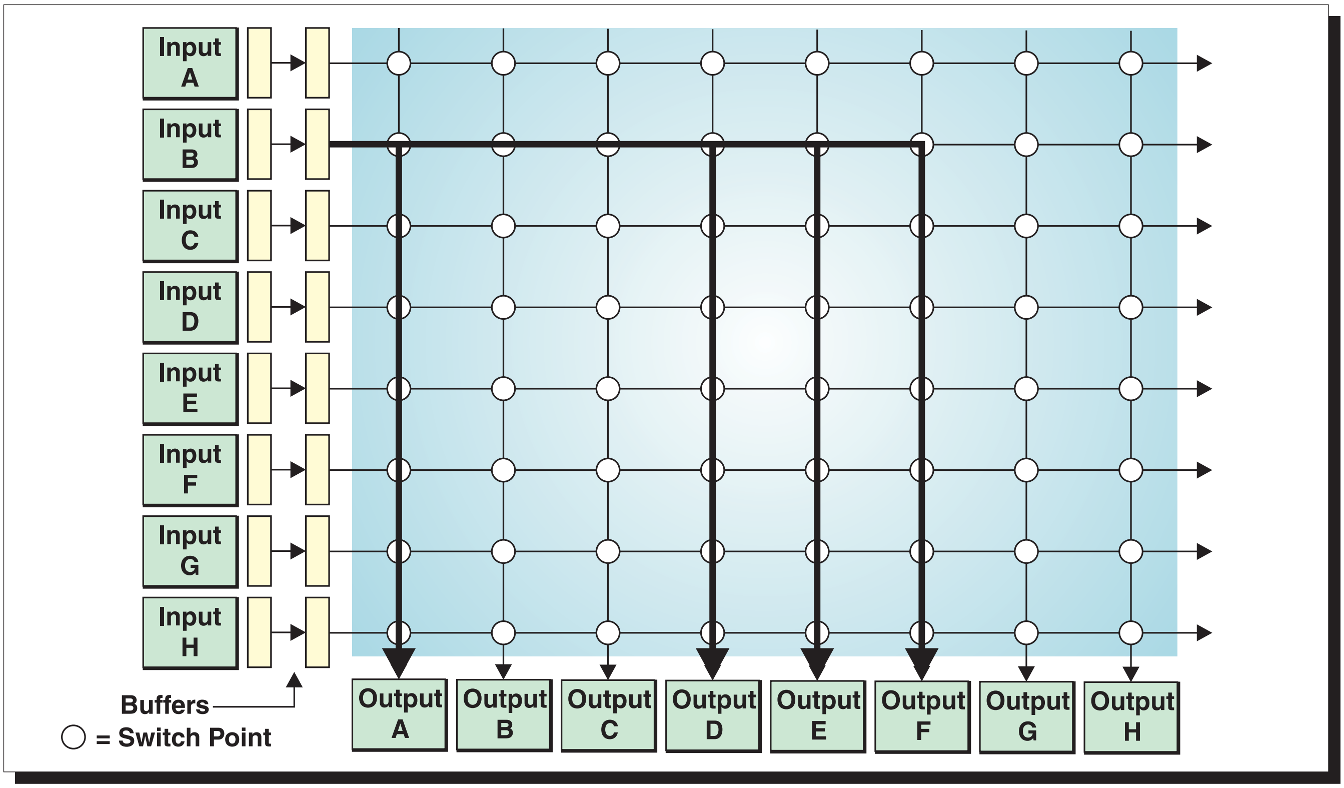

Figure 68. Multicast in a Crossbar Switch

In this example, input port A is enabled to send frames to output ports A, D, E, F and H. Access to output ports B, C and G is inhibited because the switch points to those destinations are disabled. Likewise, input port B is enabled to send frames to output ports B, C, G and H with access to output ports A, D, E and F disabled.

Limiting access in this manner does not imply that the disabled switch points are not present, simply that they cannot be activated. Inhibiting or allowing a set of output switch points associated with a particular input port could be controlled by a bit map that is set up by a system administrator or through other means. For example, the map for input ports A, D, E and F in this example would be ‘10011101’b, the map for ports B, C and G is ‘01100011’b and the map for port H is ‘11111111’b (where a ‘1’ bit represents enabled and a ‘0’ bit indicates disabled).

11.2.5 Crossbar Switch Summary

While the crossbar switch offers several attractive characteristics, it may not scale well to larger configurations. An the eight-ported switch requires 64 switch points, a 16-port switch requires 256 switch points and a 256-port switch requires 65,536 switch points. At some point, the number of switch points may become unwieldy and a different approach is needed.

11.3 Multi-Ported Memory

Another technique for routing frames is the multi-ported memory switch shown in Figure 70 on page 152. It shows a significantly different approach from the crossbar switch shown earlier.

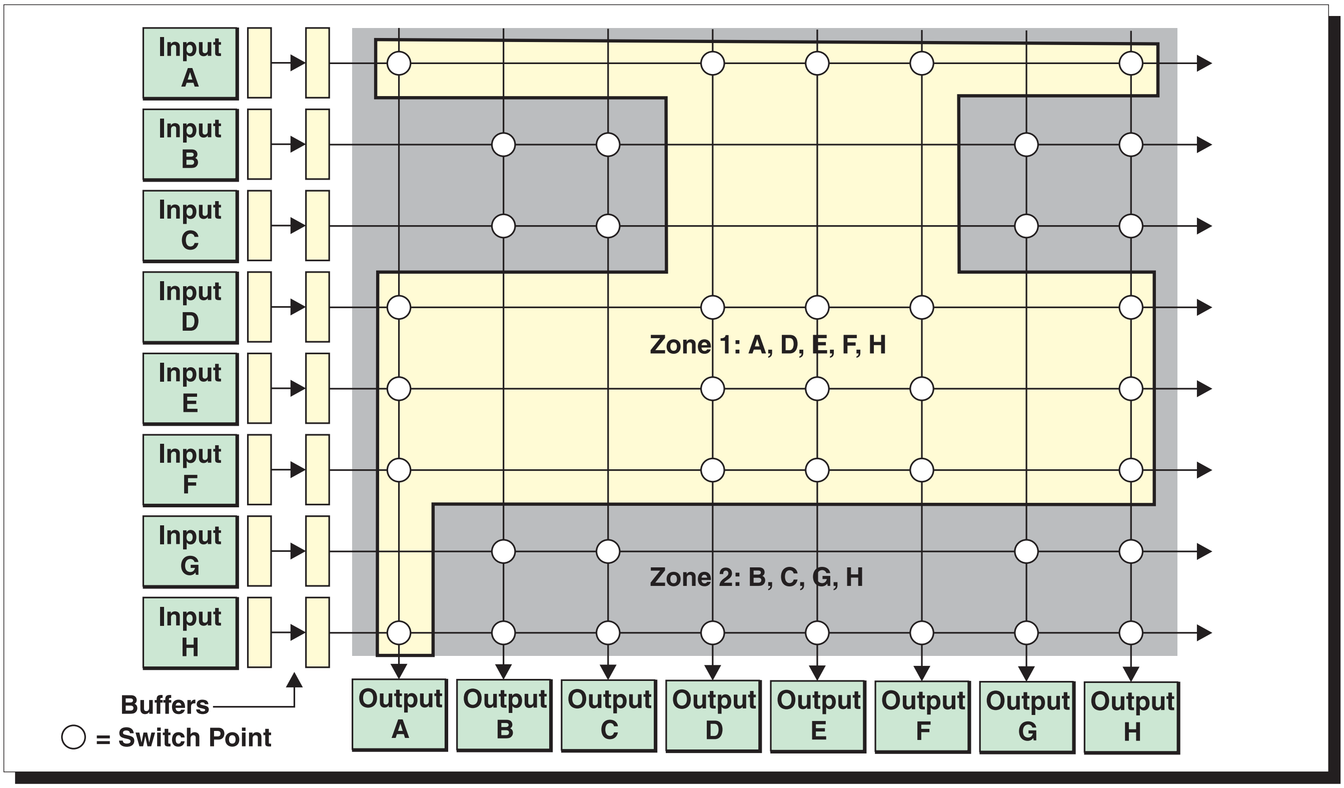

Figure 69. Access Control via Disabled Switch Points

In this routing mechanism, each input port has a queue associated with each output port. When a frame is received by the input port, the frame is stored in the appropriate queue based on the destination address in the frame header. The queues provide the buffering necessary to provide flow control for the associated input ports.

Each output port removes frames from one of the input queues assigned to that output port. When frames exist in multiple input queues, the output port can implement a load-balancing algorithm to ensure that each input queue receives an equal amount of service. This ensures that no input port monopolizes an output port.

Input queues can be implemented using a cut-through technique that allows the output port to begin processing a frame from the queue without waiting for the entire frame to be received. It is only necessary that enough of the frame be present to ensure that the output port does not empty the queue before the end of frame is sent.

The memory used for the queues could be implemented with individual memories, one per queue. Other packaging options could be used as well, such as one memory for all of the queues associated with a specific output port, one memory containing all the queues for a block of output ports or even one memory for the entire switch.

The maximum throughput of the switch is determined by the maximum throughput of the memories used to implement the queues. In a 16-port switch with full-speed Fibre Channel links, the bandwidth required to support full throughput on all links is in excess of 1.6 GB per sec-

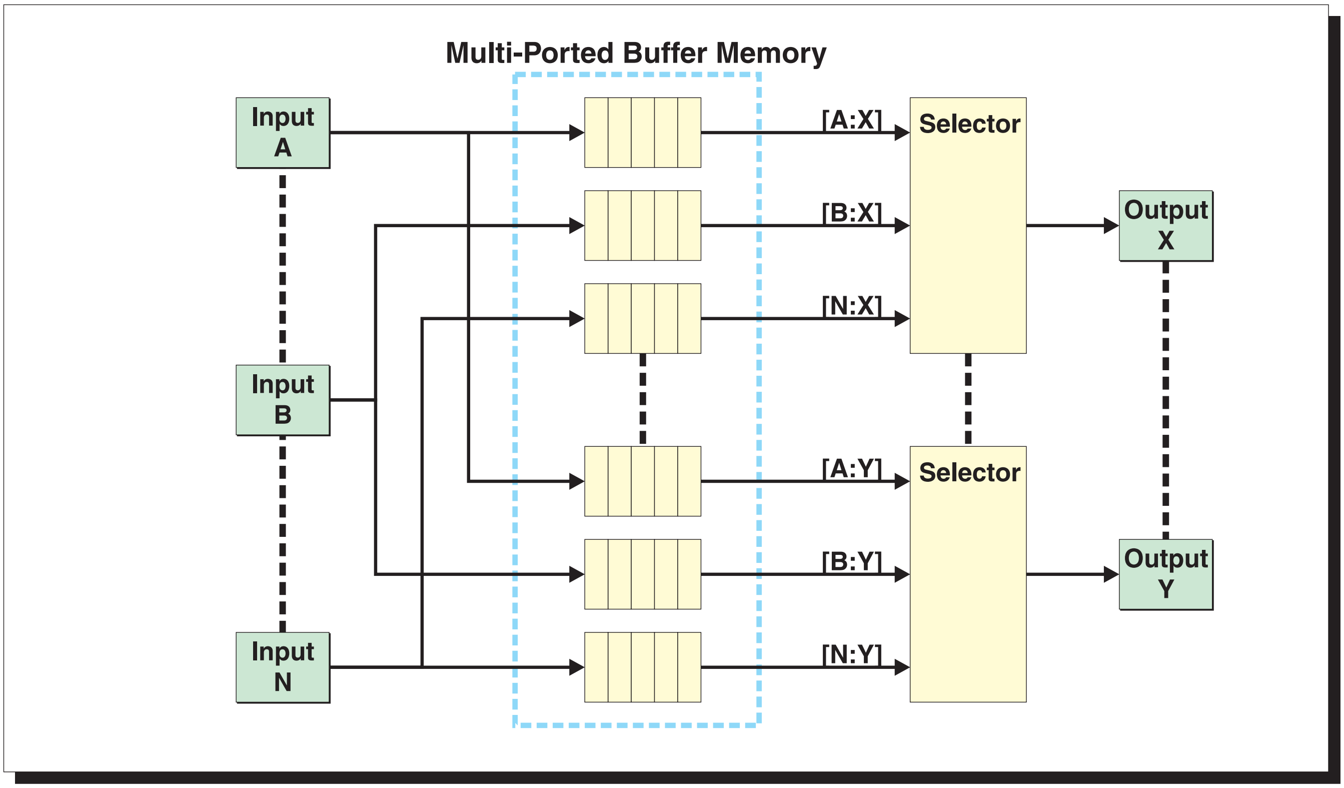

Figure 70. Multi-Port Memory Switching Construct

ond. Due to the difficulty of achieving this throughput with a single memory, some type of a multiple memory approach would probably be used.

11.3.1 Connection Modes

At first glance, it appears that this type of routing mechanism would not support Class-1 operation because the path from the input port to the output port cannot be guaranteed. In fact, this may be true if multiple output ports share the same memory and the memory does not have sufficient bandwidth to sustain the full link rate of all associated ports. However, if the memory bandwidth is sufficient and the output selector is locked to one of the input ports, then Class-1 can potentially be supported.

11.3.2 Multicast and Broadcast

Multicast and broadcast can be supported by storing the frame into multiple queues at the same time. To perform a multicast, the frame is stored in all the queues corresponding to output ports in the multicast group. As was discussed earlier for the crossbar switch, the ports contained in the multicast group could be represented by a bit map that is subsequently used to control the gating of frames into the appropriate queues.

For broadcast operation, the frame is simply stored into all the queues and will be subsequently processed by all the output ports.

Unlike the crossbar switch, the frames sent during multicast or broadcast may appear at the different output ports at different times depending upon their position in the respective queues. Normally, this does not represent a problem, but should be kept in mind if simultaneous delivery of frames is desired (simultaneous delivery is not required by the standard).

11.3.3 Access Control (Zoning)

An input port's ability to access the queue associated with an output port can be controlled in the switch to restricted access by a particular input port. Just as the crossbar example used a bit map to indicate the associated switch points, the multi-ported memory can use a bit map to indicate access to the queues associated with the set of output ports.

11.4 Bus-Based Switching

Another technique commonly used in switch designs is based on a high-speed shared data bus. An illustration of this approach is shown in Figure 71 on page 153.

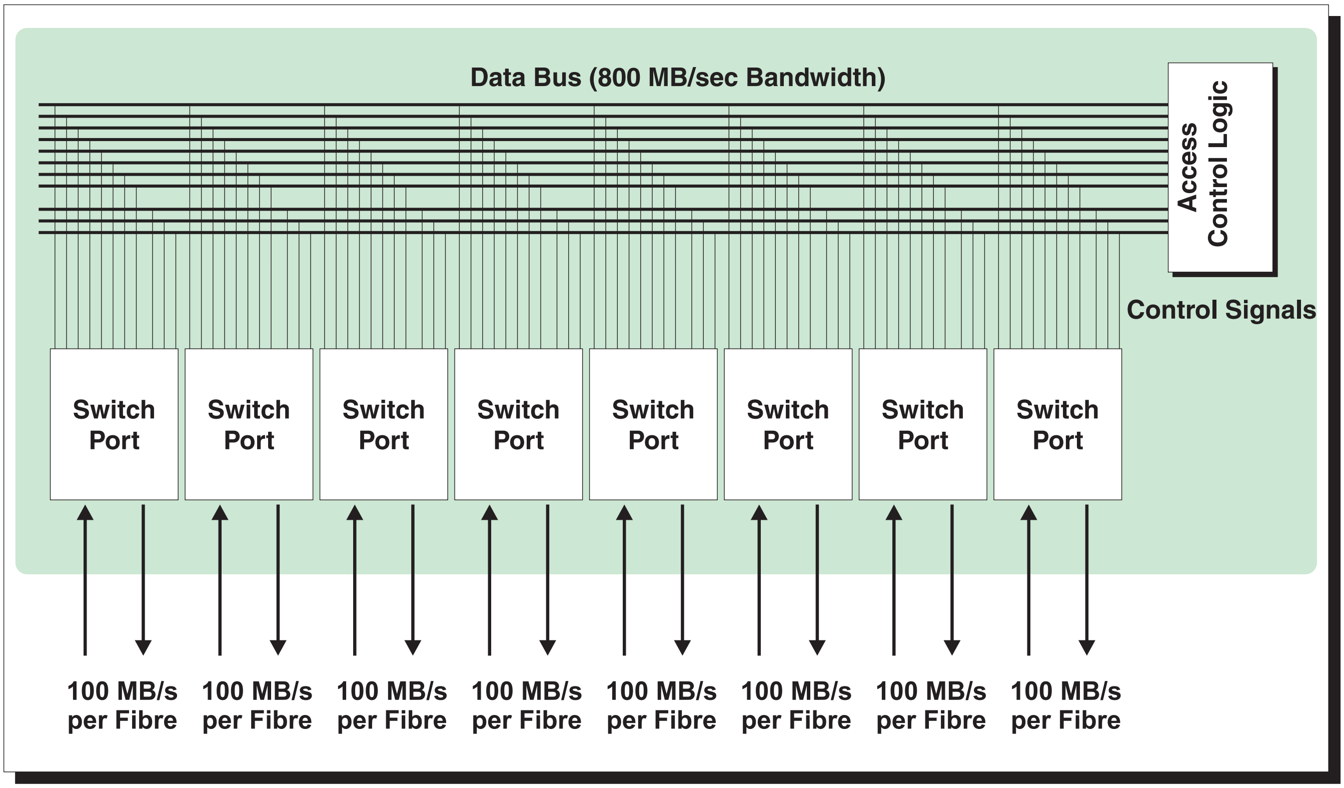

Figure 71. Bus Based Switching Construct

In a bus-based switching design, the switch ports connect to a common high-speed internal data bus. When a switch port has information for another switch port, it gains access to the data bus and sends that information to the other port.

11.4.1 Bus Management and Control

Because the bus is shared by all of the switch ports, bus bandwidth determines the overall throughput of the switching construct. If the bus bandwidth is sufficient to support the full requirements of the switch ports it does not limit the throughput of the switch.

If the bandwidth of the bus is less than the total bandwidth required by the ports, the bus may not be able to provide sufficient throughput to fully satisfy the switch ports. This may result in congestion during periods of heavy traffic and cause the switch ports to throttle the attached node ports using the buffer-to-buffer flow control mechanism. This may be considered an acceptable design tradeoff in return for using a lower-cost internal bus structure.

In a time-sliced implementation, the bus control logic gives each port an access slot on a periodic basis. This allows a switch's input port to forward information internally once per time slot. Timely access to the bus and bandwidth are guaranteed. If the number of access slots is not sufficient to allow the switch's input ports to keep up with the attached node ports, the switch's input ports must buffer frames from the node ports until they can be forwarded. If all of the switch's input ports buffers fill, the switch port throttles frame transmission by the attached node port using buffer-to-buffer flow control (i.e., withholds R_RDYs).

If an input port does not have anything to transfer during its time slot, the bandwidth provided by the slot is lost as it cannot be used by a different switch port. The larger the time slot, the more bandwidth is lost if the slot is not used. For this reason, it may be desirable to keep the time slots small in order to minimize lost bandwidth when an access slot is not used.

Assume the bus design shown in Figure 71 on page 153 transfers eight bytes of data at a 100 MHz. clock rate. The internal bus bandwidth is 800 megabytes per second (each bus cycle is approximately 10 nanoseconds). If a switch has eight ports and each port is given one bus cycle in a round-robin fashion, the longest a switch port waits for access to the bus is 70 nanoseconds (seven other ports' access slots).

Each port has access to the bus 12.5 million times per second (once every 80 nanoseconds) and can transfer eight bytes per access, resulting in a guaranteed bandwidth of 100 megabytes per second per port. As information arrives at the output port, the output port begins reassembling the frame and transmitting it to the destination; it isn't necessary to wait until a complete frame is available.

If multiple switch ports wish to transfer a frame to the same switch output port, it is necessary to ensure the integrity of the frames. The design may inhibit transfers from other switch input ports when a frame is being sent to a specific switch output port. While simplistic, this approach may waste bus bandwidth and create congestion at the input port.

Instead, the switch's output port may provide transmit buffers for the reassembly of frames from the switch's input ports. As transfers from an input port are received, they are reassembled in the correct transmit buffer. When a transmit buffer holds a complete frame, the frame can be scheduled for transmission on the output fibre.

Bus-based switching designs have been used for many different applications and the design requirements and trade-offs are well understood.

11.5 Chapter Summary

Switching Construct

- Each Fibre Channel switch contains a switching construct
  - The switching construct transports frames between the switch ports
  - The standards do not specify the design of the switching construct
- The classes of service supported impose requirements on the switching construct
  - Guaranteed bandwidth
  - Guaranteed latency
  - Frame delivery order

Switching Characteristics

- The switching construct may support different switching modes
  - Circuit switching (Class-1)
  - Frame switching (Class-2, Class-3 and Class-F)
- Frames for different ports may use the same internal resources (time-division multiplexing)
- Frames for different ports may use different internal resources (space-division multiplexing)

Zoning Enforcement

- The switching construct may provide zoning enforcement
- Limits access to only those ports in the same zone
  - Access to ports outside the zone is prevented
  - Provides ‘hard zoning’
- This may be used in addition to other types of zoning enforcement
  - Such as Name Server zoning which only filters the information made available to the port

Crossbar Switch

- One switching construct is the crossbar switch
- Uses an xy switching matrix allowing any input to be connected to any output
- A connection between one pair of ports does not block communication between other ports
  - Referred to as non-blocking
- Allows as many concurrent transfers as pairs of ports
  - 8-port switch can provide 800 MB/sec bandwidth
- Input buffering is required to hold frames while the path is setup

Multi-Ported Memory Switching

- Routing can also be done using a multi-ported memory
- Each input port puts received frames in a queue associated with the output port
- Output port selects frames from the queue for delivery
  - Output port can load balance traffic by selecting appropriate queue
- Routing begins as soon as possible
  - Cut-through routing, not store and forward

Bus-Based Switching

- A high-speed bus can be used to connect switch ports
- Each port has access to the bus to send frames to an output port
  - Access may be via arbitration or time slots
- Frames may be fragmented for delivery across the bus
  - 32-bit, 64-bit or even larger bus transfers
- Frame delivery at the output port begins as soon as possible
- Want to avoid a ‘store-and-forward’ operation
  - Adds too much latency to frame delivery

Head-of-Line Blocking

- ■ Frames may be blocked in switch buffers
  - A previous frame is occupying a buffer
  - That frame can't be delivered at this time (e.g., the path to that destination is blocked)
  - Subsequent frames to other ports are blocked by the previous frame
- ■ Look-ahead can alleviate head-of-line blocking
  - Switch port looks behind the blocked frame to see if other frames can be delivered
  - The deeper the look-ahead, the better the performance
  - Look-ahead can improve performance but adds complexity to the switch port

Congestion Control

- ■ Congestion on ISLs can cause performance problems
  - All of the receive buffers at a destination node port fill
  - The node port throttles the egress fabric port using flow control
  - This causes the egress port's buffers to fill
  - Frames back up causing the E_Port's receive buffers to fill
  - The receiving E_Port throttles the sending E_Port causing its transmit buffers to fill
- ■ The ISL is now blocked and no frames for any destination can be sent

End-to-End Flow Control

- ■ End-to-end flow control can alleviate congestion problems
  - If a destination node port is unable to accept and process frames it withholds EE_Credit
  - This throttles the source port and prevents congestion on the ISL (the source can still send frames to other destinations)
  - As the destination is able to accept additional frames it replenishes EE_Credit
- ■ This may not solve the problem if multiple sources send frames to the same destination
- ■ Most implementations are using Class-3 which doesn't support end-to-end flow control

Surrogate End-to-End Flow Control

- ■ Switch ports may mimic end-to-end flow control to alleviate congestion problems
  - If an egress switch port becomes congested, it signals the ingress switch port
  - The ingress switch port throttles the source node port using buffer-to-buffer flow control
  - As the egress port is able to accept additional frames it signals the ingress switch port
- ■ This may not solve the problem if multiple sources send frames to the same destination
- ■ Allows existing implementations to continue to use Class-3
- ■ This is a proprietary solution, not standardized

Congestion and Virtual Channels

- ■ Virtual channels provide another (proprietary) approach to congestion control
- ■ E_Port allocates its buffers among some number of virtual channels
  - Traffic flows are assigned to virtual channels
  - VC_RDY is used to manage the flow on each virtual channel
- ■ If one virtual channel becomes congested, frames for other virtual channels can still be delivered
- ■ Virtual channels do not necessarily guarantee a specific amount of bandwidth
  - Simply alleviates blocking due to congestion