# Simulation Study of Fibre Channel Fabrics with Particular Emphasis on 64-Node Clusters

Ludmila Cherkasova, Vadim Kotov, Tomas Rokicki
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94303

**Abstract.** The Fibre Channel (FC) standard was developed by ANSI X3T9.3 task group to define a serial I/O channel for connecting a number of peripheral devices to computer systems as well as interconnecting the computer systems themselves.

In this report, we investigate the performance of a FC switch having 16 ports and a specific high-level architecture. First of all, the basic performance of a single Fibre Channel switch is analyzed. The evaluation of the switch is done for different types of workload: random and bursty traffic. We investigate how the FC switch performance depends on the class of service used.

We also investigate the performance of a few basic topologies for cascading FC switches in order to connect approximately 64 nodes. Among these topologies are the star, house, envelope, and complete configurations.

When connecting Fibre Channel switches with multiple trunk links, randomization might be thought helpful in balancing the traffic. The simulation results show that the deterministic routing strategy is slightly more efficient than the random one when FIFO scheduling is used inside the Fibre Channel switch, but the difference is slight.

We discuss some typical problems that occur when FC switches are cascaded such as deadlocks and unfairness, and we present some potential solutions.

**Key Words:** Fibre Channel switch, cascading, performance analysis, routing strategies, random and bursty traffic, end-to-end credits, simulation model, C++ Sim.

# Contents

# 1 Introduction

The Fibre Channel standard [AC92, GR92] provides a mechanism to interconnect heterogenous systems containing peripheral devices and computer systems through optical fiber media. In this report, we consider a basic Fibre Channel switch with 16 ports at speeds of 265.625 MBaud. First, the basic performance of a specific Fibre Channel switch architecture is analyzed through simulation, identifying the primary bottlenecks of system performance. We consider both a random and a bursty traffic workload, and we illustrate how the delivery class affects the performance of the switch.

In this report, we only consider delivery classes 2 and 3; the primary difference is whether or not acknowledgement frames are generated for each received data frame. The Fibre Channel switch we consider has a limit on the frame size that corresponds to a payload of approximately 2048 bytes. Longer messages than this are split into multiple frames; the full set of these frames is called a sequence. Class 2 service is based on connectionless data transfer with the acknowledgement of delivery or failure. Under this service, each frame arriving at its destination generates an acknowledgement frame that is delivered to the source These acknowledgement frames are not acknowledged. The Class 3 service is also connectionless and is similar to Class 2, except no acknowledgement frames are generated.

We also consider cascaded topologies of Fibre Channel switches containing between five and eleven switches, and approximately 64 terminal nodes. Typically, the bottleneck in Fibre Channel fabrics is the throughput of the trunk links (links connecting neighbor switches). For a given number of switches, there is a trade-off between the number of terminal nodes (that generate traffic) and the amount of traffic that each terminal node can generate. For high traffic requirements, it is sometimes advantageous to have multiple trunk links between a given pair of switches.

We consider two routing strategies when using multiple trunk links: randomized vs. deterministic. Under the randomized strategy the routing choice of which trunk link to use is done randomly. The deterministic strategy assigns the different trunk links to particular paths. The intuitive idea behind randomized routing is that it allows implicit load balancing (assuming a uniform traffic demand matrix) between the trunk links, thereby increasing throughput. The simulation results counter this, and show that in some cases, the deterministic routing strategy is slightly more efficient than the randomized one. However, there exist situations in which randomized routing does win. And, for some topologies, these two routing strategies are practically indistinguishable. Thus, we cannot conclude that one strategy is significantly better than the other, and the choice should probably be made by considering implementation details and the necessity of avoiding deadlock.

We present some results that indicate that multiple trunk links should only be used once each switch has a trunk link to each other switch. We present and discuss some fairness problems that arise in a cascade of Fibre Channel switches, and present some solutions to this problem.

This paper is organized as follows. Section 2 describes the Fibre Channel switch design, the main parameters and the basic model assumptions. Section 3 presents the performance results of a single FC switch for different classes of services and workloads. Section 4 presents a performance study of special topologies for cascading FC switches in a fabric such as the star, house, envelope, and complete topologies. This section introduces some of the problems that occur when cascading FC switches and some potential solutions.

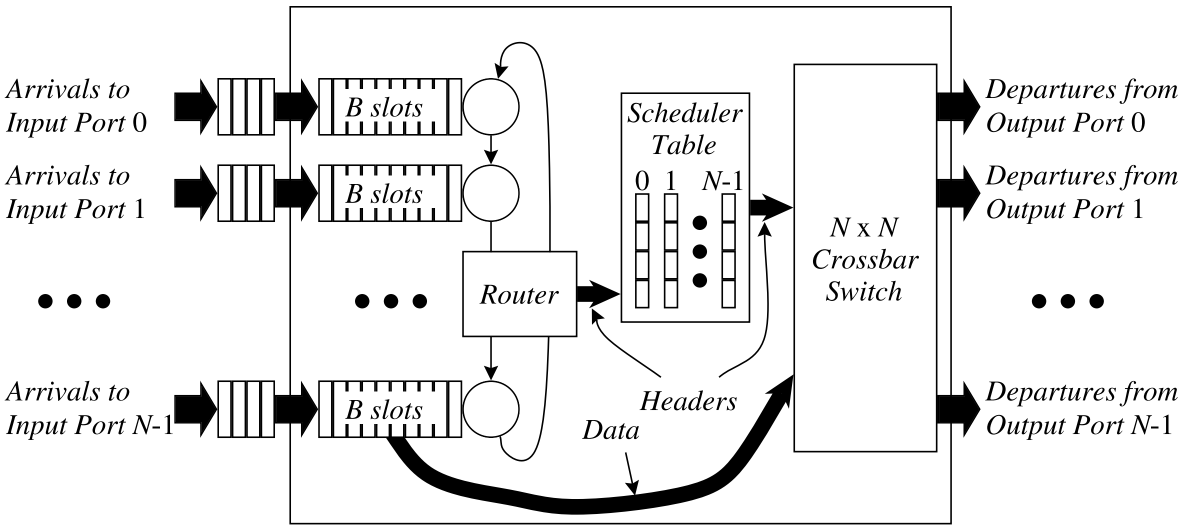


Figure 1: System structure

## 2 Fibre Channel: Model Specification and Assumptions

We consider a switch with 16 ports, numbered from 0 to 15 (see Figure 1). Each port has an associated pool of 15 buffers to store incoming frames. When a frame arrives at a port, it starts being read into a buffer as soon as one is available. A switch router iteratively polls each port for new frames in a cyclical fashion. The router polls port $i$ to check whether it contains frames to be routed. If so, the router reads the header information for the first of them and inserts it into a scheduler table, organized as 16 FIFO queues, one for each possible frame destination (we assume that a frame arriving from port $p$ chooses any of the other 15 ports with uniform probability). Then, it polls port $(i+1)$ mod 16, and so on. If there is no frame to be routed, the amount of time spent by the router at a port is negligible. Note that the router can start routing a frame even if the frame has not been completely read into the buffer. A switch scheduler uses the scheduler table to control the use of a $16 \times 16$ crossbar switch. The constraints that the scheduler must observe are:

- The crossbar switch can transfer at most one frame from each source port at any given time.
- The crossbar switch can transfer at most one frame to each destination port at any given time.
- A frame can be transferred to its destination port only when it is at the top of the queue for that destination.

For example, considering only four ports, if the content of the scheduler table is as in Table 1, the crossbar could be transferring the frames $\{1 \mapsto 0, 2 \mapsto 1, 0 \mapsto 3\}$ or the frames $\{2 \mapsto 1, 1 \mapsto 2, 0 \mapsto 3\}$. Note that, in the first case, the transfer $3 \mapsto 2$ is not possible, even if the source 3 and the destination 2 are idle, because the frame from source 3 is not at the top of the queue for destination 2. In other words, source 1 blocks source 3. Such a situation is called *a head of line blocking*.

| | Destination | | | |
|---|---|---|---|---|
| | **0** | **1** | **2** | **3** |
| **First** | 1 | 2 | 1 | 0 |
| **Second** | 2 | | 3 | 2 |
| **Third** | 1 | | | 1 |
| ... | | | | |

Table 1: A possible scheduler table configuration

The bandwidth of each port is 26.6 Mbytes/sec. The value of the bandwidth affects the rate at which data moves through each of the components of the switch. The maximum (and typical for our study) length of a frame is 2084 bytes which consists of 2048 bytes of payload and 36 bytes of header. Acknowledgement frames have a 0 bytes of payload and the same 36 bytes of header.

A frame can start being forwarded through the crossbar as soon as its entire header has arrived and has been checked for validity; this is known as the virtual cut through technique [Fujimoto83]).

We constructed a simulation model capturing the essential architectural features of the switch. This simulation model was built with the CSIM C++ class library and consisted of approximately 3,000 lines of code.

All simulation runs involved over 1,000,000 frames, and the 95% confidence intervals using the batch means approach were in all cases tighter than $\pm 1\%$. For low traffic, the confidence intervals were significantly better than this.

# 3 Fibre Channel Switch: Basic Performance

## 3.1 System Performance for Class 2 Frames

Class 2 service is based on a connectionless data transfer with the acknowledgement of delivery or failure. Under this service, a message sent from one node to another one is split into frames of fixed length (the maximum payload size of approximately 2048 bytes). By each frame's arrival to a destination node, an acknowlegement about its delivery is sent back to a source node. The Class 2 service has a flow control mechanism provided by a switch to make frame transfer more reliable and predictable. Each port (more exactly, terminal node) has a table of end-to-end credits (end-to-end credits) which is a vector with an entry for each destination terminal node (of the switch or configuration of the switches). Each such an entry stores a nonnegative number of end-to-end credits which defines exactly how many frames can be sequentially sent from this source node to that destination node without waiting for an acknowledgement. A limited number of end-to-end credits restricts the node from injecting frames if there is congestion on the route to the frame's destination.

A class 2 frame can be injected from a terminal (source) node $x$ to the switch if the following three conditions are satisfied:

- the local port is available;
- there is an available buffer in the buffer pool associated with the the local port to store this frame;
- there exists an end-to-end credit corresponding to the destination node $y$ of the sending frame.

Whenever these conditions are satisfied and the frame is injected to the switch through the local port, the number of available buffers in the switch and the number of end-to-end credits corresponding to a destination node $y$ are decreased. When an acknowledgement of that specific frame is received from the destination node, the end-to-end credit value is incremented.

For the rest of the paper the following terminology for performance metrics is used.

The *frame latency* is measured from the time the local port starts to inject the frame to the FC switch to the time the frame is completely received by its destination node.

The *acknowledgement frame latency* is measured from the time the acknowledgement frame generated at the destination node to a time it is received by the source node.

The *round-trip frame latency* consists of the total frame transfer latency plus the acknowledgement frame latency. It is measured from the time the local port starts a frame injection to the time the full corresponding acknowledgement frame arrives at the original source node.

The *total message latency* is measured from the time the message was generated (appeared at the source node site) to the moment when all the frames belonging to this message are received at the destination port and all the acknowledgement frames are received by the original source node.

Figure 2 shows the switch performance, in terms of message latency, with different initial number of end-to-end credits under uniform random traffic. This traffic consists solely of single-frame 2048 byte messages with sources and destinations chosen at random. The horizontal axis is the percentage of fibre utilization.

The maximum switch performance (throughput) varies from 47% to 53% depending on the end-to-end credit value. Each step in increasing the end-to-end credits from 1 to 2, from 2 to 3,and from 3 to 4 increases the maximum switch throughput by about 2% of the fiber capacity. However, further increase of end-to-end credits apparently does not increase the maximum throughput.

Figure 3 shows the set of basic switch performance metrics. From the bottom curve up, these are the acknowledgement frame latency, the frame latency, the round-trip frame latency and the total message latency for random traffic and an end-to-end credit value of two.

This picture illustrates that frame transfer latency is the major part of the overall message latency. Message waiting time in the queue is quite small. Acknowledgement frame latency takes a fair portion of overall message latency: between 10% for light traffic and 30%-40% for heavier traffic. This reflects the increasing amount of time that frames wait in the switch

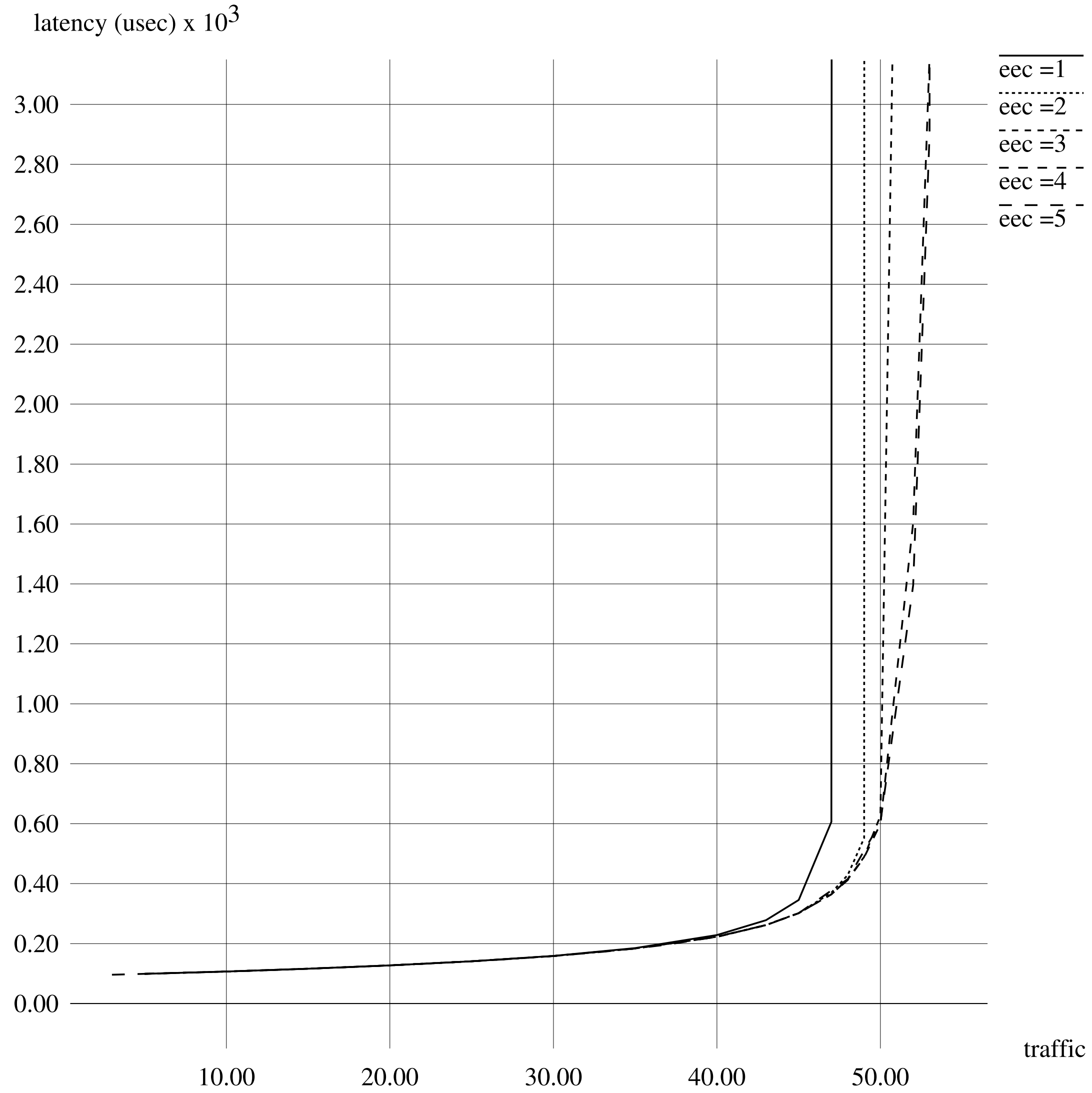


Figure 2: FC switch performance with varying end-to-end credit values under random traffic

as traffic rises. Under low traffic, the head of a data frame leaves shortly after it arrives at the switch, and most of the latency is due to the time to transfer the bytes through the port; under heavier traffic, the switch must wait in the internal queues and much of the latency comes from this waiting time.

We also performed a number of simulations using bursty workloads. These workloads are defined primarily by a message length distribution. Rather than considering many different message length distributions, we consider only bimodal distributions consisting of short messages and long messages. We define short messages to be from one to five frames in length (i.e. from 2,048 bytes to 10,240 bytes of payload), and we give each length equal probability. We choose a size of twenty-five frames for long messages (i.e. with 51,200 bytes of payload)

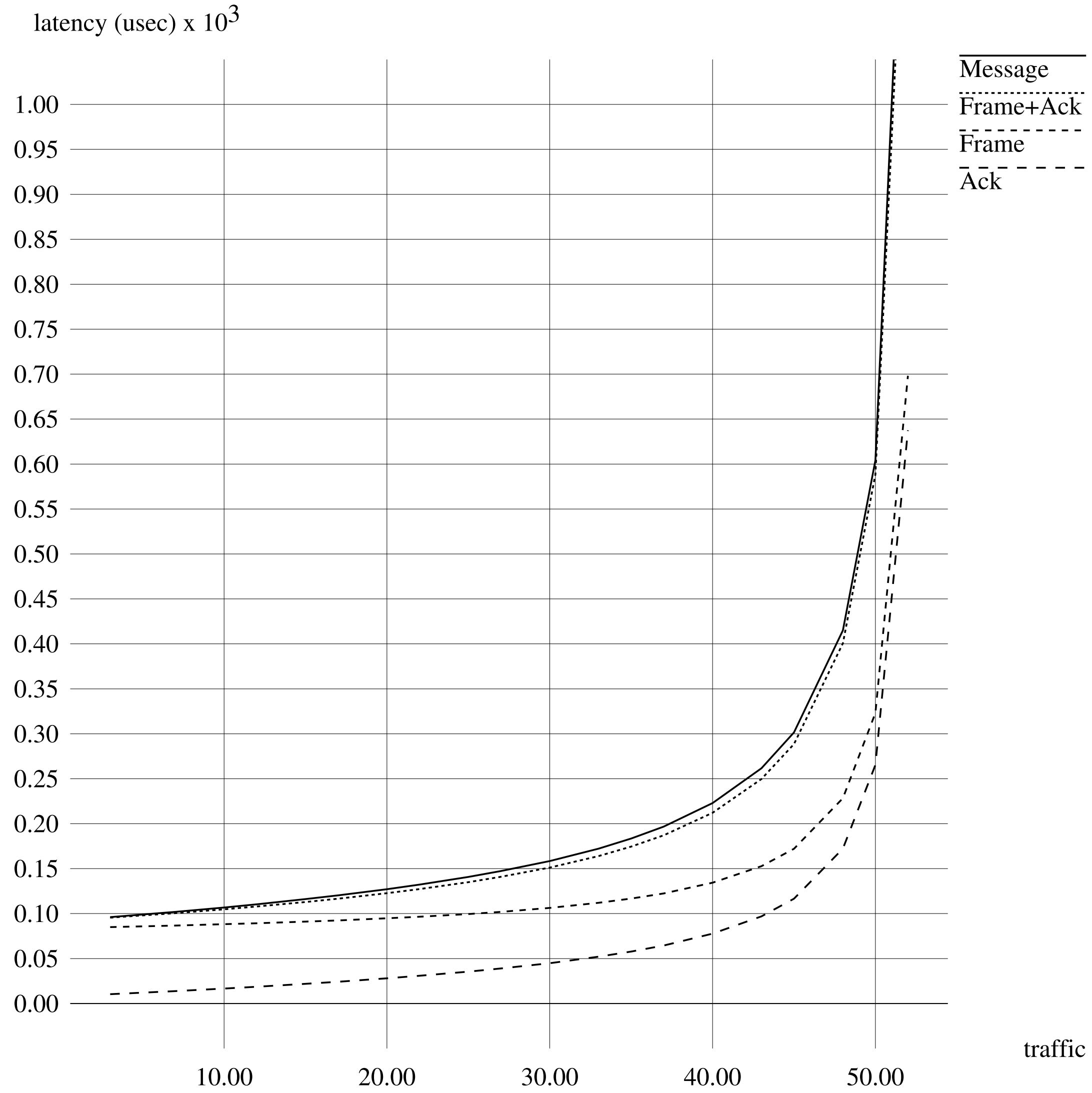


Figure 3: Basic set of latencies under uniform random traffic

We define our workloads by the percentage of long messages in the workload; this is the primary variable defining the workloads. For instance, a workload with 10% long messages has an average message length of 5.2 frames, and 48% of the frames were part of a long message.

Figure 3 shows the switch performance with different initial number of end-to-end credits under a workload with 10% long messages.

First of all, we can observe that switch throughput under the bursty workload is almost 10% of the fiber capacity less than under the uniform random traffic, and varies between 41% and 43% depending on the end-to-end credit value. Due to the higher variance in message lengths and the limited end-to-end credit values compared with the average frame length, the head

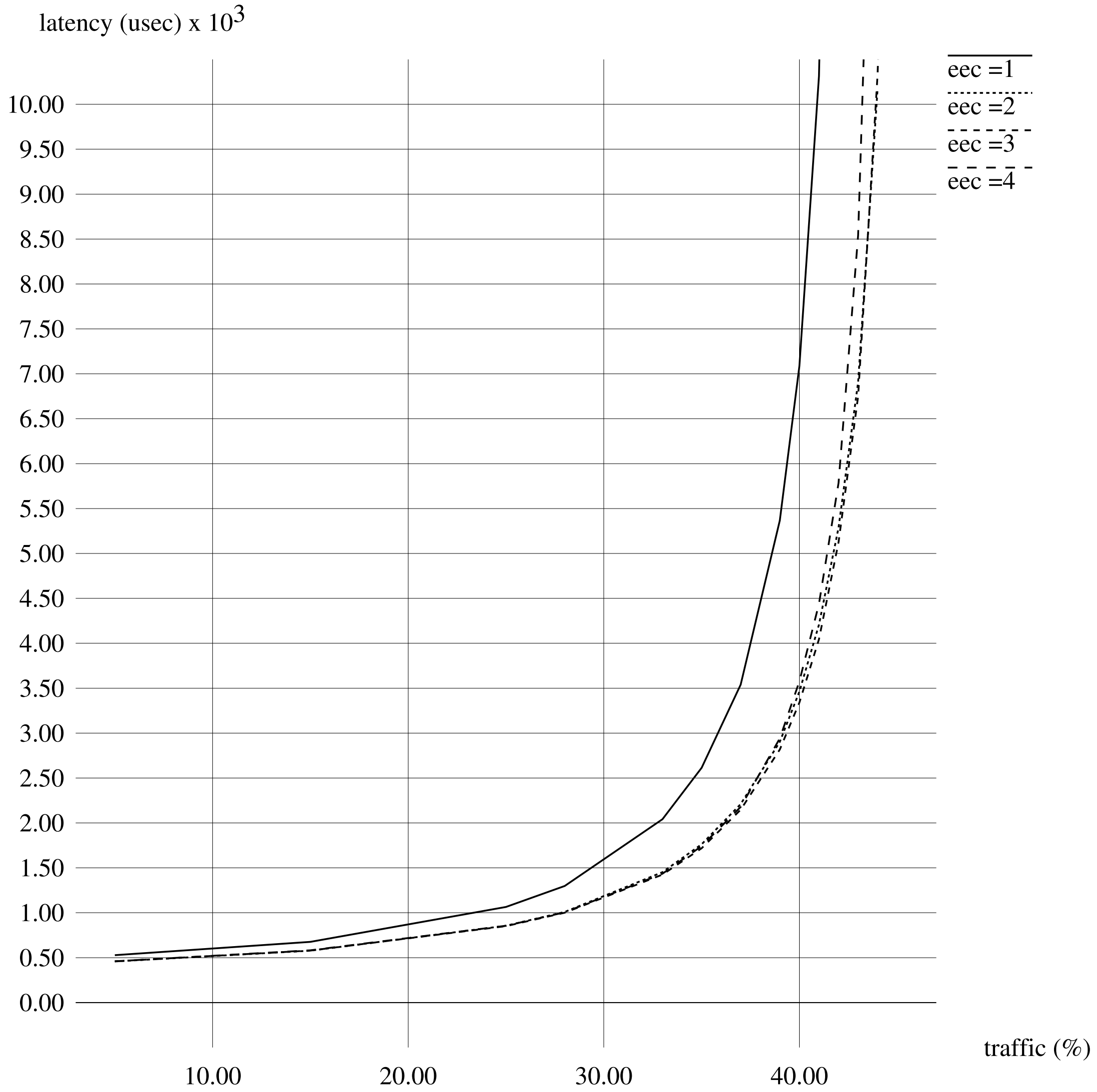


Figure 4: Switch performance with different values of end-to-end credits under 10% long messages workload

of line blocking probabilities are essentially higher than under uniform random traffic. Thus, while increasing end-to-end credits from 1 to 2 increases switch performance (by allowing the pipelining of data frames and acknowledgement frames), further small increases have little effect. Figure 4 shows that an end-to-end credit value of 2 is typically a good choice for an uncascaded FC switch.

The end-to-end credits are set equal to 2 for the rest of simulation experiments reported in this work if not otherwise specified.

### 3.2 System Performance for Class 3 Frames

Class 3 service is similar to class 2 service, except no acknowledgements are generated and the end-to-end credit value is ignored. Therefore, the switch throughput for the class 3 frames is almost 10% of the fiber capacity higher than for the class 2 frames. The primary reason for this is because the class 2 average frame length is half that of the class 3 average frame length (because the acknowledgement frames have a zero-byte payload), and thus the scheduler must schedule twice as many frames in the same period of time to attain the same fiber utilization. Because of its limited speed, the scheduler becomes more of a bottleneck for class 2.

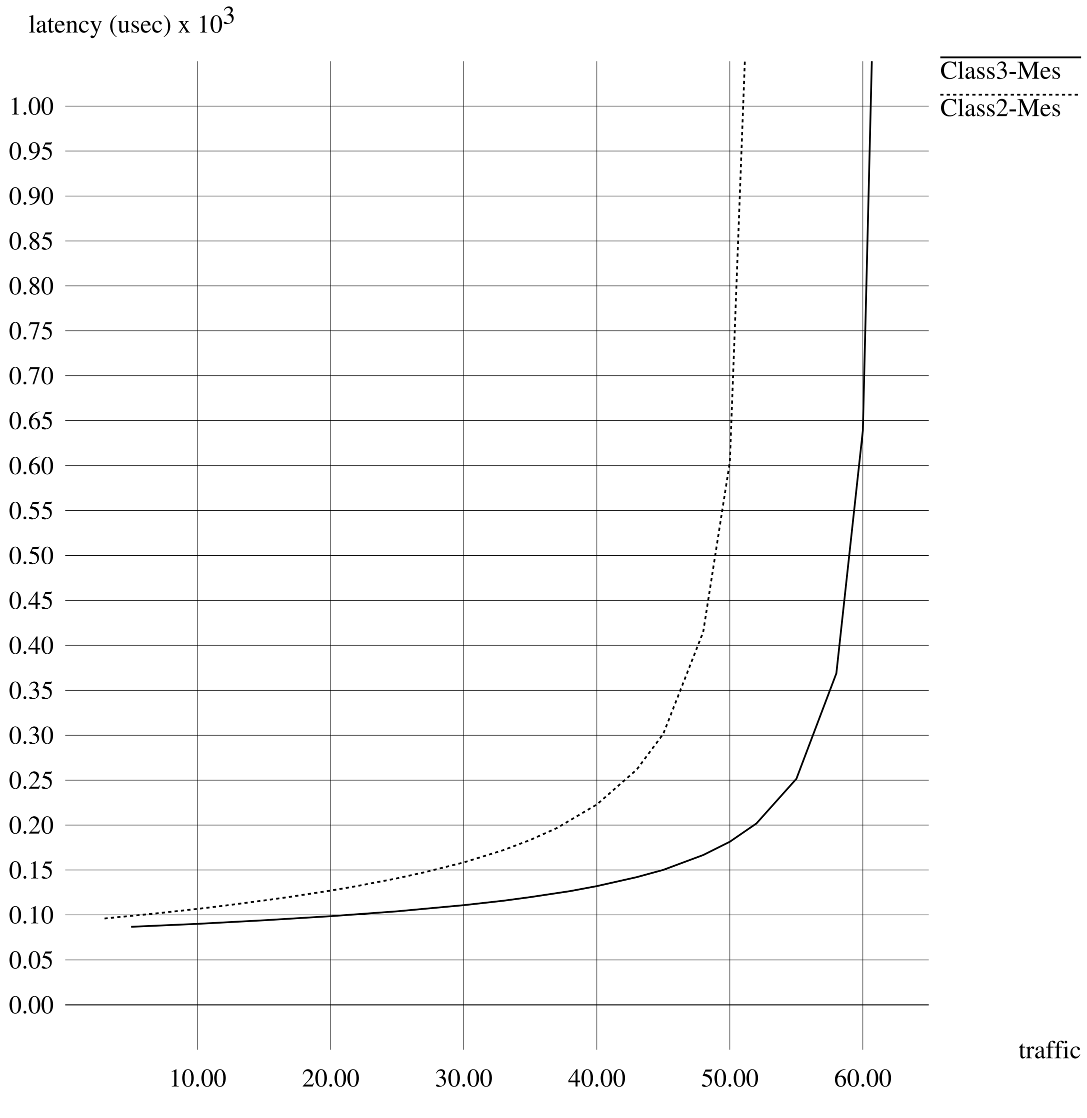


Figure 5: Switch performance for class 3 service vs. class 2 service under uniform random traffic

Figure 5 shows the switch performance for class 3 service vs class 2 service under the uniform random traffic.

Figure 6 shows the buffer usage per port for class 3 service vs class 2 service under the uniform random traffic. Each port has 15 buffers to store incoming frames. As Figure 7 shows the buffer usage per port typically is very low, and sharply increases only when the switch is functioning near its maximum capacity.

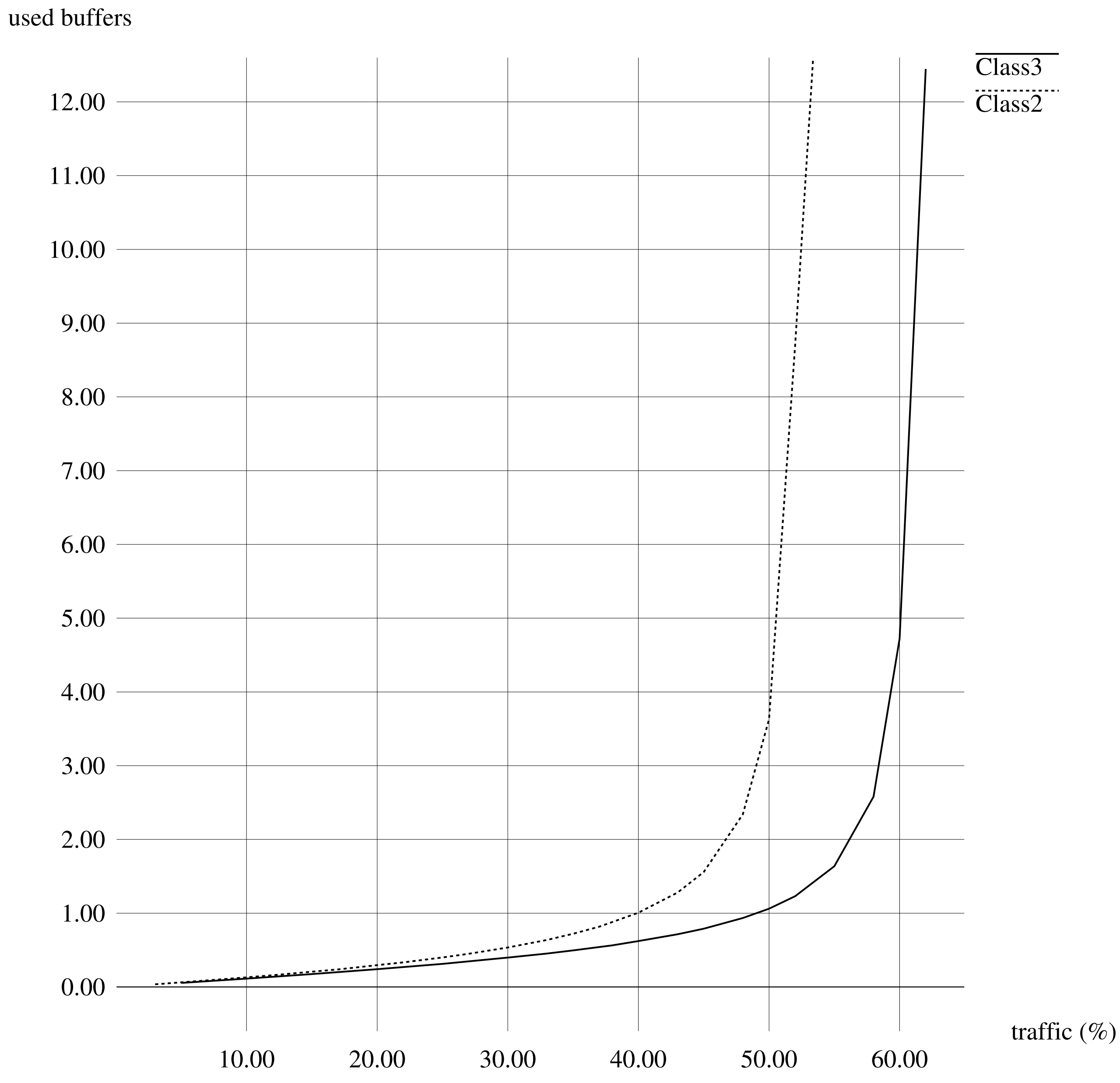


Figure 6: Switch buffer usage per port for class 3 service vs. class 2 service under uniform random traffic

# 4 Simple Case of Cascading Five Fibre Channel Switches: Performance Evaluation Study

In this Section four simple and popular configurations to connect five 16 port FC switches will be considered. These configurations have the names star, house, envelope, and complete.

We will analyse fabric performance for each topology as well as point out some particular performance problems that arise.

## 4.1 Fibre Channel Fabric with Multiple Trunk Links: Randomized vs. Deterministic Routing

Typically, when cascading Fibre Channel Switches in a fabric the trunk links (links connecting two neighbor switches) are the system bottlenecks. There is always a trade-off between the number of terminal ports and the sustainable aggregate throughput. By decreasing the number of terminal ports and using the freed switch ports for additional trunk links, performance may be improved.

When connecting the Fibre Channel switches with multiple trunk links, such that there is more than one trunk link between a pair of switches, two routing strategies are analyzed: *randomized* vs. *deterministic*.

Under the *randomized* strategy the routing choice of which trunk link between the switches to use is done randomly on a frame-by-frame basis. This may not be practical, but the results will yield some useful insight. Intuitively, it is felt that using randomized routing will help balance the traffic across the trunk links, leading to an overall improvement in throughput; by randomizing the routing on a frame-by-frame basis, we should maximize this advantage.

The *deterministic* routing strategy is based on partitioning the terminal nodes and prescribing which trunk link should be used for each destination, i.e. there is predefined path (table) to route each particular frame.

### 4.1.1 The Star Configuration

In this section, the star configuration with different numbers of trunk link multiplicity will be considered.

The first star configuration we will consider is shown in Figure 7. Trunk links connecting switches 1,2,3,4 to a hub switch 5 have a multiplicity equal to 2. This configuration allows 64 terminal nodes: 14 terminal nodes on each of switches 1,2,3,4 and 8 terminal nodes on the hub switch 5.

Simulations were performed to analyse the advantages of randomized vs. deterministic routing strategies for both previously described workloads.

Figure 8 shows the overall fabric performance using different routing strategies and workloads.

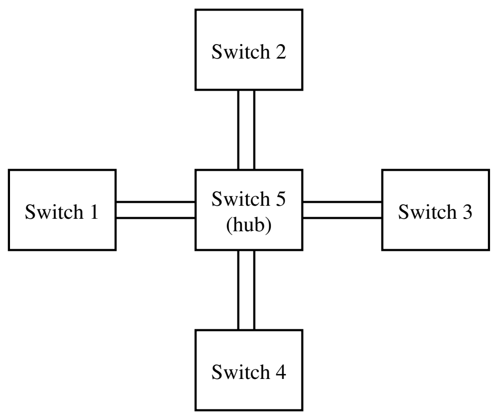


Figure 7: Star configuration with trunk link multiplicity = 2

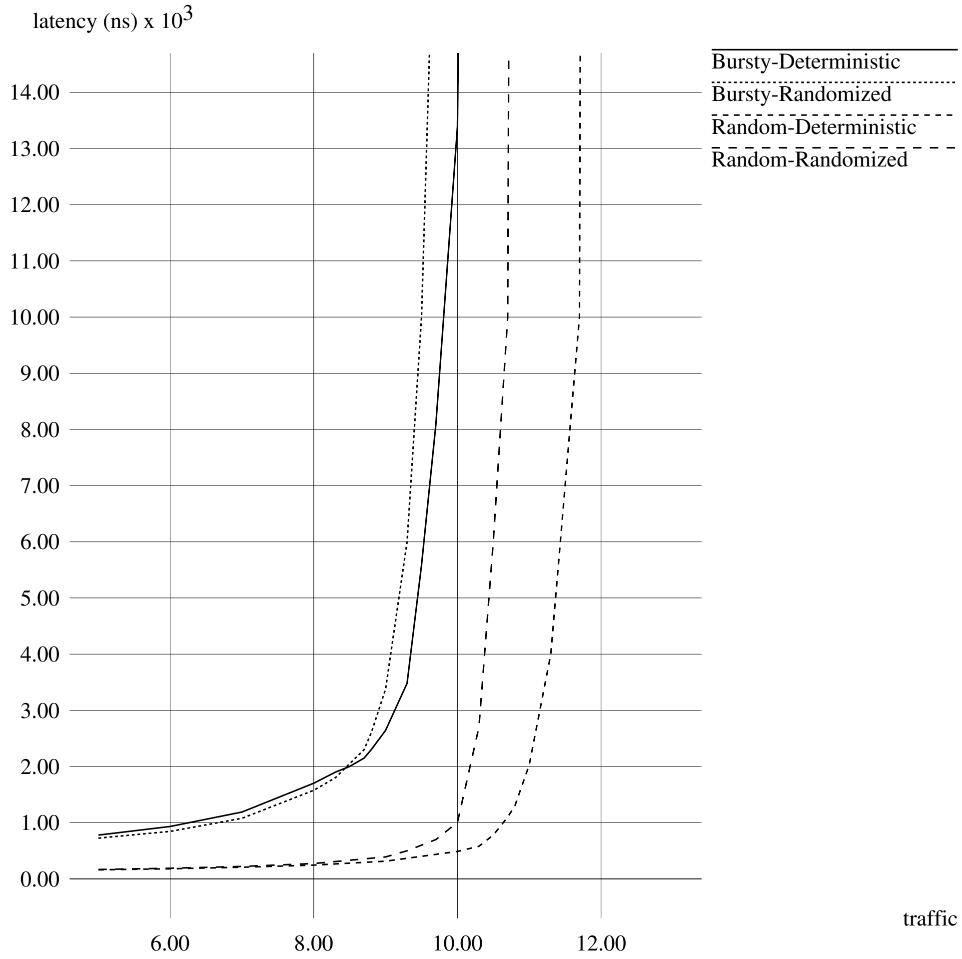


Figure 8: Star with trunk link multiplicity = 2: randomized vs. deterministic routing under bursty and uniform random traffic

The results show that the deterministic routing strategy slightly outperformes the randomized strategy for both workloads. This seems to be counterintuitive because accordingly to randomized routing strategy when a frame is routed through the trunk links (which are a system bottleneck) the frame is uniformly assigned to use either trunk link. This would seem to lead to a better dynamic distribution of trunk links usage than using a deterministic strategy.

However, because of the FIFO scheduling discipline and head of line blocking the results are different. To explain this, let us consider the following example. Consider a FC switch with two trunk links to a neighbor switch, as shown in Figure 9. If some node, say 7, is sending a burst of frames with the same destination node using the randomized routing strategy then the frames from the node 7 are added to the output queues for both trunk links 1 and 2. If at some moment, the frames from node 7 are happened to be in the head of both queues, then only one trunk link will be sending a frame since each needs the source port of the switch to send the frame across the crossbar. In such a situation the second trunk link capacity will be wasted. This situation is excluded in deterministic routing: all frames from the same burst use only the assigned trunk link. A more aggressive internal switch scheduling strategy might help recover some of the advantage of randomized routing.

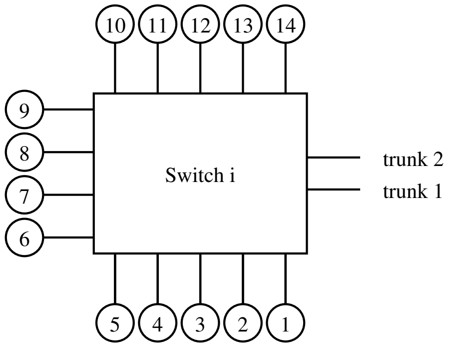


Figure 9: Switch with two trunk links to a neighbor switch

The star configuration with trunk link multiplicity equal to 3 is shown in Figure 10. Since there are more trunk links between the switches, there are fewer terminal nodes per switch; this configuration allows only 56 terminal nodes, but has more trunk link capacity.

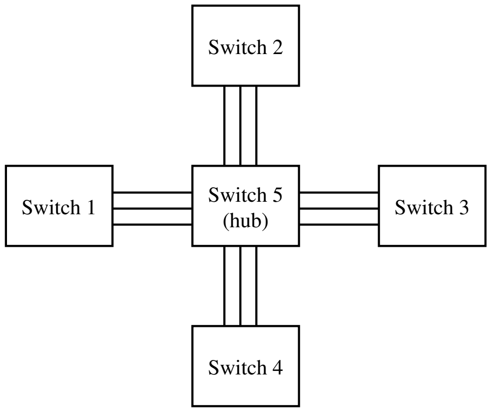


Figure 10: Star configuration with trunk link multiplicity = 3

Figure 11 shows the overall fabric performance using different routing strategies and workloads. The overall fabric performance much greater than the performance of star configuration with trunk link multiplicity equal to 2. However this time, the fabric performance with randomized routing strategy is slightly better when with the deterministic one.

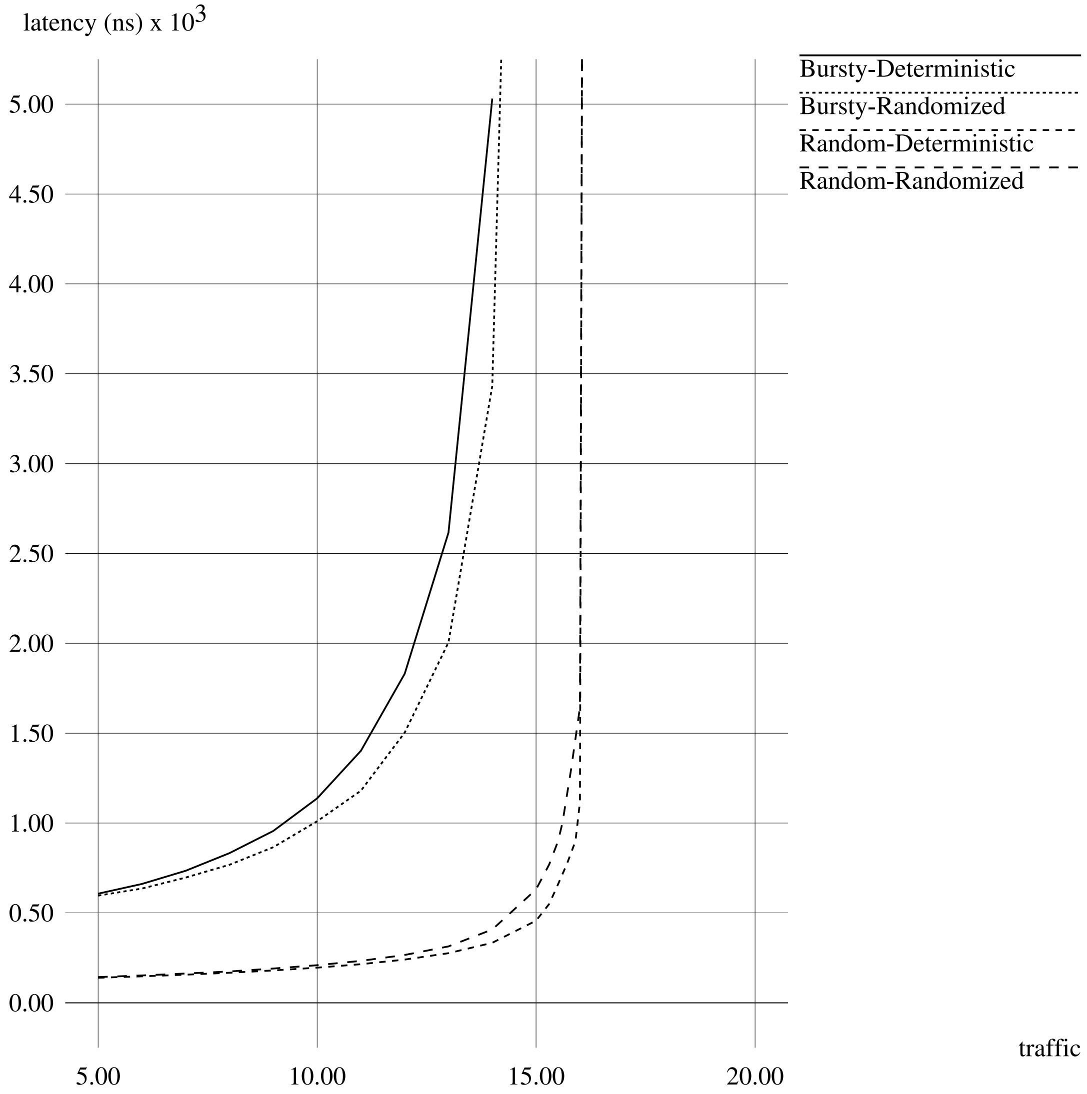


Figure 11: Star with trunk link multiplicity = 3: randomized vs. deterministic routing under bursty and uniform random traffic

As the next logical case, let us consider the star configuration with trunk link multiplicity equal to 4 as shown in Figure 12. This configuration allows to connect 48 terminal nodes; in this configuration, there are no terminal nodes on the hub switch 5.

Figure 13 shows the overall fabric performance using different routing strategies and workloads. The results show that the deterministic routing strategy again outperforms the randomized strategy for both workloads.

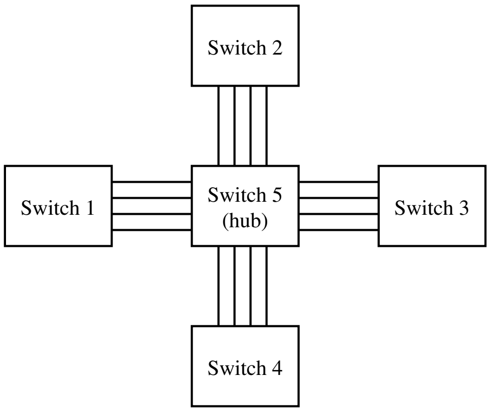


Figure 12: Star configuration with trunk link multiplicity = 4

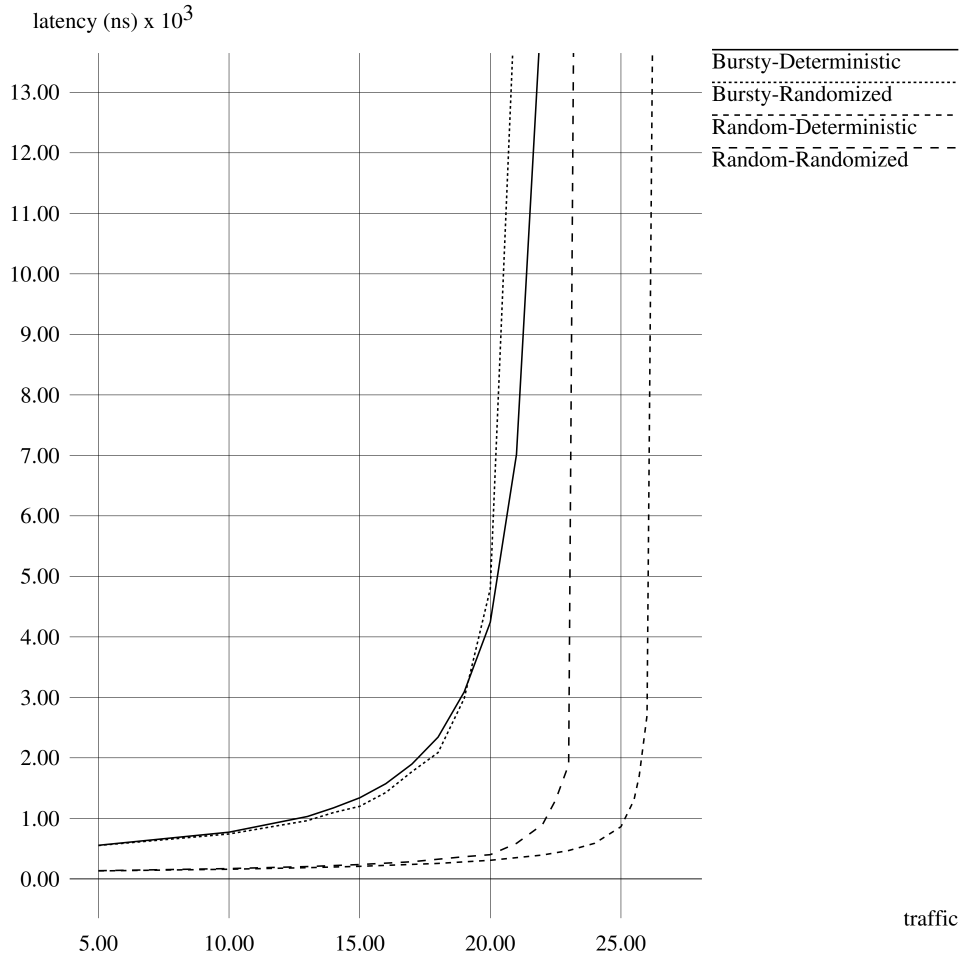


Figure 13: Star with trunk link multiplicity = 4: randomized vs. deterministic routing under bursty and uniform random traffic

The choice of topology for cascading a fixed number of switches is defined by two basic parameters: how many terminal nodes the fabric should connect, and what an acceptable system throughput might be. For a fixed value of either parameter, there is some topology that optimizes the other parameter. In general, higher throughput can be traded off against a higher number of terminal nodes. As an example, the star configurations with different trunk link multiplicities are shown together in Figure 14 under bursty workload with 10% long messages using the deterministic routing strategy.

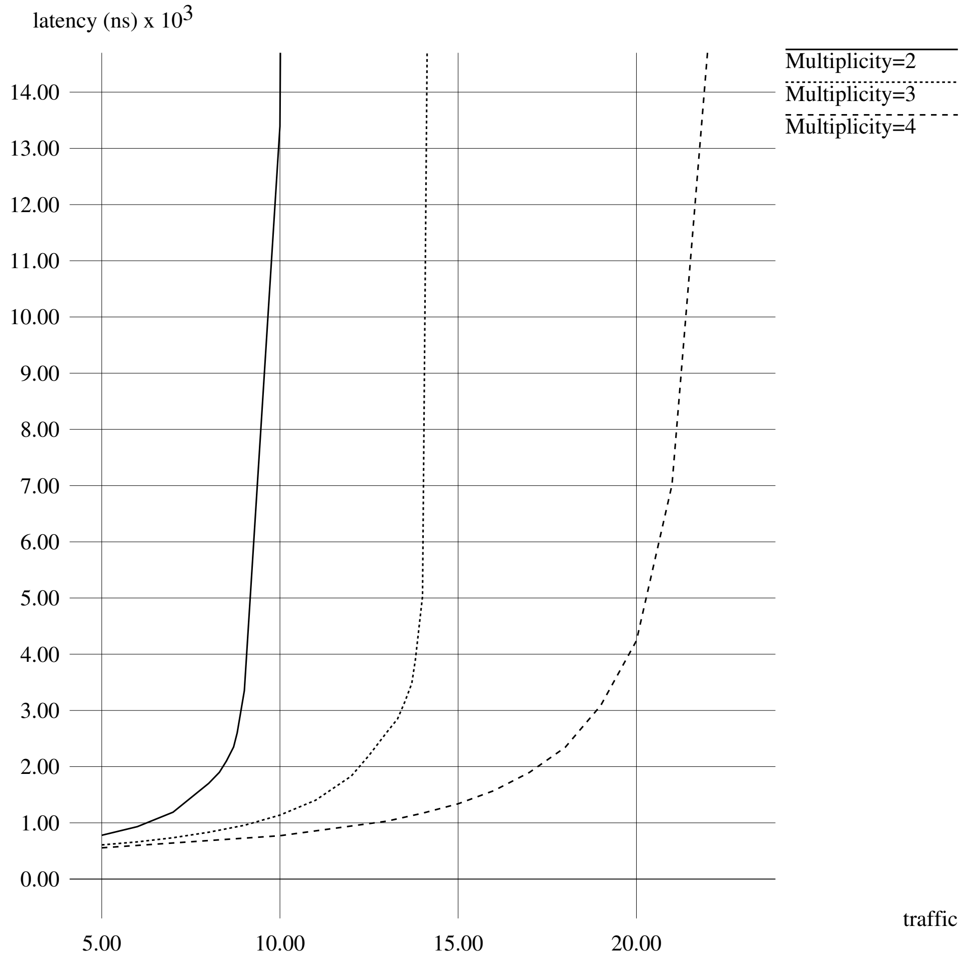


Figure 14: Star with different trunk link multiplicity under bursty workload

As Figure 14 shows, the star configuration with trunk link multiplicity 2 connects 64 terminal nodes. However the system performance is very poor: at most, it is about 10% of the FC port bandwidth. The star configuration with trunk link multiplicity 3 connects 56 terminal nodes. The system performance improves somewhat, to a potential maximum of about 14% of the port bandwidth. Finally, the star configuration with trunk link multiplicity 4, connecting 48 terminal nodes, has a somewhat better overall performance, of up to 22% of the port bandwidth.

The simulation results for the star configurations with respect to routing strategies can be split in two groups. With trunk link multiplicity equal to 2 or 4, the deterministic routing strategy outperformes the randomized strategy. However, the simulation results for a trunk link multiplicity of 3 show that randomized routing strategy outperforms the deterministic one. The explanation involves how the routing table for each switch distributes the final destinations over the different trunk links.

In case of the star configuration with trunk link multiplicity 2, there are 14 terminal nodes on each of the switches 1, 2, 3 and 4. These terminal nodes can be equally distributed among the 2 trunk links by assigning 7 terminal nodes to each. There are 8 terminal nodes on the hub switch 5 which can also be equally distributed among 2 trunk links. A similar situation exists with multiplicity four, where 12 terminal nodes can be distributed among 4 trunk links.

However, the case of multiplicity 3 is different. There are 13 terminal nodes on switches 1, 2, 3 and 4; these cannot be evenly distributed among 3 trunk links. The best we can do is assign four to each of two trunk links, and five to the remaining ones. The resulting distribution is not balanced: some of the trunk links serve more terminal nodes than the other ones, and thus get more traffic.

The overall system performance is defined by the performance of its weakest link. This explains why deterministic strategy advantages disappear in case of uneven (unbalanced) distribution of terminal nodes among multiple trunk links.

Finally, it is useful to notice that the conclusion on deterministic strategy advantages shown here is valid under a switch with FIFO scheduling. More aggressive switch scheduling strategies can change these results.

### 4.1.2 The House Configuration

Another interesting configuration is the house configuration which is shown in Figure 15.

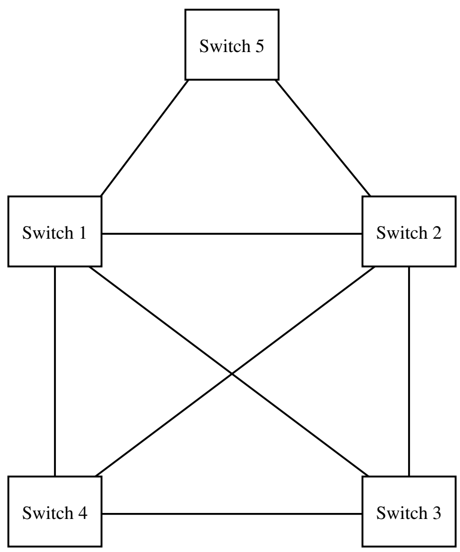


Figure 15: The house configuration

With trunk link multiplicity equal to 1 between the switches, the configuration supports 64 terminal nodes: 14 terminal nodes on a switch 5, 13 terminal nodes on switches 3 and 4, and

12 terminal nodes on switches 1 and 2.

Figure 16 shows the message latency for the fabric using the house configuration under bursty and uniform random traffic.

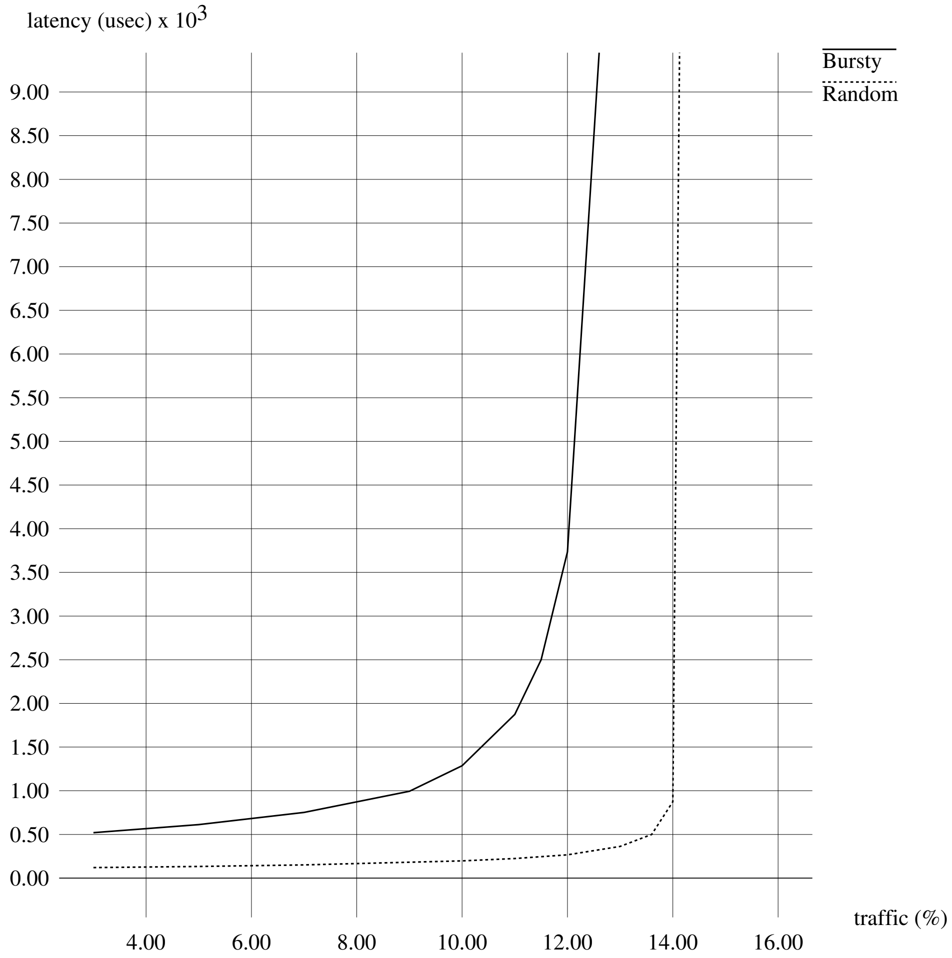


Figure 16: The house configuration under bursty and uniform random traffic

Both the house configuration considered above and star configuration with trunk link multiplicity equal to 2 connect 64 terminal nodes. However, the house topology provides higher fabric performance than the star topology. The overall fabric throughput increases to 13% of the port bandwidth for the house configuration, compared against 10% of the port bandwidth supported by the star configuration, assuming bursty traffic with 10% long messages. For random traffic, the increase is from 11% to 15%.

To collect more statistics to analyse the advantages of randomized vs deterministic routing strategies, let us consider the house configuration with multiplicity of trunk links connecting switches 1,2,3,4 and 5 equal to 2. This configuration connects 48 terminal nodes: 12 terminal nodes on a switch 5, 10 terminal nodes on switches 3 and 4, and 8 terminal nodes on switches 1 and 2. This configuration is shown in Figure 17.

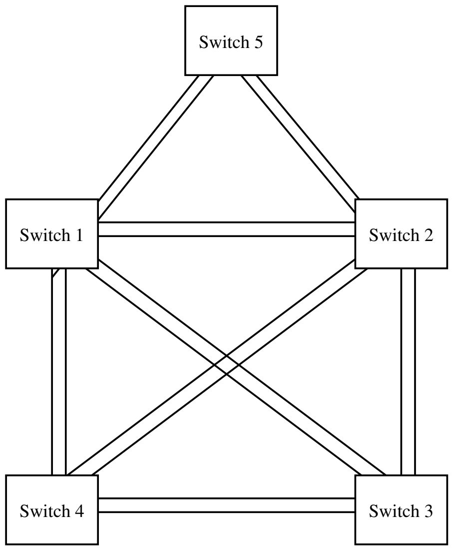


Figure 17: The house configuration with trunk link multiplicity = 2

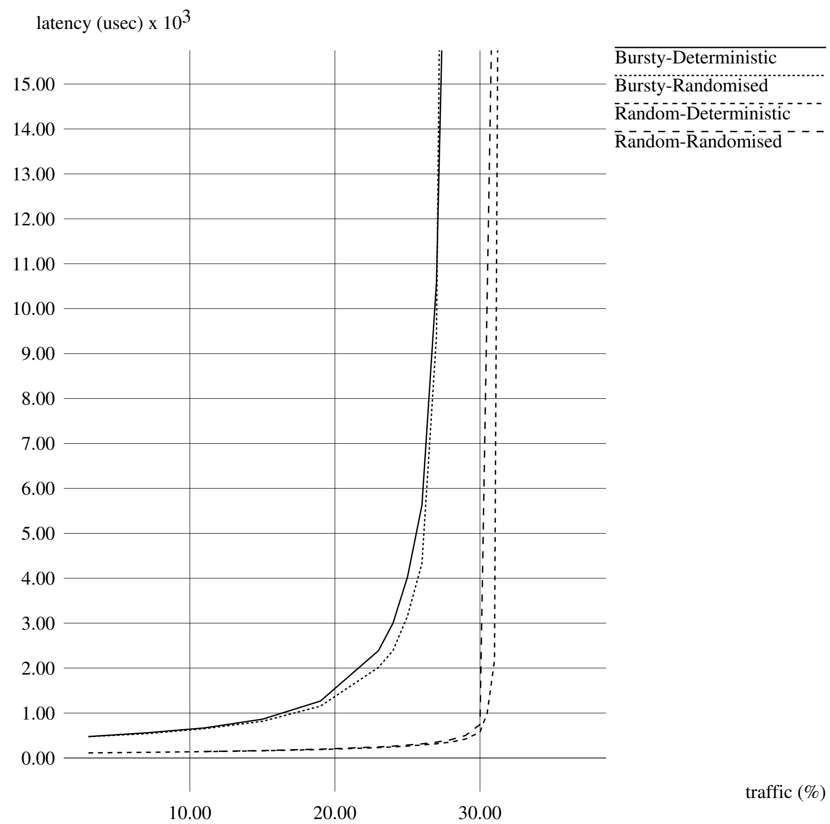


Figure 18: The house configuration with trunk link multiplicity = 2: randomized vs. deterministic routing under bursty and uniform random traffic

The simulations were performed for our two different kind of workloads. Figure 18 summarizes the overall fabric performance using different routing strategies and workloads. These results show that the fabric performance using deterministic routing strategy vs. randomized strategy practically is indistinguishable for both workloads.

#### 4.1.3 The Envelope Configuration

Another fabric configuration, called the envelope, is shown in Figure 19. All trunk links connecting switches 1, 2, 3, 4 and 5 have a multiplicity equal to 1. This configuration connects 64 terminal nodes: 13 terminal nodes on switches 1, 2, 3 and 4, and 12 terminal nodes on switch 5.

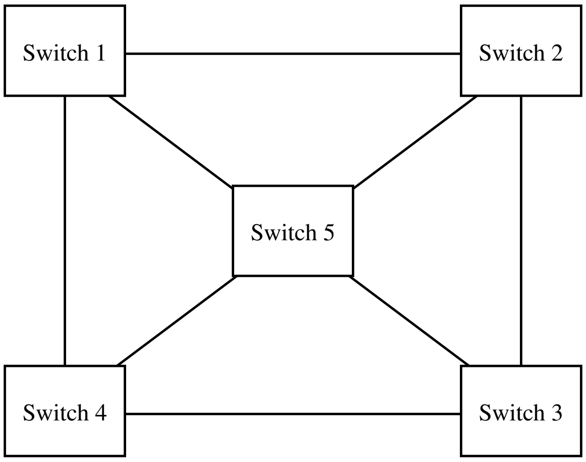


Figure 19: The envelope configuration

Figure 20 shows the message latency using the envelope, house, and star configurations under bursty workload with 10% long messages.

All three fabric configurations connect 64 terminal nodes, and are thus directly comparable. However the envelope topology provides essentially higher fabric performance than both house and star topology.

While trunk multiplicity in the star network helps improve fabric performance by providing a wider data transfer channel, that extra trunk capacity is better utilized by connecting more switches, lowering the average distance between switches.

As an example, compare the envelope topology with the star topology. The overall fabric throughput increases from 10% of the port bandwidth (for star) to 16% of the port bandwidth (for envelope). The envelope topology is the overall optimal topology when connecting 64 terminal nodes with 5 switches.

To collect more statistics to analyse the advantages of randomized vs. deterministic routing strategies, let us consider the envelope topology with a trunk link multiplicity of two; this configuration connects 48 terminal nodes. This configuration is shown in Figure 21.

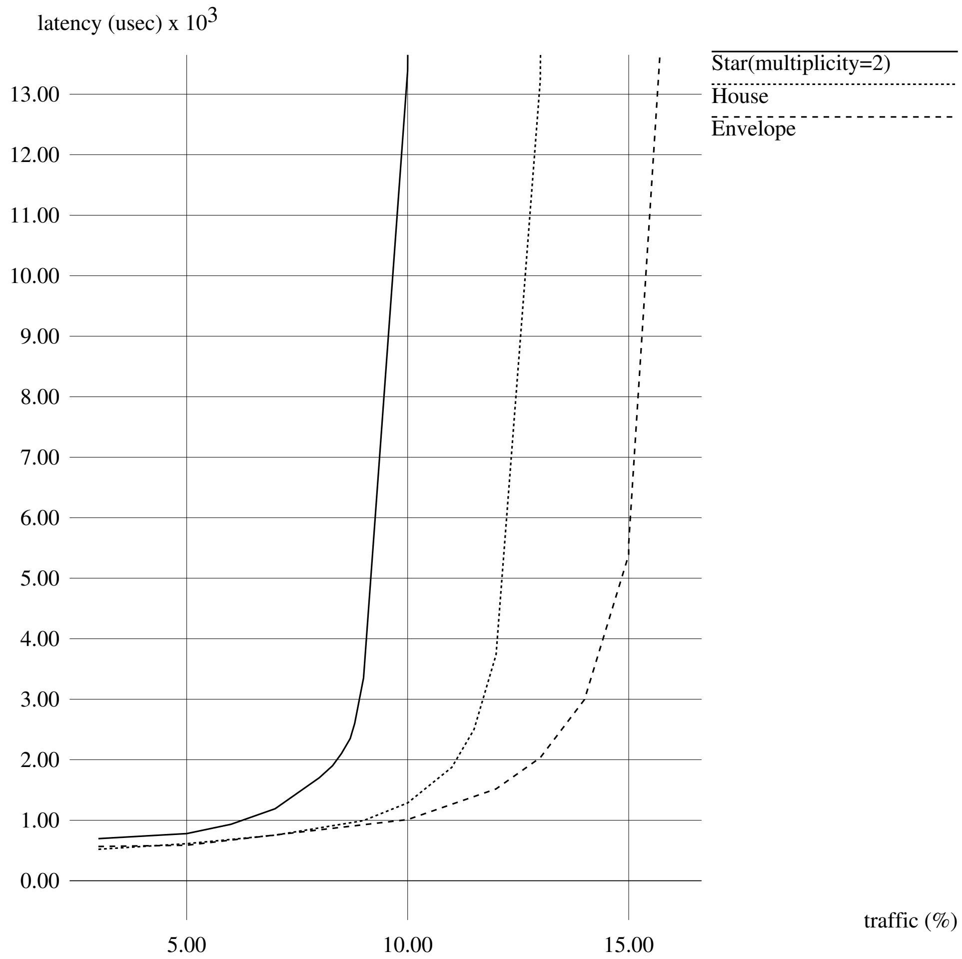


Figure 20: The envelope, house, and star configurations with 64 terminal nodes: message latency under bursty traffic

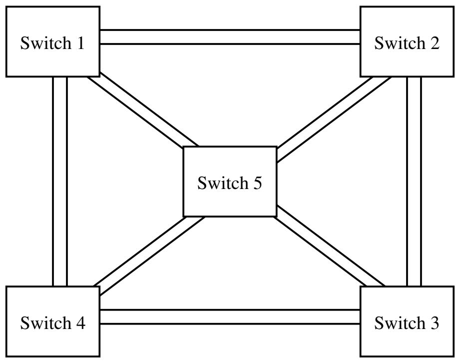


Figure 21: The envelope configuration with trunk link multiplicity = 2

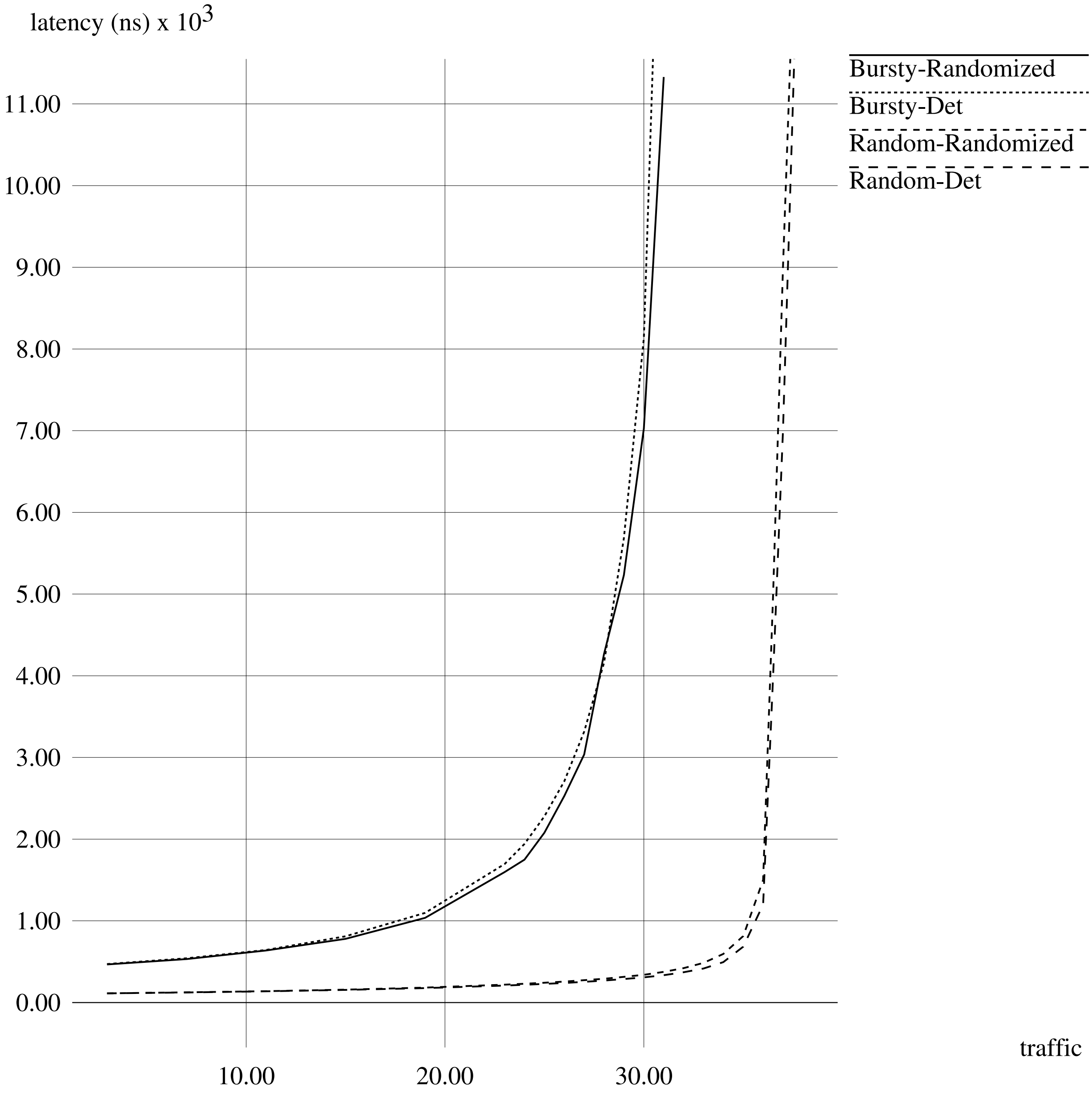


Figure 22: The envelope configuration with trunk link multiplicity = 2: randomized vs. deterministic routing under bursty and uniform random traffic

Figure 22 shows the overall fabric performance using different routing strategies and workloads. These results show that the fabric performance using deterministic routing strategy vs. randomized strategy is practically indistinguishable for both workloads.

In conclusion, which routing strategy should be chosen depends on the implementation cost as well as additional goals such as deadlock-freeness and fairness.

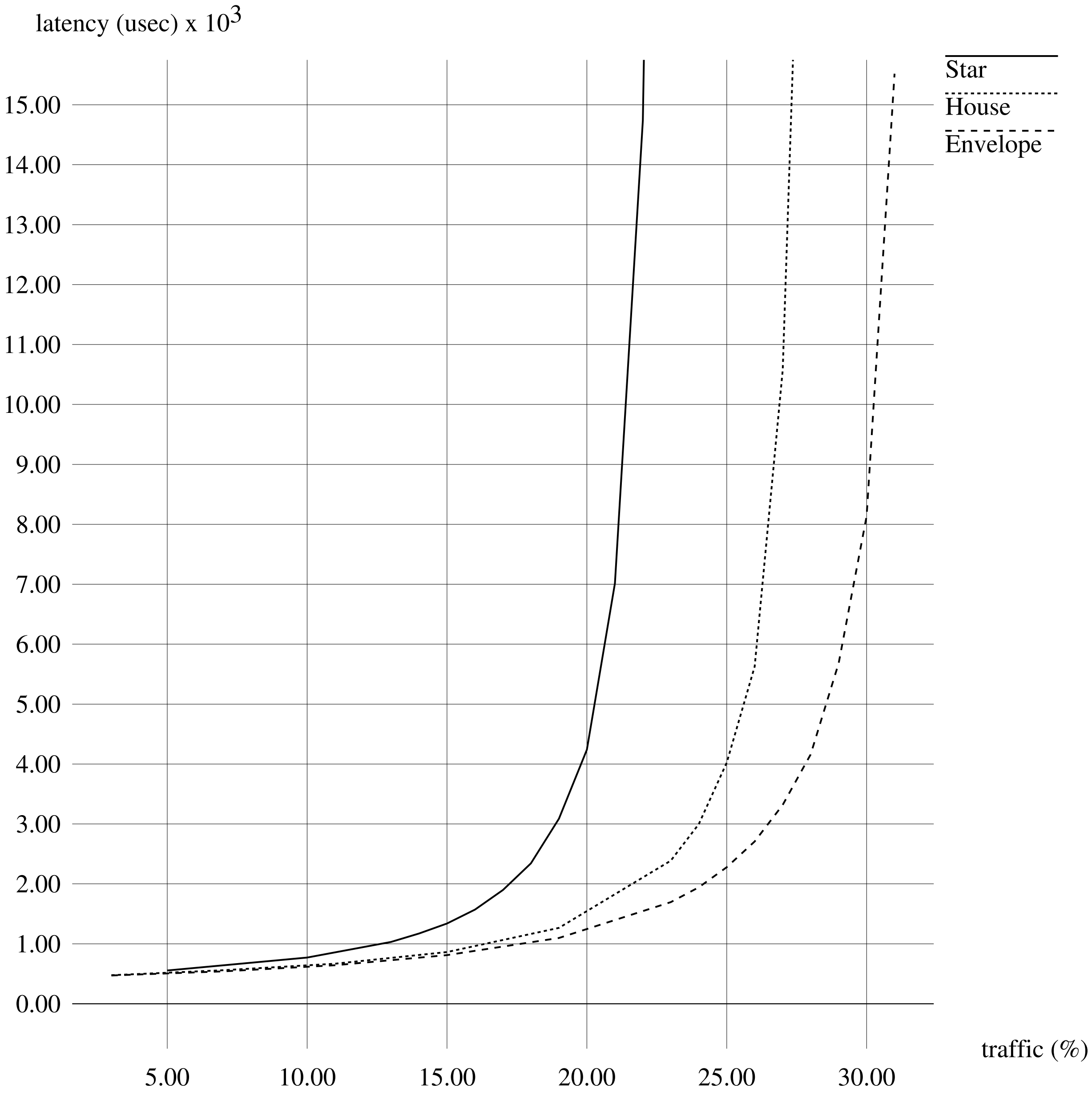


Figure 23: The envelope, house, and star configurations with 48 terminal nodes: message latency under bursty traffic

Figure 23 shows the message latency for the envelope, house, and star configurations connecting 48 terminal nodes under our bursty workload with 10% long messages. Again, the envelope topology provides essentially higher fabric performance than both the house and star topology.

## 4.2 Fairness and Deadlock

When choosing a topology for cascading FC switches there are two additional aspects to consider. The first aspect is whether a topology could lead to a *deadlock*. This question is not a subject of this report, the following up report [CKR95-2] is devoted particular to this problem. The only topology subject to deadlock that we mention in this report is the envelope topology (which turns out to have the best performance); it is possible to route this topology in such a way that deadlocks do not arise.

The second aspect is *fairness*: whether the frames sent from each terminal node are treated fairly when competing for resources such as trunk links.

We will illustrate both of the problems using examples.

The only topology considered in this report which could lead to a deadlock is the envelope topology. By deadlock we mean here the situation when some cycle of trunk link buffers are filled with frames that need to be transferred forward in the cycle. Since all buffers are full, no individual frame can be forwarded, causing deadlock. For example, suppose that all the terminal nodes on switch 1 are sending frames to the terminal nodes on switch 3 and vice versa, as well as all the terminal nodes on switch 2 are sending frames to the terminal nodes on switch 4 and vice versa. Let us assume that the frames are routed to their destinations in a clock wise manner: the frames originating on switch 1 to switch 3 are sent via switch 2, the frames originating on switch 3 to switch 1 are sent via switch 4 etc. This routing quickly leads to a deadlock when all the switch 2 buffers on a trunk link from switch 1 to switch 2 (1–2 trunk link) are filled with frames sent to switch 3, all the switch 3 buffers on a 2–3 trunk link are filled with frames sent to switch 4 etc. That is, the respective trunk link buffers are filled with in-transit frames which can not be forwarded since the next switch does not have any available buffers to accept them.

This situation can be corrected by using a different routing scheme. If we change the routing so that all the frames with destinations 2 hops away (as in the previous considered case) are routed via central switch 5, then the routing scheme is deadlock free. Unfortunately, this routing also tends to concentrate traffic in the links connected to switch 5, so they become the bottleneck prematurely. Thus, this routing is unbalanced, leading to worse performance. It is possible to derive a deadlock-free, balanced routing for this topology; we call such a routing scheme a "smart" routing. It turns out that, for most optimal network topologies, it is possible to generate a "smart" routing scheme.

Another major difficulty with some topologies is unfairness. Consider a star topology with trunk link multiplicity of 1 shown in Figure 24.

Let us consider frames sent by terminal nodes on switch 1 (called satellite frames) to some terminal node on a switch 3. Additionally, we will consider frames sent by the terminal nodes on a hub switch 5 (called hub frames) also to some terminal node on a switch 3. Satellite and hub frames compete for the trunk link between switches 3 and 5 (the 3–5 trunk link). The question is: how much of 3–5 trunk link capacity do the satellite frames receive, vs. the capacity that hub frames receive? To reach hub switch 5, the satellite frames use the trunk link between switches 1 and 5 (the 1–5 trunk link). Thus, all satellite frames from the 15 terminal nodes on switch 1 are stored (when sent to hub switch 5) in buffer slots of the 1–5 trunk link. Each node (or trunk link) on switch 5 has an equal probability of obtaining the 3–5 trunk link of 1/15. Since the 1–5 trunk link represents the chances of all 15 terminal nodes on switch 1, the overall chance of satellite frames from a particular switch 1 terminal node is

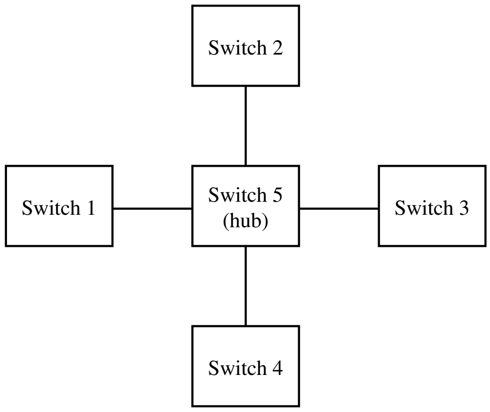


Figure 24: The star configuration with trunk link multiplicity = 1

1/225, vs. 1/15 for satellite frames from switch 5 terminal nodes. Thus, traffic injected by hub terminal nodes have an unfair advantage over traffic generated by satellite terminal nodes in obtaining trunk links, by a factor of 15 to 1. This proportion gives such an advantage to the hub terminal nodes that could lead to almost complete starvation of the sattelite terminal nodes.

The star topology with trunk link multiplicity of 2 shown in Figure 7 has a similar unfairness problem among its terminal nodes. Hub terminal nodes win against satellite terminal nodes in obtaining the trunk link by a proportion of 7 to 1.

In order to avoid unfairness in the star configuration, it is better to not attach any terminal nodes to a hub switch. The problem of unfairness and some possible solutions could be found in more detail in [CKR95-3].

## 4.3 Complete Topology

In this Section, to complete the performance evaluation study for connecting 5 Fibre Channel switches in a fabric with about 64 Nodes, the complete topology will be considered and compared against star, house and envelope. Complete topology has exactly one trunk link to connect each pair of switches. In such a way, this configuration allows to connect 60 terminal nodes: 12 terminal nodes per each switch.

Figure 25 shows the message latency for the FC fabric using different topologies under bursty workload with 10% long messages.

The envelope topology provides highest fabric performance for the configuration with 64 terminal nodes. However, sacrifying only 4 additional terminal nodes (complete topology with 5 switches connects 60 terminal nodes), FC fabric cascaded using complete topology could almost double its performance: the overall fabric throughput increases up to 26%.

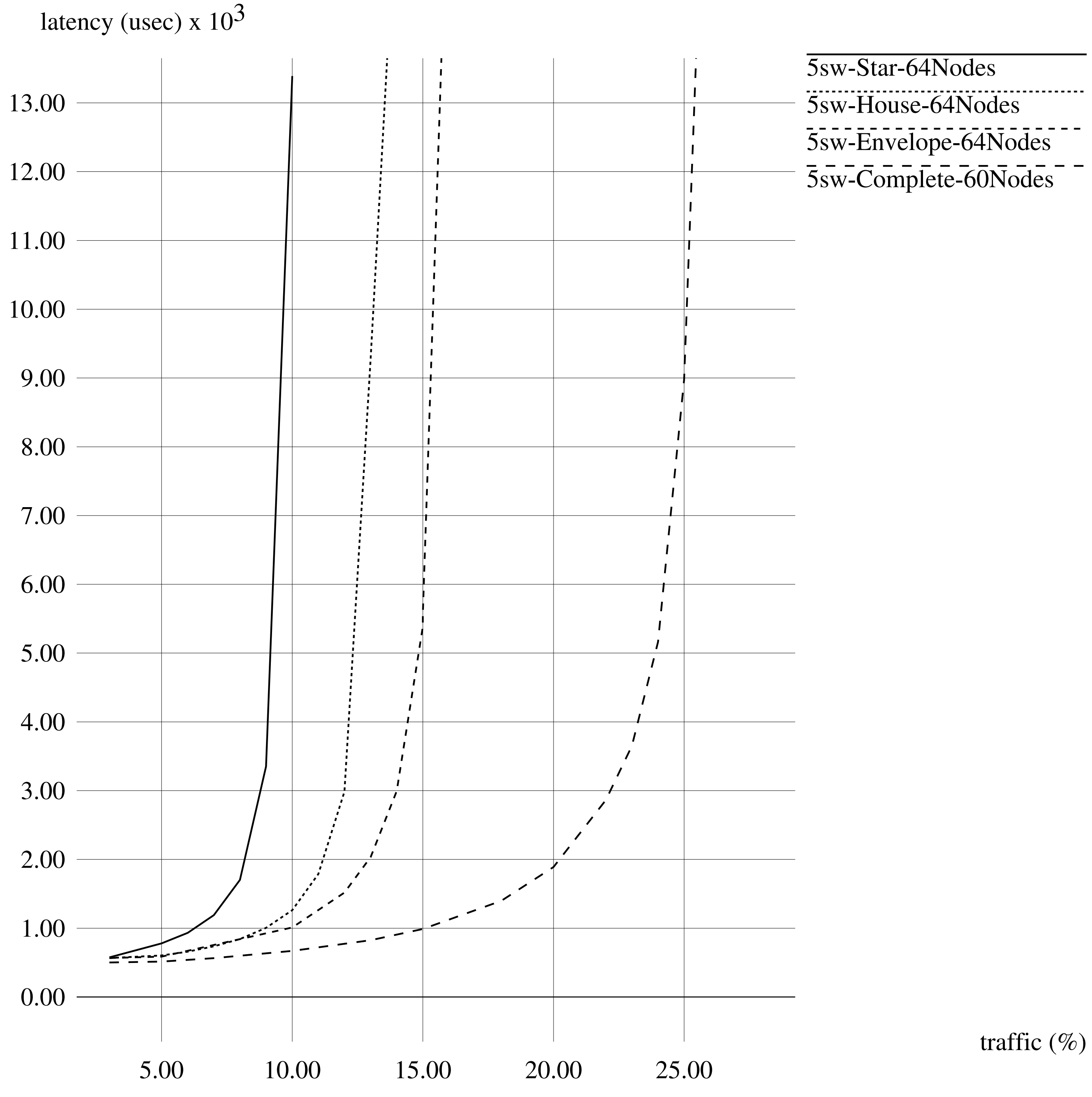


Figure 25: Connecting 5 switches in a fabric with about 64 nodes (performance under the bursty traffic

# 5 The Complete Topology: Its Power and Limitations

In this section we extend our study of FC fabrics by considering complete topologies for 6, 7, and 8 switches. We will also project the complete topology fabric performance for 9, 10, and 11 switches.

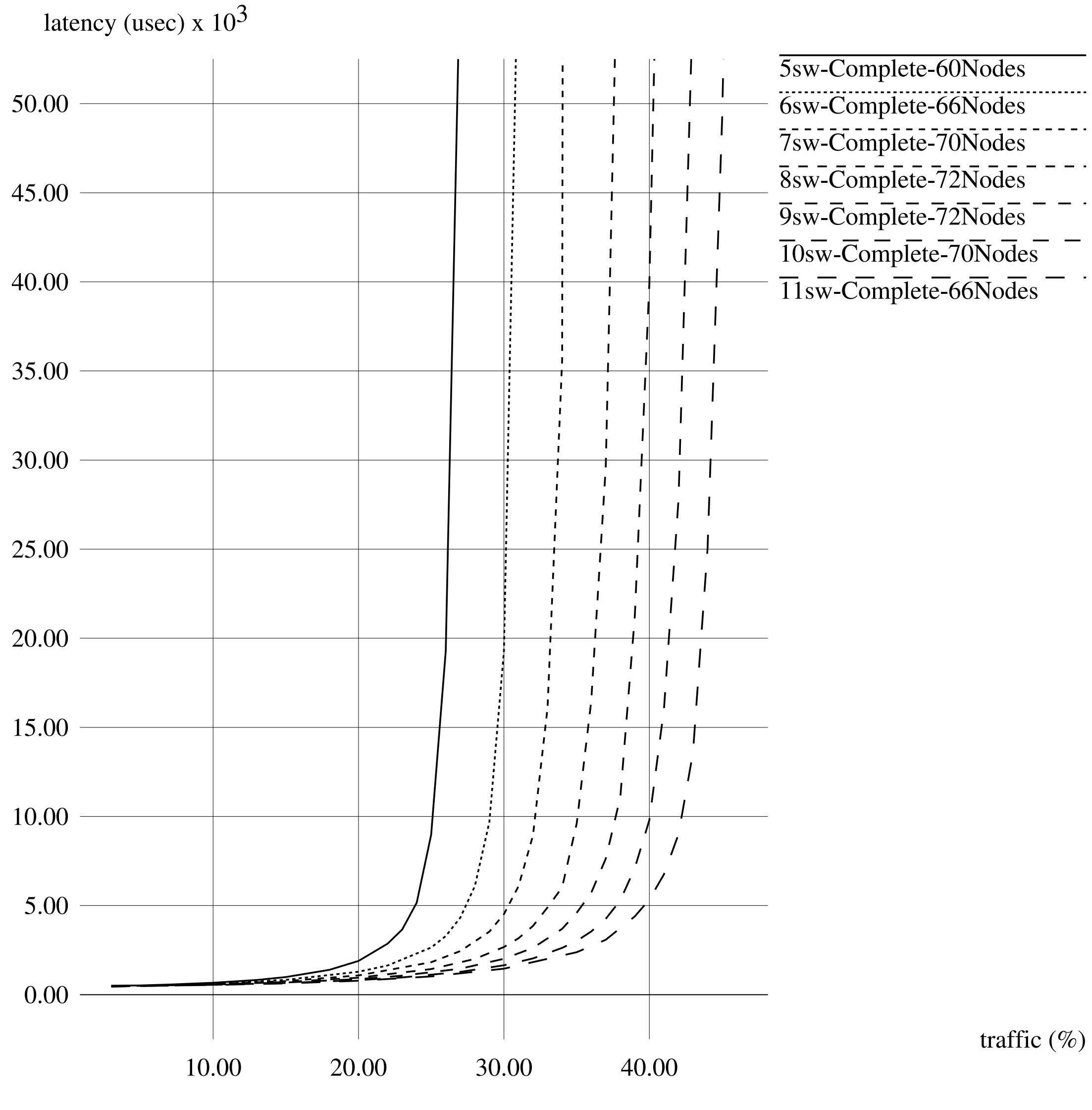


Figure 26: Cascading from 5 to 11 switches using the complete topology (under the bursty traffic)

A complete topology with 6 switches connects 66 terminal nodes, with 7 switches, 70 terminal nodes, and with 8 switches, 72 terminal nodes. Increasing the number of switches does not increase the number of connected terminal nodes: a complete topology with 9 switches connects 72 terminal nodes, with 10 switches, 70 terminal nodes, and with 11 switches, 66 terminal nodes. So, the maximum number of terminal nodes supported by a complete topology made out of 16-port switches is 72 terminal nodes. This configuration can be implemented with either 8 or 9 switches.

A complete topology with 8 switches has the trunk links as a system bottleneck, while for 9 and larger complete fabrics, the switches themselves tend to be the bottleneck. As the number of switches increase, even beyond 9 and 10, the overall performance increases, since the aggregate traffic is distributed among more switches.

Another interesting observation is related to the role of end-to-end credits for different configurations. Figure 26 shows the message latency for FC fabric cascaded using different topologies under a bursty workload with 10% long messages.

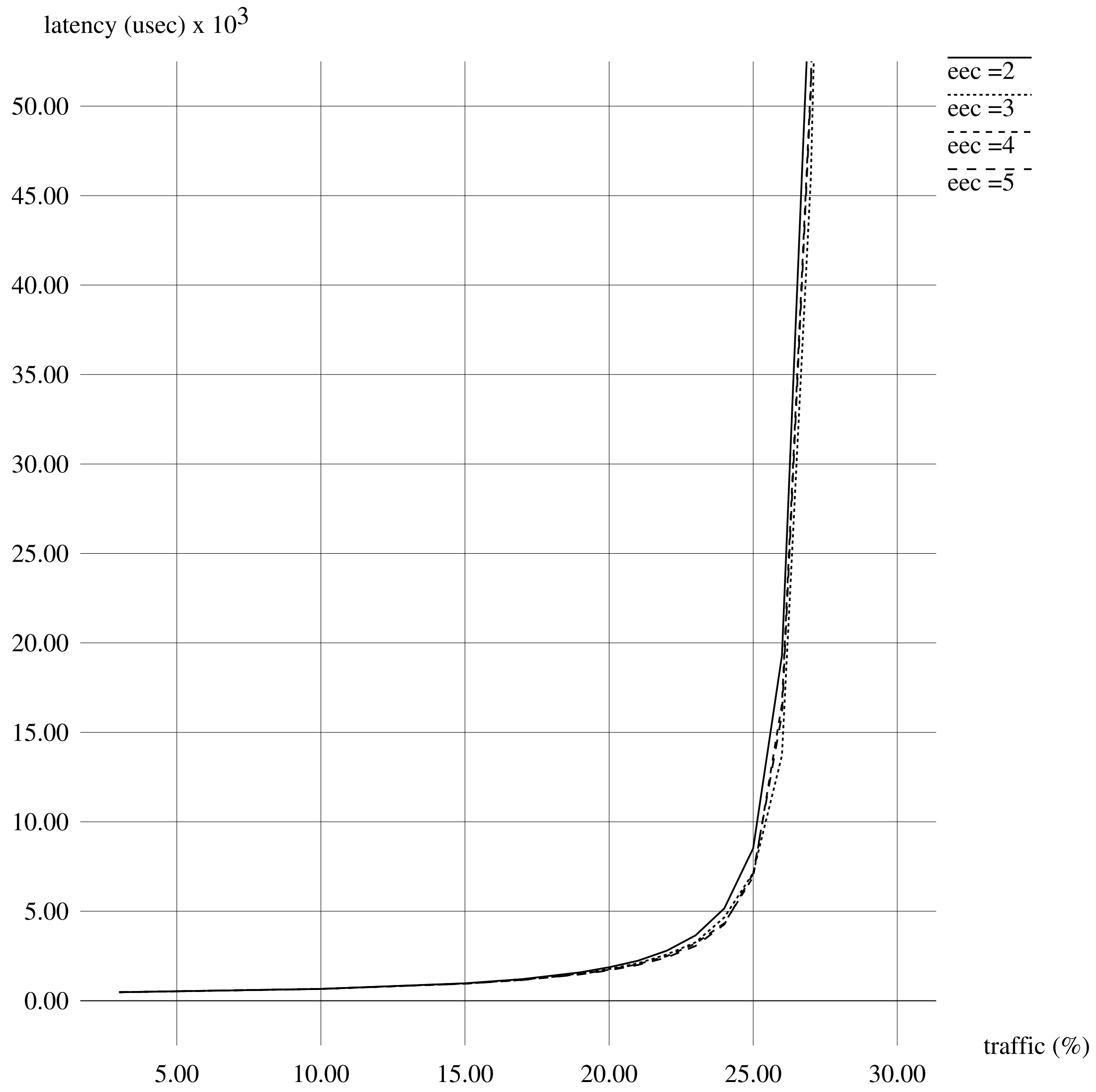


Figure 27: The complete topology with 5 switches under the bursty traffic and different values of end-to-end credits

The complete topology with 5 switches has the trunk links as the major system bottleneck.

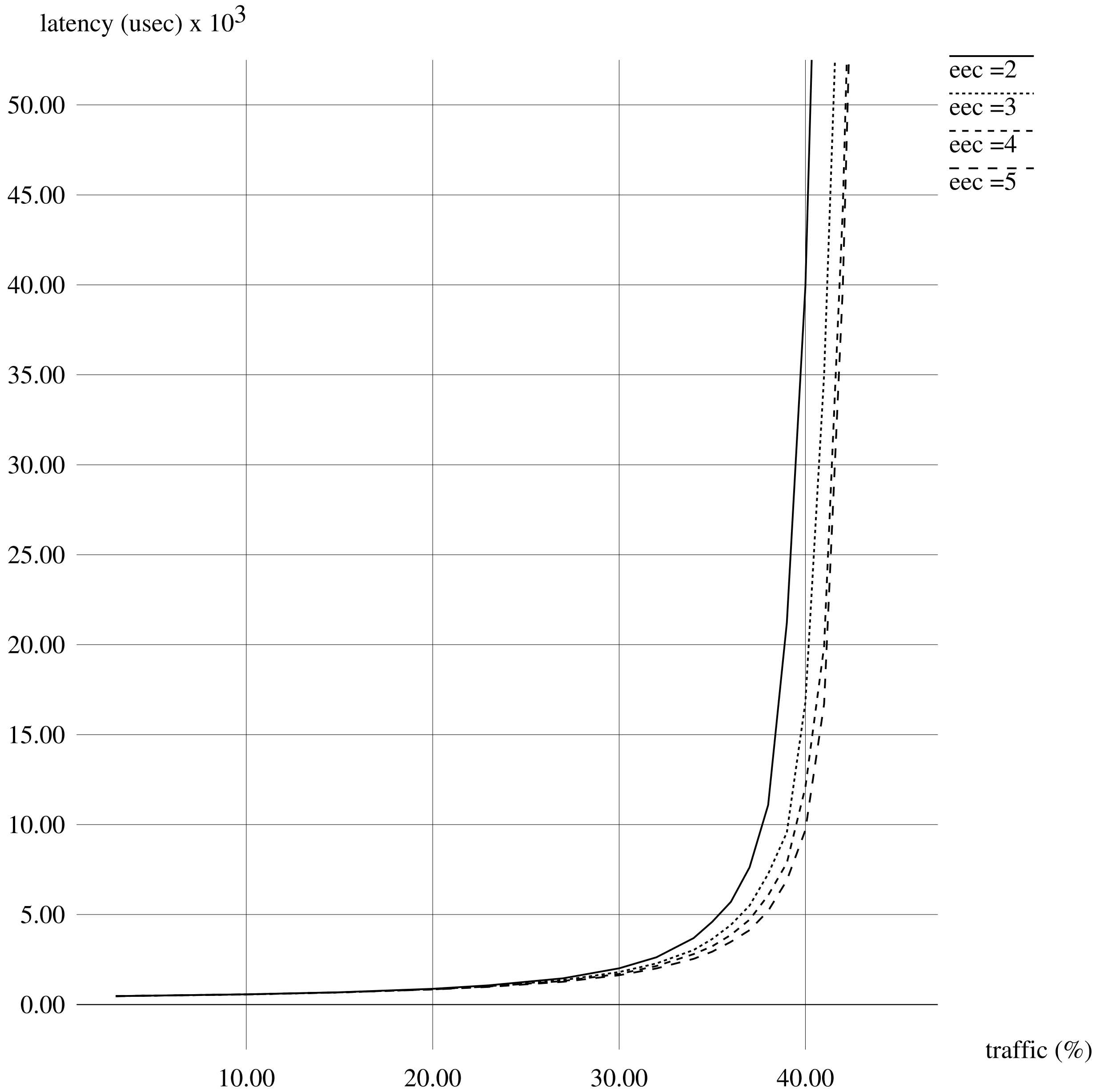


Figure 28: Complete topology with 9 switches under the bursty traffic and different numbers of end-to-end credits

In this situation, tuning the end-to-end credits value does not influence the overall fabric performance. Figure 27 shows the message latency and throughput for fabrics using the complete topology with 5 switches under a bursty workload with 10% long messages. Both message latency and throughput do not depend on the end-to-end credit value.

However, the situation changes once the trunk links are not system bottlenecks any more. By

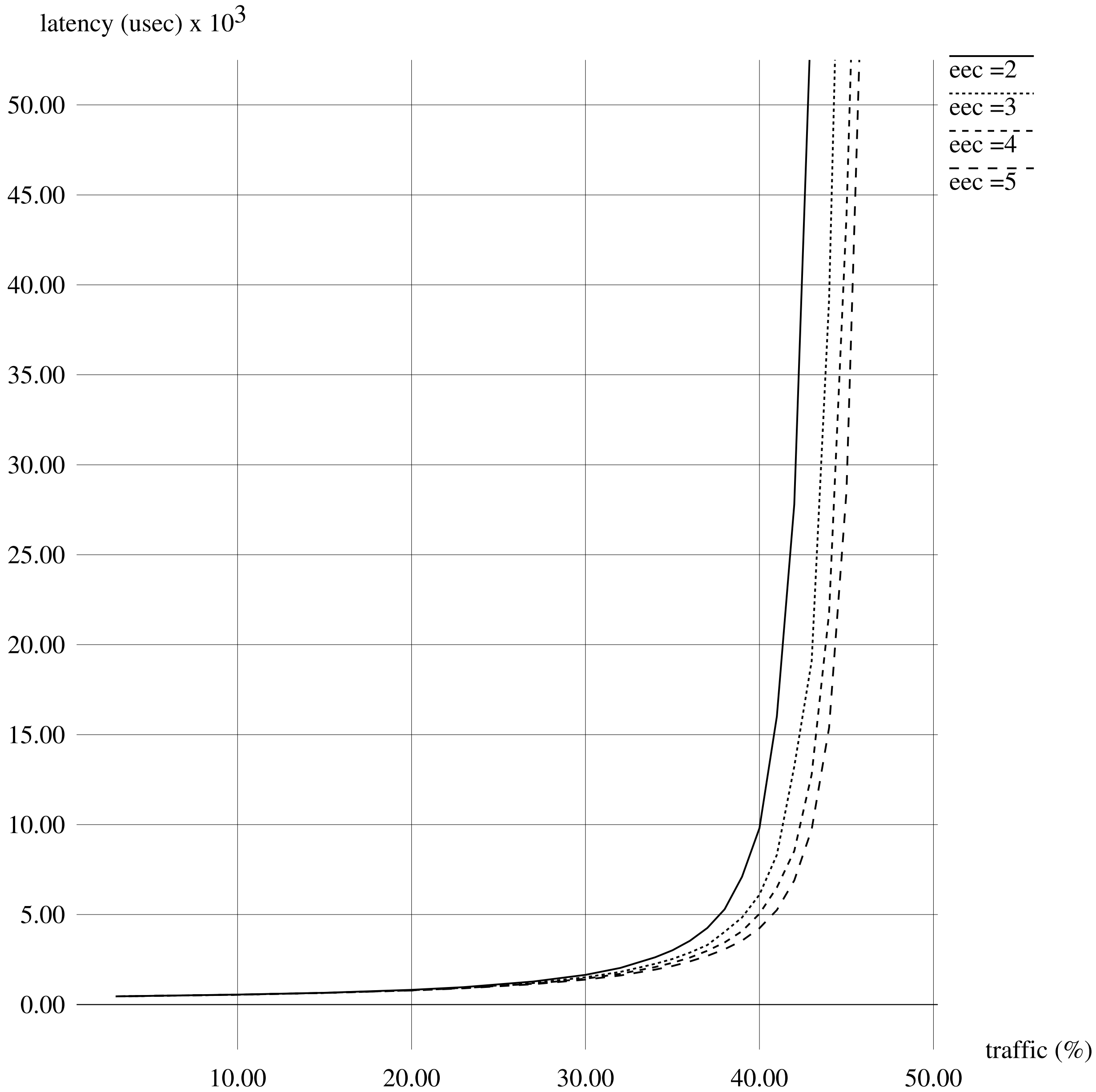


Figure 29: Complete topology with 10 switches under the bursty traffic and different numbers of end-to-end credits

increasing the number of end-to-end credits, the overall fabric throughput can be improved by about 10%. Figures 28, 29, 30 show the FC fabric performance using complete topology with 9, 10, and 11 switches, respectively, under our bursty workload with 10% long messages. The end-to-end credit value varies from 2 to 5.

The performance improvement derives primarily due to the pipelining between the data frames

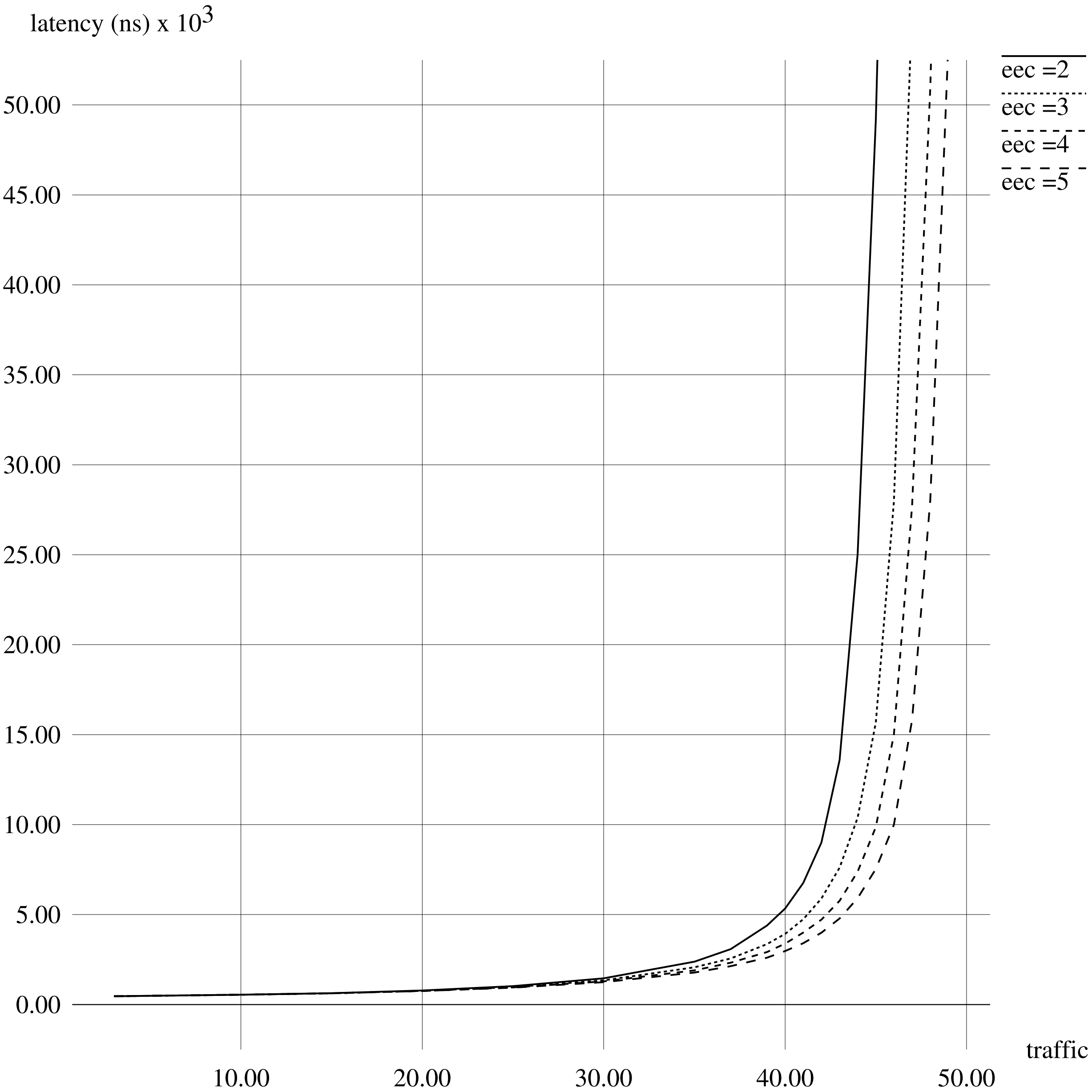


Figure 30: Complete topology with 11 switches under bursty traffic and different numbers of end-to-end credits

and the acknowledgement frames that occurs inside the fabric. By increasing the number of end-to-end credits, more pipelining effect can be exploited. For other cascading topologies with larger path lengths, this effect can be even more pronounced.

## 6 Conclusion

In this report, we concentrated on cascading FC switches with 16 ports. We analyzed the basic performance of a single Fibre Channel switch. We considered two different workloads: random and bursty traffic. The main switch bottleneck is due to FIFO scheduling of the output ports queues and the resulting head of line blocking. We investigated how the performance of the switch depended on the class of service used.

Next, we considered the performance analysis of a few basic topologies for cascading FC switches, with a target of approximately 64 terminal nodes. The topologies we considered were the star, house, envelope and complete configurations. Typically, such topologies have their trunk links as the bottlenecks. It is possible to sacrifice few additional terminal nodes in order to increase fabric throughput by putting multiple trunk links between the switches.

Multiple trunk links introduce the possibility using a randomized routing strategy. With the randomized strategy, routing choice between trunk links is done by choosing randomly among the possible routes, while the deterministic strategy is based on a fixed partitioning of the destination terminal nodes to trunk links. The simulation results show that in some cases, the deterministic routing strategy is slightly more efficient than the randomized one. However, there exist topologies when the randomized routing strategy slightly outperforms the deterministic one (typically, for topologies with uneven distribution of terminal nodes). And finally, for some topologies these two routing strategies are practically indistinguishable.

In considering these different topologies, we illustrate a set of problems that occur. We determine that trunk link multiplicity should only be used after each switch has a trunk link connection to every other switch.

The choice topology for cascading FC switches should consider two different sets of parameters:

- *quantative* characterization based on
  a) a number of switches and terminal nodes and
  b) fabric throughput and latency;
- *qualitative* characterization of the properties
  a) whether the topology has a deadlock-free routing and
  b) whether the topology is fair.

This report does not consider the effects of modifications inside the switch to improve frame scheduling or throughput, nor does it consider intelligent scheduling of frames from outside the switch. Either approach would increase throughput, and both would yield near-optimal throughput, as shown by some early results [CKR94, CR94].

## 7 Acknowledgements

Many thanks are due to Suhas Badve and Robin Purohit as well as their teams for showing interesting problems and asking good questions, and encouraging support during our work on Fibre Channel performance evaluation study.

# 8 References


[AC92] Anderson, T. and Cornelius, R.: High-Performance Switch with Fibre Channel, Digest of Papers, IEEE COMPCON, February, 1992.

[CKR94] Cherkasova, L., Kotov, V., Rokicki, T.: On the Effect of Message Scheduling for Packet Switching Fabrics. HPL-94-70, August, 1994.

[CR94] Cherkasova, L. and Rokicki, T.: Alpha Message Scheduling for Packet-Switched Interconnects. HPL-94-71, August, 1994.

[CKR95-1] Cherkasova, L., Kotov, V., Rokicki, T.: Evaluation of Network Topologies. HPL Report, to be published.

[CKR95-2] Cherkasova, L., Kotov, V., Rokicki, T.: Smart Routing: Deadlock-free, Balanced Routing for Fibre Channel Fabrics. HPL Report, to be published.

[CKR95-3] Cherkasova, L., Kotov, V., Rokicki, T.: Global Unfairness in Networks ofLocally Fair Switches. HPL Report, to be published.

[Fujimoto83] Fujimoto R. M. VLSI Communication Components for Multicomputer Networks.Ph.D. Thesis, University of California at Berkeley, August 1983.

[GR92] Getchel, D. and Rupert, P. Fibre Channel in the Local Area Network. IEEE LTS, Vol.3, No. 2, May, 1992, pp.38-42.