Design and Simulation of a New Queuing Architecture for Large-Scale ATM Switches
Mohsen Guizani and Ala Isam Al-Fuqaha

SIMULATION 2002; 78: 431
DOI: 10.1177/0037549702078007581

The online version of this article can be found at:
http://sim.sagepub.com/cgi/content/abstract/78/7/431
Design and Simulation of a New Queuing Architecture for Large-Scale ATM Switches

Mohsen Guizani
Computer Science Department
University of West Florida
mguizani@cs.uwf.edu

Ala Isam Al-Fuqaha
FirstWave Optics, Inc.
Maryland

The authors study the different buffering techniques used in the literature to solve the contention problem in asynchronous transfer mode (ATM) switching architectures. The objective of this study is to determine the buffer requirements needed to achieve a given quality of service (e.g., a given cell loss probability). On the basis of this study, the authors propose a combined central and output queuing (CCOQ) technique to be used in designing large-scale ATM switches. Also, a general design technique for an $N \times N$ large-scale ATM switch is proposed with a suitable CCOQ buffer size to reduce both the cell loss probability and the complexity of the memory modules. The switch has to be designed such that it can be implemented using the smallest number of VLSI chips possible. It should be also reliable for commercial use. The switch should support multicast and priority control functions.

Keywords: Quality of service, asynchronous transfer mode, central buffering, output buffering, cell loss probability

1. Introduction

The asynchronous transfer mode (ATM) has been selected by the International Consultive Committee for Telephone and Telegraph (CCITT) as the standard transport and switching environment for the broadband integrated services digital network (B-ISDN). Thus, ATM has recently been the subject of intensive research interest. Particularly, the hardware design of ATM switching architectures has been studied extensively. Despite this extensive research on the ATM switching architectures, architectural innovation can still be achieved.

One of the important components of an ATM network is the ATM switch. The objective of our study is to determine the buffer requirements needed by an ATM switch to achieve a given quality of service (e.g., a given cell loss probability). On the basis of this study, we propose a combined central and output queuing (CCOQ) technique to be used in designing large-scale ATM switches. Also, we propose a general design technique for an $N \times N$ large-scale ATM switch with a suitable CCOQ buffer size to reduce both the cell loss probability and the complexity of the memory modules. The switch has to be designed such that it can be implemented using the smallest number of VLSI chips possible. It should be also reliable for commercial use. The switch should support multicast and priority control functions.

2. Difficulties with Large-Scale ATM Switches

ATM switches with a large number of input and output ports have some design difficulties. In the following, we discuss three main problems that arise as $n$ and $m$ increase for $n \times m$ shared-buffering ATM switches.

2.1 Memory Size

The waiting time for a shared queuing switching system is the same as that of an output queuing one [1]. However, multiple queues are combined in a single physical memory that is shared between all the input and output ports of the switching system. The major advantage of a shared queuing switching system is reflected in the number of cells to be stored in the shared memory (shared queue).

As the number of input and output ports of an ATM switch increases, the shared buffer memory (SBM) increases to achieve a desired cell loss probability. The maximum sharing effect of the memory (minimum required
shared buffer memory) is achieved when the whole \( n \times m \) switch is designed such that all the input and output ports can share a single SBM. It is difficult to design such a switch if we assume that the most critical design condition is the buffer size that needs to be accommodated in a single switching element. Another consequence of a large random access memory (RAM) is the relatively slow operating speed. Because of this, it is desirable to design a large-scale \( n \times m \) ATM switch using a smaller switching element as the basic building block such that the total amount of RAM that needs to be integrated in the switching element is small.

### 2.2 Memory Access Speed

If a single-port RAM is used as shown in Figure 1, then this memory has to be accessed \( n + m \) times during one cell time slot because each input and output port has to access the same memory simultaneously. \( n \) out of those \( n + m \) access times should be used to write up to \( n \) cells that could be coming on the \( n \) input ports. The remaining \( m \) of the \( n + m \) access times will be used to read up to one cell for each of the \( m \) output ports. Although the cell can be bit sliced and parallelized to reduce the memory access speed, there is a limit because the bus size and the cell length are limited. So, if a single-port RAM is used to implement the SBM, the access time can be expressed as follows:

\[
\text{Memory Access Time} = \frac{W}{(N + M) \cdot F},
\]

where \( W \) is the width of the memory data bus, \( N \) is the number of input ports, \( M \) is the number of output ports, and \( F \) is the bit rate of each input/output port.

If a dual-port memory is used to implement the SBM, this memory has to be accessed \( n + m \) times during one time slot because each input and output port has to access the same memory simultaneously. But since this memory can handle two accesses (one read and one write) simultaneously, then the access time of this memory can be expressed as follows:

\[
\text{Memory Access Time} = \frac{W}{\text{Maximum}(N, M) \cdot F}.
\]

So, the memory access time of a dual-port memory is more than that of a single-port memory. This implies that the access speed required from a dual-port memory is less than that required from a single-port memory. On the other hand, the chip area of a dual-port memory is more than that of a single-port memory by a factor of 1.5.

In general, for a memory with \( p \text{write} \) input ports and \( p \text{read} \) output ports, the required access time of this memory can be expressed as follows:

\[
\text{Memory Access Time} = \frac{W}{\text{Maximum}((N^p \text{write}) \cdot (M^p \text{read})) \cdot F}.
\]

Thus, the gain with respect to the memory access time when using a memory with more input and output ports should be evaluated against the chip area required for the VLSI implementation.

### 2.3 Address Queue Size

The addresses of the SBM in a shared-buffering ATM switch are managed by a memory management controller (MMC), as shown in Figure 2. The MMC manages the SBM by handling all the read and write operations to and from the SBM.

Inside the MMC, there is an address queue that plays an important role to support multicasting. In the literature, two architectures have been proposed for multicast address queuing: link list–based architectures and content address memory architectures [2, 3].

#### 2.3.1 Link List–Based Architectures

In this technique, the cells destined for each output port are chained using this mechanism. For each output port \( O_i \), cell \( C[O_i][J] \) (destined to \( O_i \) output port) is stored in the SBM at address \( A[O_i][J] \), along with the address of the next cell destined for the same output port \( O_i \). This next address is also stored in register \( W[A[O_i]] \) for the most recently written cell destined to output port \( O_i \).

For an \( N \times N \) switch with \( M \) cell buffer capacity installed in the switch, the total amount of memory needed in this technique is as follows:

- Shared buffer memory (SBM for cell storage) = \( 424 \cdot M \) bits (because the ATM cell is 424 bits long).
- Memory used by the link list = \( M \cdot \log(M) \).
- Memory used by the write and read address registers = \( 2 \cdot N \cdot \log(M) \).
- Memory used by the idle address buffer FIFO = \( M \cdot \log(M) \).
- Total used memory = \( 424 \cdot M + (2 \cdot M + 2 \cdot N)\log(M) \).

In this technique, many pointers have to be initialized and maintained along with registers for the write and read addresses for each output port. Also, using this technique complicates the addition of extra features such as handling priority and multicasting. The amount of memory used in this technique will increase as the number of input and output ports of the switch increases.

#### 2.3.2 Content Access Memory (CAM)–Based Architectures

In this technique, the SBM is implemented using a content address memory/random access memory (CAM/RAM) structure instead of the RAM structure used in the link list technique discussed above. The RAM stores the cell, and the CAM stores the tag associated with the cell. A cell is uniquely identified by a tag that constitutes an output port
Figure 1. Shared buffer memory

Figure 2. Memory management controller

number and a sequence number. Incoming cells are written into the first free location in the RAM, and the CAM is used to store the associated tag. Outgoing cells are accessed for reads by searching the CAM for the desired tag and then accessing the associated RAM location. Figure 3 shows the general architecture of a switch that is using this technique.

For an $N \times N$ switch with $M$ cell buffer capacity installed in the switch and $S$ bit sequence numbers, the total amount of memory needed in this technique is as follows:

- Shared buffer memory (SBM for cell storage) = $424 \cdot M$ bits (because the ATM cell is 424 bits long).
- CAM size = $M \cdot (N + S)$.
- Read and write sequence number registers = $N \cdot 2 \cdot S$.
- CAM valid bit = $N$ (to indicate empty buffer words).
- Total used memory = $(424 + N + S) \cdot M + (1 + 2 \cdot S) \cdot N$.

In this technique, the link list is indirectly replaced with a CAM tag, and the read and write pointers and registers are replaced with sequence numbers. Also, using this technique complicates the addition of extra features such as handling priority and multicasting. The amount of memory used in this technique will increase as the number of input and output ports of the switch increases.

2.4 Reliability of Large-Scale Switches

Large-scale ATM switches are constructed from smaller switches by interconnecting these small switches using multistage interconnection networks (MINs). As the
MIN size grows, reliability becomes crucial. Any single-component failure may crash the network if there is no redundancy. Much of the fault-tolerance work reported in the multistage interconnection networks literature is at the fabric topology level, concentrating on the use of multiple paths to route failures. System reliability is increased by offering multiple alternative paths between each input and output. The system could be overdimensioned initially, so that failures will not affect the quality of service, or it could be gracefully degrading. There are different possible techniques to improve the fault tolerance of MINs. In general, fault tolerance of MINs can be achieved by space redundancy or time redundancy. When space redundancy is used, multiple paths are created between inputs and outputs using redundant hardware that can be achieved by adding an extra stage of switches, varying the switch size, adding extra links, and adding extra ports and fault-tolerant switching elements. These techniques preserve the full-access property for the MINs in the presence of some faults. When time redundancy is used, multiple passes are made through the MIN to reach the desired destination. This technique is called dynamic full access (DFA).

Almost all of the implementations of fault-tolerant MINs introduce redundancy in the network. Different types of redundancy can be used as explained previously. These solutions are expensive in terms of number of extra switches per stage and/or the size of the switching elements. Moreover, these solutions have a high hardware complexity that needs complex routing algorithms.

3. Buffering Techniques

Mainly three different buffering strategies are available in ATM switch architectures. The physical location of the buffers in the switch determines the type of buffering technique used. In the following subsections, we present and evaluate the performance of these three different techniques.

3.1 Input Buffering

In input buffering, the contention problem is solved at the input buffer of the inlet of the switching element. Each inlet contains a dedicated buffer that is used to store the incoming cells until the arbitration logic decides to serve the buffer. The switching transfer medium then switches the ATM cells from the input queues to the outlet avoiding an internal contention. The arbitration logic can be as simple as a round-robin process or can be complex, taking into account the input buffer filling levels. However, this scheme has a head-of-line (HOL) blocking problem. This means that if two cells of two different inlets contend for the same output, one of the cells is to be stopped, and this cell blocks the other cells in the same inlet that are destined to different outlets. This queuing discipline is shown in Figure 4.

In an input-buffered switch, the bandwidth between each input port and the switch fabric needs to be only slightly greater than that of the port itself. The same holds for the bandwidth between the switch fabric and the output ports. This permits the input queues to be located separately from the switch fabric. This simplifies the implementation of the switch fabric and avoids the need for buffers operating at some multiple of the port speed.

A large nonblocking switch with first-in, first-out (FIFO) input buffers, saturated with uniform random traffic, has a throughput of about 58% compared to that of the ideal output-buffered switch. The performance of the

Figure 3. Content access memory (CAM)-based architecture
input-buffered delta or Banyan network is degraded even further because of the internal blocking within the network. The performance may be improved by a technique known as input queue bypass if access is permitted to other cells in the input queues besides the cell at the head of the queue. This technique, however, complicates the implementation of the input buffers and requires a more complex contention resolution scheme [4].

3.2 Output Buffering

In this queuing discipline, queues are located at each outlet of the switching element, and the output contention problem is solved using these queues. The cells arriving simultaneously at all inlets destined for the same output are queued in the buffer of the outlet. The only restriction is that the system must be able to write \( N \) cells in the queues during one cell time to avoid the cell loss where \( N \) is the total number of inlets of the switch. In this mechanism, no arbitration logic is required as all the cells can be switched to their respective output queue. The cells in the output queue are served using the FIFO discipline to maintain the integrity of the cell sequence. Figure 5 illustrates this mechanism.

In an ideal output-buffered switch, every output port must be able to accept a cell from every input port simultaneously (or at least within a single time slot). Using anything larger than a small switch module, it is unreasonable to expect the switch fabric and output buffers to have sufficient capacity to achieve ideal output-buffered operation. Thus, in an output-buffered switch of reasonable size, there is always the possibility that more cells will request access to a particular output port than the switch fabric output buffer can support. In this case, the excess cells must be discarded.

It is the switch designer’s task to ensure that the cell loss probability is sufficiently low for all reasonable patterns of input traffic and acceptable operating loads. An output-buffered switch can be more complex than an input-buffered switch because the switch fabric and output buffers must effectively operate at a much higher speed than that of each port to reduce the probability of cell loss.

3.3 Performance Analysis of Output Buffering

Let us consider an \( n \times n \) ATM switch. The load at each input port is \( \rho \), and the cell arrivals at the input ports follow the Bernoulli process. Destinations of a cell are uniformly distributed. Furthermore, we will assume that there is no correlation between the input links. However, to design an ATM switch to accept up to \( N \) cells per time slot at each output port is very complex. Therefore, an upper bound \( L \) is placed such that up to \( L \) simultaneous arriving cells at an output port during a time slot are accepted, and any excess cells are dropped, resulting in cell loss.

Each output port can send only one cell per time slot. Therefore, buffers are provided at each output port to queue the excess cells in the case of multiple cell arrivals for future transmission in future slots. If the buffer capacity per port is \( B \) cells, cell loss occurs when the buffer overflows. Therefore, one of the ATM switch design problems is to determine the buffer size \( B \) so that the cell loss probability will not exceed a predetermined value.

Figure 6 plots the simulation of the relationship between the cell loss probability and \( L \) (number of input ports of the memory) with \( \rho \) (input loading) as a parameter for a \( 16 \times 16 \) switch. The figure shows that for a specific traffic load, the cell loss probability decreases as \( L \) increases. It also shows that the cell loss probability increases as the traffic load increases for the same value of \( L \).

Figure 7 plots the relationship between the cell loss probability and \( \rho \) for a specific \( L \) and \( N \) as a parameter. The figure shows that for a specific switch size, the cell loss probability increases as the traffic load increases. It also shows that the cell loss probability increases as the switch size increases for the same traffic load.

Figure 8 plots the relationship between the cell loss probability and the required buffer size for \( L = 10 \), \( N = 16 \), and \( \rho \) as a parameter. The figure shows that the cell loss probability decreases as the buffer size increases for a specific traffic load. It also shows that the cell loss probability increases as the traffic load increases using the same buffer size. Figure 9 plots the same relationship for a switch size of \( 64 \times 64 \).

Figure 10 plots the relationship between the cell loss probability and the required buffer size for \( L = 64 \), \( N = 64 \), and \( \rho \) as a parameter. Compared to Figure 9, this figure shows that by increasing \( L \), the cell loss probability can be decreased using the same buffer size.

Figure 11 is derived from the analytical study using Maple for output buffering. The figure plots the relationship between the cell loss probability and \( L \), with the traffic load \( \rho \) as parameter. This figure shows that the cell loss...
probability decreases as $L$ increases for a specific traffic load. It also shows that the cell loss probability increases as the traffic load increases for the same $L$. This analytical study verifies the correctness of our simulation results.

### 3.4 Shared Buffering

In this scheme, the queuing buffers are shared between all inlets and outlets. All incoming cells are stored in the shared queue, and each outlet chooses the cells that are destined for it from the shared memory. Since cells for different outlets are merged in this shared queue, the FIFO discipline is not followed in the reading and writing of this queue. Cells can be written and read at random memory locations, and this needs a complex memory management system. Figure 12 shows this mechanism.

### 3.5 Performance Analysis of Shared Buffering

The buffer requirements within a switch can be reduced using a buffer that is shared between all the input ports and all the output ports rather than individual buffers at the

---

**Figure 6.** Log(cell loss probability) versus $L$ for $N = 16$ (output buffering)

**Figure 7.** Cell loss probability versus $\rho$ for $L = 4$ (output buffering)
switch output. The sharing performs a smoothing effect since the individual queues at the output ports are different at any instant.

Figure 13 plots our simulation results of the relationship between the cell loss probability and the buffer size, with $L = 10$ and the input loading $\rho$ as a parameter. The figure shows that the cell loss probability decreases as the buffer size increases for the same input load. It also shows that the cell loss probability increases as the input load increases using the same buffer size. Figure 14 plots the same relationship for a $64 \times 64$ switch size.

Figures 15 and 16 plot the analytical results of the relationship between the cell loss probability and the buffer size for $4 \times 4$ and $64 \times 64$ switch sizes, respectively. These figures verify the correctness of our simulation results.

4. Combined Central and Output Queuing (CCOQ)

For large-scale ATM switches (in terms of input and output ports), the amount of memory that needs to be installed in the switch is large, so reducing the size of the memory is of great importance.
Figure 10. Cell loss probability versus buffer size for $N = 64$ and $L = 64$ (output buffering)

Figure 11. Analytical results of log(cell loss probability) versus $L$ for $N = 64$ (output buffering)

Figure 12. Shared buffering
In many cases, the amount of memory that needs to be installed in a large-scale ATM switch is large, and it cannot be integrated on a single chip either because of its size or because of the lower operating speed that it will have. In such cases, and on the basis of our simulation studies, we propose implementing the large-scale ATM switch out of smaller switches where a small amount of memory needs to be installed in each of these smaller switches. This technique will slightly increase the total amount of memory that will be used to implement the large-scale ATM switch. But a number of other advantages of this approach can be listed as follows:

- The amount of memory that needs to be integrated on a single switch is small.
- The operating speed of these smaller memories installed in the small switches that make up the large-scale ATM switch is more than that of one large memory.
- Using this approach, the design of the switch will be more...
modular. The number of output ports of the switch can be increased at any time by adding smaller switches inside the large-scale ATM switch.

5. Performance of CCOQ

Figure 17 shows different buffering configurations that can be used within a $16 \times 16$ ATM switch. These configurations include output buffering, shared buffering, and two other CCOQ configurations in which all the input ports and part of the output ports have access to the same memory module. Table 1 shows the number of memories, size of memory integrated on a single chip, and the total memory needed in all of these configurations. The table shows that in the case of shared buffering, the amount of memory that needs to be integrated on a single chip is $600 \times 424$ bits.
Table 1. Comparison of buffering techniques for 16 × 16 asynchronous transfer mode (ATM) switch

<table>
<thead>
<tr>
<th>Number of Memories Used</th>
<th>Size of Memory Integrated on a Single Chip</th>
<th>Total Memory Size Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (shared buffering)</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>2 (CCOQ M = 2)</td>
<td>450</td>
<td>900</td>
</tr>
<tr>
<td>4 (CCOQ M = 4)</td>
<td>250</td>
<td>1000</td>
</tr>
<tr>
<td>16 (output buffering)</td>
<td>150</td>
<td>2400</td>
</tr>
</tbody>
</table>

CCOQ = combined central and output queuing.

In the case of output buffering, the table shows that the amount of memory that needs to be integrated on a single chip is 150 · 424 bits, but 16 of these memories are needed.

Studying Table 1, we conclude that the two CCOQ configurations (with \( M = 2 \) and \( M = 4 \)) are better than the output-buffering configuration in terms of the total memory that needs to be installed in the switch. The table also shows that the two CCOQ configurations are better than the shared-buffering configuration in terms of the amount of memory that needs to be integrated on a single chip.

Figure 18 shows that when the number of memories used in the switch (16 × 16 in this example) increases from one to two and then to four, the amount of memory that needs to be integrated on a single chip decreases dramatically. When the number of memories used in the switch is more than four, we see only a slight decrease in the buffer size that needs to be integrated on a single chip. Thus, using four memory modules for a 16 × 16 switch to achieve \( 10^{-5} \) cell loss probability is a good selection in this example, assuming that the most critical design condition is the amount of memory to be integrated on a single chip and the speed of this memory.

6. Proposed Architecture

This section presents a new fault-tolerant multicast 4 × 4 ATM switch. The switch is designed based on queuing strategies, multicasting considerations, and error detection for fault-tolerance implementation. It is based on a novel idea on realizing multicasting. A small address memory will be used to hold a reference for the cell that needs to be switched to several outputs.

Figure 19 depicts the proposed 4 × 4 ATM switching architecture. Each of the two switching modules (SM) in the switch receives all the four inputs. The memory controller module (MCM) of the SM then determines where to store the incoming cells if they are intended for one of the two output ports controlled by that SM. Otherwise, it discards the cell.

---

© 2002 Simulation Councils Inc.. All rights reserved. Not for commercial use or unauthorized distribution.
Figure 18. Size of memory integrated on a single chip (in cells) versus number of memories used to implement a $16 \times 16$ asynchronous transfer mode (ATM) switch and achieve $10^{-5}$ cell loss probability.

Figure 19. Architecture of a $4 \times 4$ combined central and output queuing (CCOQ) switch.
In the following sections, we describe the memory module (MM) and the memory controller module (MCM) that make up each SM.

6.1 Memory Module (MM)

The MM of each SM shown in Figure 19 should be accessed six times (four times to write and two times to read) instead of eight times in the case of the shared-buffering architecture. If the memory unit available cannot be accessed six times per cell slot, the MM can be designed using more than one memory unit.

Figure 20 shows the SM used in the $4 \times 4$ switch shown in Figure 19. This SM is designed with three memory units; each memory should be accessed two times per cell slot (one memory unit can be accessed for read, and the other two units can be accessed for write).

Table 1 shows the simulation results of the buffer requirements to achieve a given cell loss probability for a $16 \times 16$ switch under 0.9 loading conditions for different memory locations: $M = 16$ is output buffering, and $M = 1$ is the shared buffer. For $M$ between 1 and 16, we have CCOQ, where $M$ represents the number of memory modules shared by the outputs.

These results show that the buffer requirements for our buffering scheme ($M = 4$) are more than that needed in the case of shared buffering and less than that needed in the case of output buffering to achieve a given cell loss probability.

Our motivation to use this buffering scheme, despite that it requires more memory, was to reduce the memory complexity by reducing the number of memory access needed. The other motivation is that normally, the memory that is needed for a large-scale switch is larger than the amount that can be integrated on a single chip. So, by adopting our scheme, the memory is already partitioned, and those partitions can be designed in such a way to be integrated on a single chip.

6.2 Memory Control Module (MCM)

Our architecture uses two types of memory controller modules—mainly, MCM and MCM-F. In the following, we describe in general the main functionality of each of those two modules.

6.2.1 MCM

This module compares the two bits of the error control field of the ATM cell with the two most significant bits of the destination field. If the bits do not match, then the MCM will discard the cell, assuming that an error occurred in one switching element of a previous stage in the switching fabric. If the bits match, the MCM will write the incoming ATM cell to the location pointed to it by the write pointer. Figure 21 shows the traffic generated by a properly working switch [5].

6.2.2 MCM-F

This module will be activated in case one or more MCM modules in the switch are not working. The main functionality of this module is to broadcast the incoming cells to all possible outgoing links while attaching to them the appropriate error flag. Figure 21 shows the traffic generated by a faulty switch [5].

The MCM-F keeps track of two bitmaps per output port, as shown in Figure 22. When a cell arrives to the switch and it is destined to one or more output ports with a specific priority level, the MCM writes a “1” bit in the bitmap associated with the destination output port and the priority level needed.

During each cell period, the MCM checks the read pointers of bitmaps of all output ports and selects to route the cells based on the FIFO service scheme. The MCM also checks the bitmap of the higher priority level of each output port so that it services the higher priority cells first.
Guizani and Al-Fuqaha

Properly working switch

Faulty switch

Error field

Routing bits

101110

none

1110

none

Figure 21. Traffic generated by faulty and properly working switches

Figure 22. The memory control module description

scheme, the total memory required by the memory controller is as follows:

Total capacity = P \cdot \text{B}_s \cdot N \cdot (N/s),

where B_s is the required buffer size for s ports sharing the same memory.

7. Verilog Simulation

The Verilog hardware description language (HDL) was used to verify the functionality of the switch. Figure 23 shows some of the signals from the Verilog simulation program of the proposed architecture. These signals are taken from the main components of the switch—mainly, the memory units and the memory controller management (MCM). These signals show the correctness of the functionality of the proposed switch.

8. Reliability Analysis of Proposed Switch

Almost all of the implementations of fault-tolerant MINs introduce redundancy in the network. Different types of redundancy can be used as explained previously. These solutions are expensive in terms of number of extra switches per stage and/or the size of the switching elements. Moreover, these solutions have a high hardware complexity that needs complex routing algorithms. In our design, the reliability of the switch was increased by adding an extra unit (MCM-F) within the switch.

In our proposed switch design approach, a large-scale ATM switching fabric can be constructed by using multiple switching elements (e.g., the 4 \times 4 switching element described above). If the number of failures is k, where 0 \leq k \leq n^{2^{n-1}}, then the number of configurations in which these k failures can occur is given by

\binom{n \times 2^{n-1}}{k}.

The survival function Q(k) is given by [1]:

Q(k) = \frac{n \times 2^n}{k} \cdot \frac{(n-1)2^{n-1}}{k},

assuming that the failure rate of the memory units, MCM modules, and MCM-F is the same (equals \lambda). The reliability of the proposed 4 \times 4 switching element is given by the following (see Fig. 24):

R_{\text{switch}} = e^{-2\lambda r} \left[1 - (1 - e^{-\lambda r})(1 - e^{-2\lambda r})\right].

9. Conclusion

In this work, we studied the three different buffering configurations used in the literature. This study inspired us to propose a new buffering technique that is based on CCOQ. Using CCOQ, the total memory that needs to be installed in the switch to achieve a given cell loss probability is less than that required in the case of output buffering. Also, the total memory size will be more than that required in the case of shared queuing. But the memory speed that is required using CCOQ will be less than that required in the case of shared queuing. Analytical and simulation studies
were presented to study the performance of the proposed buffering technique.

This work also presented a new MCM to manage the memory modules installed in the switch. The proposed MCM has a small address memory that is used to hold the address of the cell that needs to be switched to its desired output(s). This leads to a better use of the available buffer space.

A $4 \times 4$ ATM switch was then proposed. The proposed switch uses the proposed CCOQ buffering technique. It also uses the proposed MCM to manage the memory modules installed in the switch. The proposed architecture was then simulated in Verilog HDL to show the correctness of the design. Finally, reliability analysis of the switch was performed to find the survivability of the switch.

10. Future Work

In the future, the CCOQ buffer requirements can be evaluated against the other buffering techniques under bursty
and self-similar traffic conditions. Also, the throughput of the proposed CCOQ can be evaluated against the other buffering techniques.

Finally, the proposed MCM can be used to manage the memory module that is installed in shared-buffering ATM switches. This will reduce the size of the address queues needed to manage the shared memory used in shared-buffering ATM switches.

11. References


Ala Isam Al-Fuqaha is a member of technical staff working for FirstWave Intelligent Optical networks on a new all optical cross-connect (OXC). Prior to joining FirstWave, Inc., he was working for the call processing and service delivery group of Sprint Telecom on projects that involved the Integrated on Demand Network (ION) and the long distance telephone network. He received his BSc degree in computer engineering from the University of Petroleum and Minerals, Saudi Arabia. He then worked as a research assistant at the University of Missouri-Columbia. His research interests include simulation, ATM, MPLS, all optical switching, and VLSI design automation. He received his MSc degree in Electrical and Computer Engineering from the University of Missouri-Columbia in 1999.