CHAPTER 9

KNOCKOUT-BASED SWITCHES

As shown in Chapter 5, output buffer switches (including the shared-memory switches) provide the best throughput/delay performance. The problem of the output buffered switches is that their capacity is limited by the memory speed. Consider the case of an ATM switch with 100 ports: What is the probability of all 100 cells arriving at the same output port at the same time slot? If the probability is very low, why do we need to have the output buffer to receive all 100 cells at the same slot? A group of researchers at Bell Labs in the late 1980s tried to resolve this problem by limiting the number of cells that can arrive at an output port in each time slot and thus the speed requirement of the memory at the output ports is no longer a constraint to the switch system. Excessive cells are discarded by the switch fabric. The concept is called the 'knockout principle'. The question is how many cells should be delivered to the output port in each time slot. If too many, memory speed may be a bottleneck. If too few, the cell loss rate in the switch fabric may be too high to be acceptable. For a given cell loss rate, this number can be determined. The number is found to be 12 for a cell loss rate of 10^{-10}, independent of the switch size.

The result seems very encouraging in a way that the memory speed is no longer a bottleneck for the output buffered switch. However, there are no commercial switches implemented with the knockout principle. This is because the results that are obtained assume that input traffic distribution from different inputs are uncorrelated, which may be unrealistic in the real world. In addition, people are not comfortable with the fact that cells are discarded by the switch fabric. Usually, cells are discarded when the buffer is filled or exceeds some predetermined thresholds.

Although the knockout principle has not been used in real switching systems, its concept has attracted many researchers in the past and various architectures based on this concept have been proposed. Some of them are discussed in this chapter. Section 9.1 describes the knockout principle and an implementation and architecture of a knockout switch. Section 9.2

High Performance Switches and Routers, by H. Jonathan Chao and Bin Liu

describes a useful and powerful concept, channel grouping, to save the routing links in the switch fabric. A generalized knockout principle that extends the original knockout principle by integrating the channel grouping concept is described. Section 9.3 describes a two-stage multicast output-buffered switch that is based on the generalized knockout principle. Section 9.4 is an appendix that shows the derivation of some equations used in this chapter.

9.1 SINGLE-STAGE KNOCKOUT SWITCH

9.1.1 Basic Architecture

The knockout switch [1] is illustrated as in Figure 9.1. It is composed of a completely broadcasting interconnection fabric and N buses interfaces. The interconnection fabric for the knockout switch has two basic characteristics: (1) each input has a separate broadcast bus, and (2) each output has access to all broadcast buses and thus all input cells.

With each input having a direct path to every output, no switch blocking occurs within the interconnection fabric. The only congestion in the switch takes place at the interface to each output where cells can arrive simultaneously on different inputs destined for the same output. The switch architecture is modular in a way that the N broadcast buses can reside on an equipment backplane with the circuitry for each of the N input/output pairs placed on a single plug-in circuit card.

Figure 9.2 illustrates the architecture of the bus interface associated with each output of the switch. The bus interface has three major components. At the top there are a row of N cell filters where the address of every cell is examined, with cells addressed to the output allowed to pass on to the concentrator and all others blocked. The concentrator then achieves an N to L ($L \ll N$) concentration of the input lines, and up to L cells in each time slot will emerge at the outputs of the concentrator. These L concentrator outputs then enter a shared buffer that is composed of a barrel shifter and L separate FIFO buffers. The shared buffer allows complete sharing of the L FIFO buffers and provides the equivalent of a single queue with L inputs and one output, each operated under a FIFO queuing discipline. The operation of the barrel shifter is shown in Figure 9.3. At time T, cells A, B, C arrive and are stored in the top three FIFO buffers. At time $(T + 1)$, cells D to J arrive and begin to

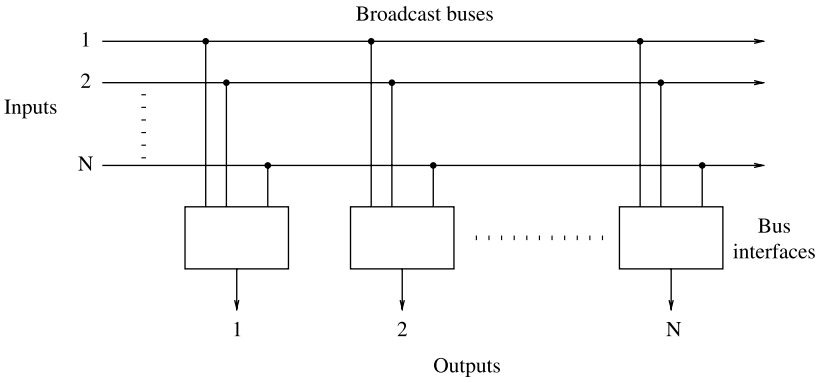


Figure 9.1 Knockout switch interconnection fabric.

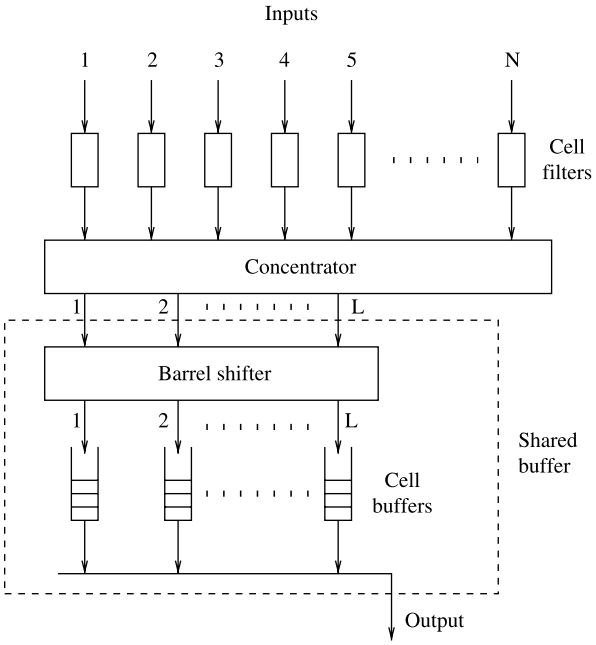


Figure 9.2 Knockout switch bus interface (©1987 IEEE).

be stored in the 4th FIFO in a round robin manner. The number of positions that the barrel shifter shifts is equal to the sum of the arriving cells mod L.

9.1.2 Knockout Concentration Principle

All cells passing through the cell filters enter the concentrator, with an N to L concentration. If there are $k \leq L$ cells arriving in a time slot for a given output, these k cells will emerge from the concentrator on outputs 1 to k after leaving the concentrator. If $k > L$, then all L outputs of the concentrator will have cells and $k - L$ cells will be dropped (i.e., lost) within the concentrator.

The cell loss probability is evaluated as follows. It is assumed that, in every time slot, there is a fixed and independent probability ρ that a cell arrives at an input. Every cell is equally likely to be destined for each output. Denote P_k as the probability of k cells arriving in a time slot all destined for the same output, which is binomially distributed as follows:

$$P_k = \binom{N}{k} \left(\frac{\rho}{N}\right)^k \left(1 - \frac{\rho}{N}\right)^{N-k} \qquad k = 0, 1, \ldots, N. \tag{9.1}$$

It then follows that the probability of a cell being dropped in a concentrator with N inputs and L outputs is given by

$$\Pr[\text{cell loss}] = \frac{1}{\rho} \sum_{k=L+1}^{N} (k - L) P_k = \frac{1}{\rho} \sum_{k=L+1}^{N} (k - L) \binom{N}{k} \cdot \left(\frac{\rho}{N}\right)^k \left(1 - \frac{\rho}{N}\right)^{N-k}. \tag{9.2}$$

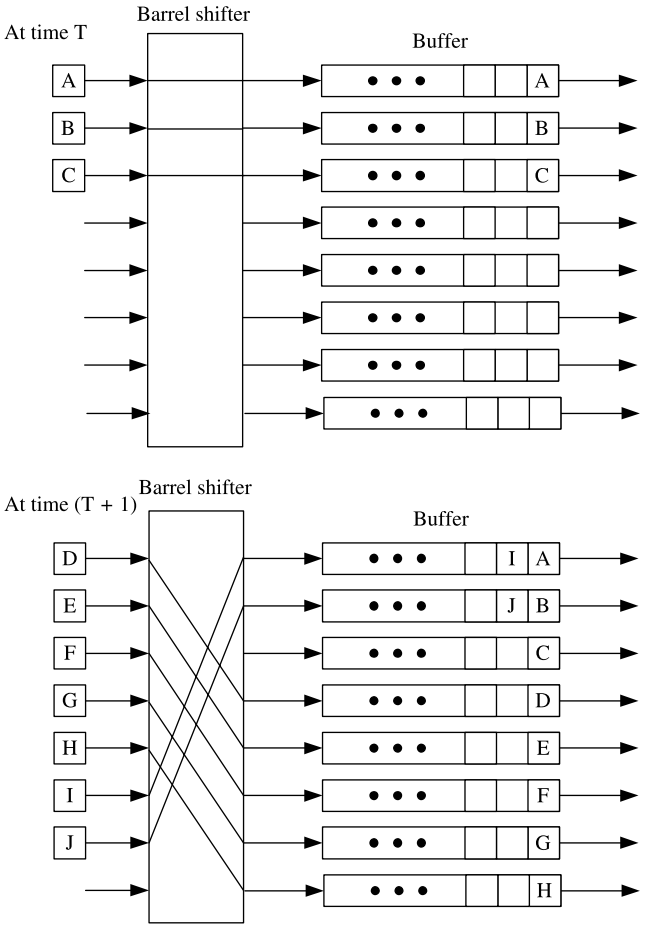
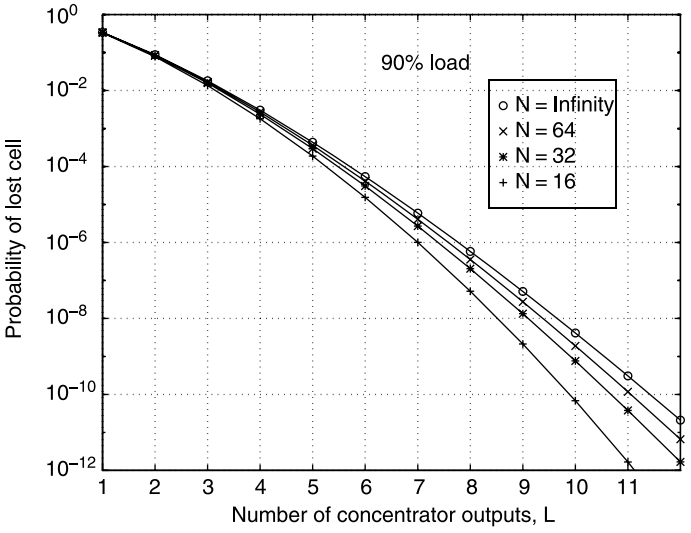


Figure 9.3 Operation of a barrel shifter.

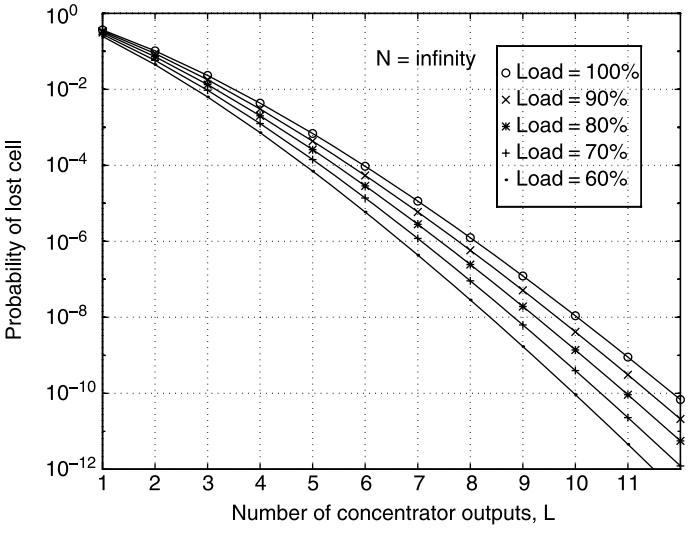
Taking the limit as $N \to \infty$, and with some manipulations,

$$\Pr[\text{cell loss}] = \left[1 - \frac{L}{\rho}\right]\left[1 - \sum_{k=0}^{L} \frac{\rho^k e^{-\rho}}{k!}\right] + \frac{\rho^L e^{-\rho}}{L!} \tag{9.3}$$

Using (9.2) and (9.3), Figure 9.4*a* shows a plot of the cell loss probability versus L, the number of outputs on the concentrator, for $\rho = 0.9$ and $N = 16, 32, 64, \infty$. Note that a concentrator with only eight outputs achieves a cell loss probability less than 10^{-6} for an arbitrately large N. This is comparable to the probability of losing a 500-bit cell from transmission errors with a bit error rate of 10^{-9}. Also note from Figure 9.4*a* that each additional output added to the concentrator beyond eight results in an order of magnitude decrease in the cell loss probability. Hence, independent of the number of inputs N, a concentrator with 12 outputs will have a cell loss probability $<10^{-10}$. Figure 9.4*b* illustrates, for $N \to \infty$, that the required number of concentrator outputs is not particularly sensitive to



(*a*)



(*b*)

Figure 9.4 Concentrator cell loss performance with (*a*) various switch sizes and (*b*) various loads.

the load on the switch, up to and including a load of 100 percent. It is also important to note that, assuming independent cell arrivals on each input, the simple, homogeneous model used in the analysis corresponds to the worst case, making the cell loss probability performance results shown in Figure 9.4 upper bounds on any set of heterogeneous arrival statistics [2].

9.1.3 Construction of the Concentrator

The basic building block of the concentrator is a simple 2×2 contention switch shown in Figure 9.5*a*. The two inputs contend for the 'winner' output according to their activity bits.

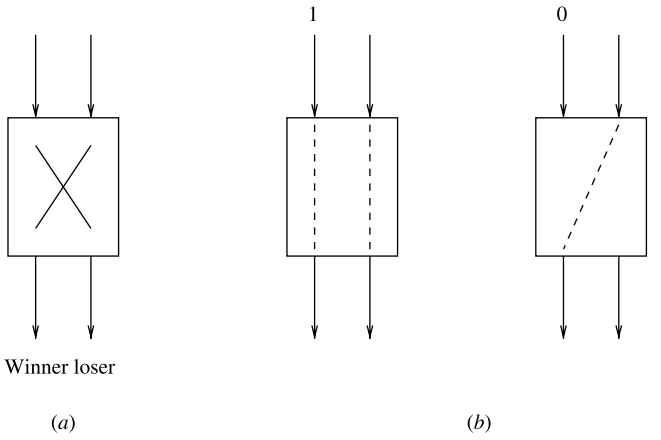


Figure 9.5 (*a*) 2 × 2 contention switch; (*b*) States of 2 × 2 contention switch.

If only one input has an arriving cell (indicated by an activity bit = 1), it is routed to the winner (left) output. If both inputs have arriving cells, one input is routed to the winner output and the other input is routed to the loser output. If both inputs have no arriving cells, we do not care except that the activity bit for both should remain at logic 0 at the switch outputs.

The above requirements are met by a switch with the two states shown in Figure 9.5*b*. The switch examines the activity bit of the left input only. If the activity bit is a '1', the left input is routed to the winner output and the right input is routed to the loser output. If the activity bit is a '0', the right input is routed to the winner output, and no path is provided through the switch for the left input. Such a switch can be realized with as few as 16 gates, and having a latency of at most one bit. Note that priority is given to the cell on the left input to the 2 × 2 switch element. To avoid this, the switch element can be designed so that it alternates between selecting the left and right inputs as winners when there is a cell arriving on both inputs in the same time slot. However, suppose the priority structure of the 2 × 2 switch element were maintained and (as described below) the concentrator were designed so that one input, say the Nth, always received the lowest priority for exiting a concentrator output. The cell loss probability for this worst case input, as $N \to \infty$, is given by

$$\Pr[\text{cell loss for the worst case input}] = 1 - \sum_{k=0}^{L-1} \frac{\rho^k e^{-\rho}}{k!}. \tag{9.4}$$

The above equation is obtained by considering there are k cells destined for the same output port from the first $N - 1$ inputs, where

$$P_k = \binom{N-1}{k} \left(\frac{\rho}{N-1}\right)^k \left(1 - \frac{\rho}{N-1}\right)^{N-1-k} \qquad k = 0, 1, \ldots, N-1. \tag{9.5}$$

As $N \to \infty$, $P_k = \rho^k e^{-\rho}/k!$. Cells at the Nth input will be transmitted to the output if the number of cells from the first $(N - 1)$ inputs destined for the same output port are less than or equal to $(L - 1)$. The entire summation in (9.4) is the probability that the cell from the Nth input will not be lost. Comparing the results of (9.4) to the cell loss probability averaged

over all inputs, as given by (9.3) and shown in Figure 9.4*b*, it is found that the worst case cell loss probability is about a factor of 10 greater than the average. This greater cell loss probability, however, can be easily compensated for by adding an additional output to the concentrator.

Figure 9.6 shows the design of an 8-input 4-output concentrator composed of these simple 2×2 switch elements and single-input/single-output 1-bit delay elements (marked D). At the input to the concentrator (upper left side of Fig. 9.6), the N outputs from the cell filters are paired and enter a row of $N/2$ switch elements. One may view this first stage of switching as the first round of a tournament with N players, where the winner of each match emerges from the left side of the 2×2 switch element and the loser emerges from the right side. The $N/2$ winners from the first round advance to the second round where they compete in pairs as before using a row of $N/4$ switch elements. The winners in the second round advance to the third round, and this continues until two compete for the championship: that is, for the right to exit as the first output of the concentrator. Note that if there is at least one cell arriving on an input to the concentrator, a cell will exit the first output of the concentrator.

A tournament with only a single tree-structured competition leading to a single winner is sometimes referred to as a single knockout tournament: lose one match and you are knocked out of the tournament. In a double knockout tournament, the $N - 1$ losers from the first section of competition compete in a second section, which produces a second place finisher (i.e., a second output for the concentrator) and $N - 2$ losers. As Figure 9.6 illustrates, the losers from the first section can begin competing in the second section before

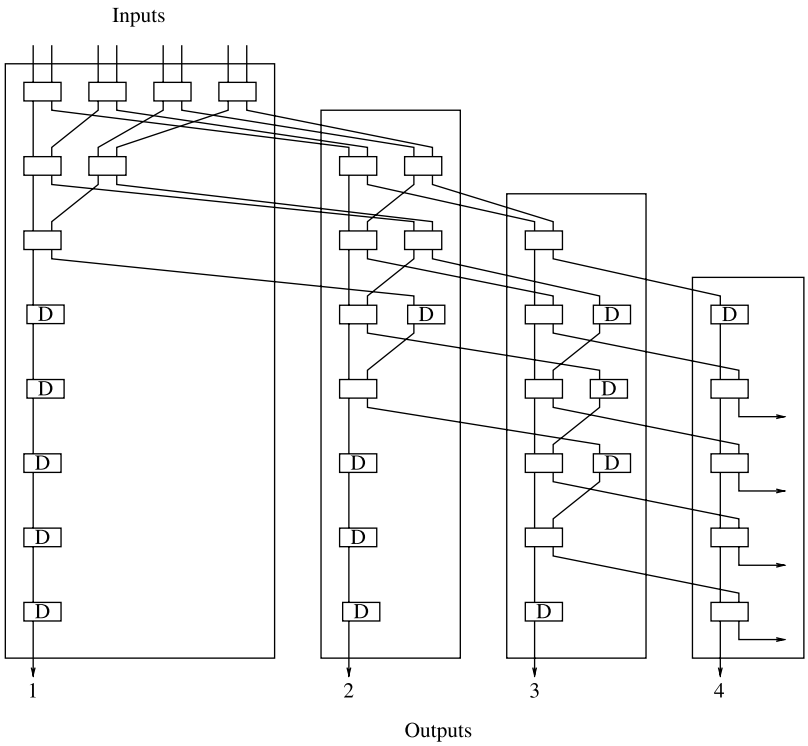


Figure 9.6 8-input to 4-output concentrator (©1987 IEEE).

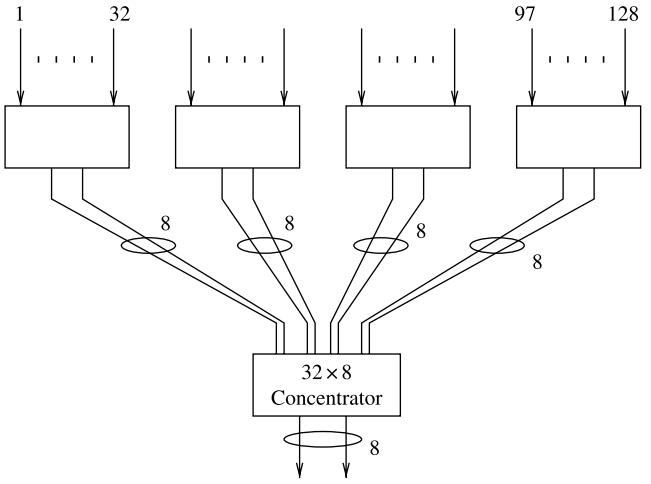


Figure 9.7 128-to-8 concentrator constructed from 32-to-8 concentrator chips.

the competition is finished in the first. Whenever there are an odd number of players in a round, one player must wait and compete in a later round in the section. In the concentrator, a simple delay element serves this function.

For a concentrator with N inputs and L outputs, there are L sections of competition, one for each output. A cell entering the concentrator is given L opportunities to exit through a concentrator output. In other words, a cell losing L times is knocked out of the competition and is discarded by the concentrator. In all cases, however, some cells are lost only if more than L cells arrive in any one time slot. As we have seen, for $L \geq 8$, this is a low probability event.

For $N \gg L$, each section of the concentrator contains approximately N switch elements for a total concentrator complexity of $16NL$ gates. For $N = 32$ and $L = 8$, this corresponds to a relatively modest 4000 gates. Once a concentrator micro-circuit is fabricated, Figure 9.7 illustrates how several identical chips can be interconnected to form a larger concentrator. The loss probability performance of the two-stage concentrator is the same as the single-stage concentrator. In general, a $K^J L$ input, L output concentrator can be formed by interconnecting J rows of KL-to-L concentrator chips in a tree-like structure, with the ith row (counting from the bottom) containing K^{i-1} chips. For the example illustrated in Figure 9.7, $L = 8$, $K = 4$, and $J = 2$.

9.2 CHANNEL GROUPING PRINCIPLE

The construction of a two-stage modular network is mostly based on the channel grouping principle [3] to separate the second stage from the first stage. With a group of outputs treated identically in the first stage, a cell destined for an output of this group can be routed to any output of the group before being forwarded to the desired output in the second stage. For instance, as shown in Figure 9.8, the cell at the top input destined for output 6 appears at the second input of the second group, while another input cell destined for output 0 appears at the first input of the first group. The first stage network routes cells to

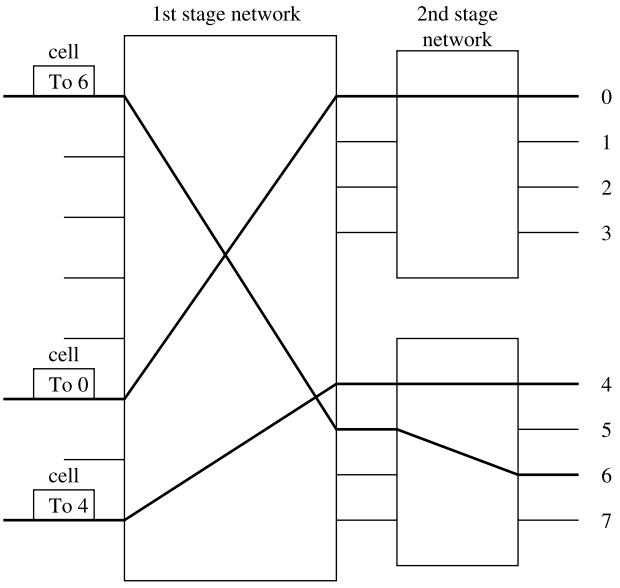


Figure 9.8 Illustration of channel grouping principle.

their proper output groups and the second stage network further routes cells to their proper output ports. This smoothes the problem of output contentions and thus achieves better performance/complexity tradeoff for the first-stage switch. More theoretic evaluations are provided as follows.

9.2.1 Maximum Throughput

This section focuses on the switch structure shown in Figure 9.9. An output group consists of M output ports and corresponds to an output address for the 1st-stage network. A cell can access any of the M corresponding output ports of the 1st-stage network. In any given time slot, at most M cells can be cleared from a particular output group, one cell on each output port.

The maximum throughput of an input-buffered switch is limited by head-of-line blocking. The symmetric case (i.e., same number of input and output ports) is evaluated in [4] and the maximum throughput is 0.586. A similar approach could be taken for the asymmetric case to a point where the solution could be found by numerical analysis [5].

Table 9.1 lists the maximum throughput per input for various values of M and K/N [5]. The column in which $K/N = 1$ corresponds to special cases studied by Hluchyj and Karol [4] and Oie et al. [6]. For a given M, the maximum throughput increases with K/N because the load on each output group decreases with K/N. For a given K/N, the maximum throughput increases with M because each output group has more output ports for clearing cells.

Table 9.2 lists the maximum throughput as a function of the line expansion ratio (the ratio of the number of output ports to the number of input ports), $(K \times M)/N$. Notice that for a given line expansion ratio, the maximum throughput increases with M. Channel grouping has a stronger effect on throughput for smaller $(K \times M)/N$ than for larger $(K \times M)/N$.

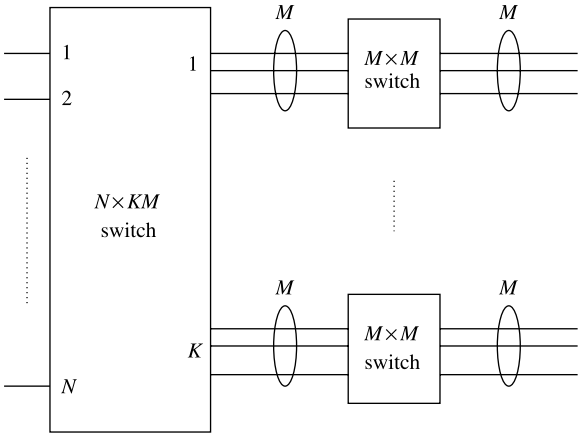


Figure 9.9 Asymmetric switch with line expansion ratio, $(K \times M)/N$.

This is because for large $(K \times M)/N$, $M = 1$, the line expansion has already alleviated much of the throughput limitation due to head-of-line blocking.

9.2.2 Generalized Knockout Principle

This section generalizes the knockout concentration loss calculation to a group of outputs [7, 8]. Consider an $N \times N$ switch with two stages routing networks, as shown in Figure 9.10. A group of M outputs at the 2nd-stage share $L \times M$ routing links from the 1st-stage network. The probability that an input cell is destined for this group of outputs is simply $M\rho/N$. If only up to $L \times M$ cells are allowed to pass through to the group of outputs, where L is called the group expansion ratio, then

$$\Pr[\text{cell loss}] = \frac{1}{M\rho} \sum_{k=L\times M+1}^{N} (k - L \times M) \binom{N}{k} \left(\frac{M\rho}{N}\right)^k \cdot \left(1 - \frac{M\rho}{N}\right)^{N-k}. \qquad (9.6)$$

TABLE 9.1 Maximum Throughput With K/N kept Constant While $K, N \to \infty$

	K/N								
M	1/16	1/8	1/4	1/2	1	2	4	8	16
1	0.061	0.117	0.219	0.382	0.586	0.764	0.877	0.938	0.969
2	0.121	0.233	0.426	0.686	0.885	0.966	0.991	0.998	0.999
4	0.241	0.457	0.768	0.959	0.996	1.000	1.000	1.000	
8	0.476	0.831	0.991	1.000	1.000				
16	0.878	0.999	1.000						

TABLE 9.2 Maximum Throughput With $(K \times M)/N$ kept Constant While $(K \times M), N \to \infty$

	$(K \times M)/N$					
M	1	2	4	8	16	32
1	0.586	0.764	0.877	0.938	0.969	0.984
2	0.686	0.885	0.966	0.991	0.998	0.999
4	0.768	0.959	0.996	1.000	1.000	1.000
8	0.831	0.991	1.000			
16	0.878	0.999				
32	0.912	1.000				
64	0.937					
128	0.955					
256	0.968					
512	0.978					
1024	0.984					

As $N \to \infty$,

$$\Pr[\text{cell loss}] = \left[1 - \frac{L}{\rho}\right]\left[1 - \sum_{k=0}^{L\times M} \frac{(M\rho)^k e^{-M\rho}}{k!}\right] + \frac{(M\rho)^{L\times M} e^{-M\rho}}{(L \times M)!}. \qquad (9.7)$$

The derivation of the above equation can be found in the appendix of this chapter. As an example, we set $M = 16$ and plot in Figure 9.11 the cell loss probability (9.7) as a function of $L \times M$ under various loads. Note that $L \times M = 33$ is large enough to keep the cell loss probability below 10^{-6} for a 90 percent load. In contrast, if the group outputs had been treated individually, the value of $L \times M$ would have been 128 (8×16) for the same cell loss performance. The advantage from grouping outputs is shown in Figure 9.12 as the group expansion ratio L versus a practical range of M under different cell loss criteria. For a cell loss probability of 10^{-8}, note that L decreases rapidly from 8 down to less than 2.5 from group sizes M larger than 16; a similar trend is evident for other cell loss probabilities.

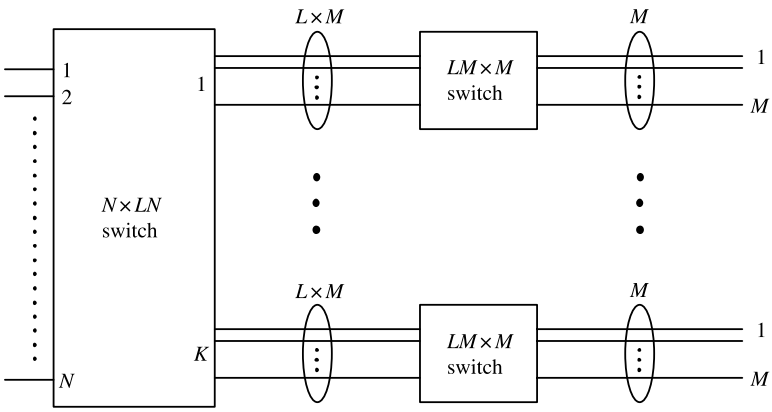


Figure 9.10 $N \times N$ switch with group expansion ratio, L.

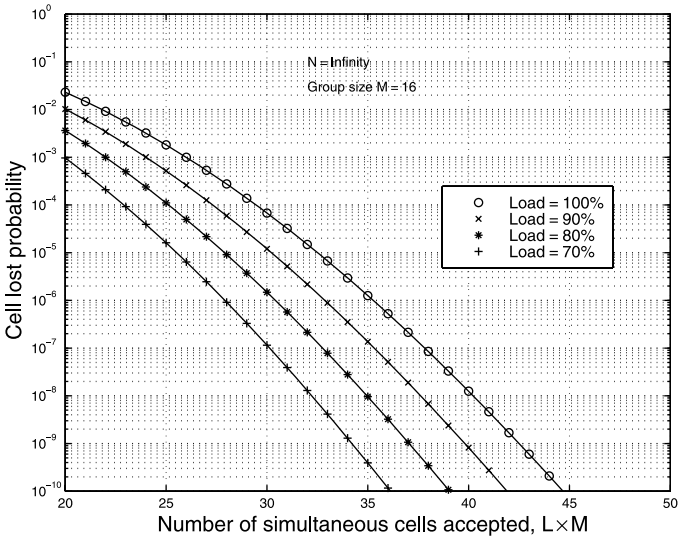


Figure 9.11 Cell loss probability when using the generalized knockout principle.

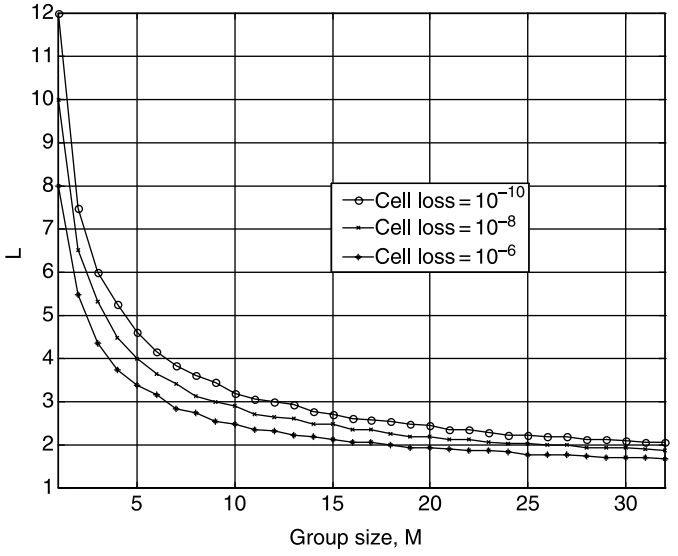


Figure 9.12 Ratio of the number of simultaneous cells accepted to group size for various cell loss probabilities.

9.3 TWO-STAGE MULTICAST OUTPUT-BUFFERED ATM SWITCH (MOBAS)

9.3.1 Two-Stage Configuration

Figure 9.13 shows a two-stage structure of the multicast output buffered ATM switch (MOBAS) that adopts the generalized knockout principle described above. As a result, the

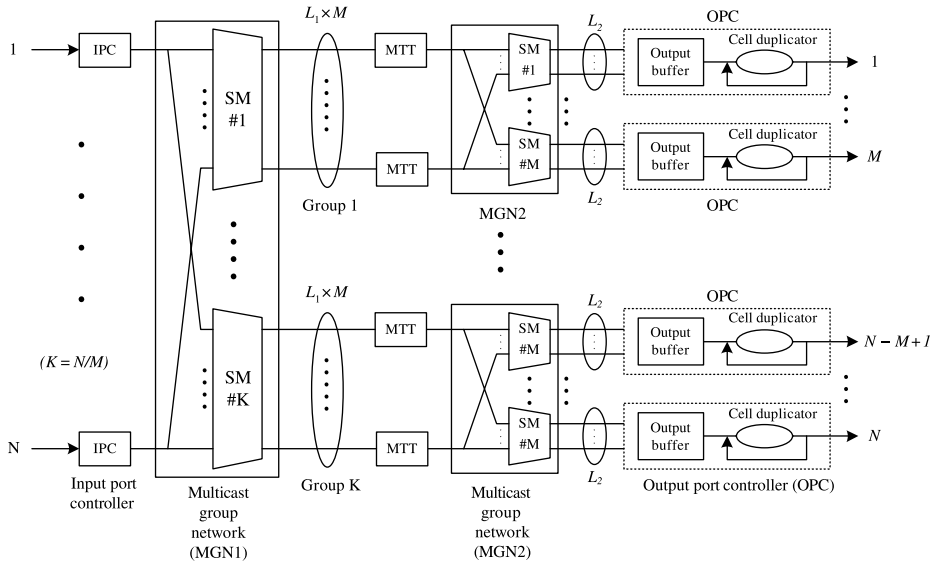


Figure 9.13 Architecture of a multicast output buffered ATM switch (MOBAS) (©IEEE 1995).

complexity of interconnection wires and building elements can be reduced significantly, for example, by almost one order of magnitude [8]. The switch consists of input port controllers (IPCs), multicast grouping networks (MGN1, MGN2), multicast translation tables (MTTs), and output port controllers (OPCs). The IPCs terminate incoming cells, lookup necessary information in translation tables, and attach the information (e.g., multicast patterns and priority bits) to the front of the cells such that the cells can be properly routed in the MGNs. The MGNs replicate multicast cells based on their multicast patterns and send one copy to each output group. The MTTs facilitate the multicast cell routing in the MGN2. The OPCs temporarily store multiple arriving cells destined for that output port in an output buffer, generate multiple copies for multicast cells with a cell duplicator (CD), assign a new virtual channel identifier (VCI) obtained from a translation table to each copy, convert the internal cell format to the standardized ATM cell format, and finally send the cells to the next switching node or the final destination.

Let us first consider the unicast situation. As shown in Figure 9.13, all M output ports are bundled in a group, and there are a total of K groups ($K = N/M$) for a switch size of N inputs and N outputs. Due to cell contention, $L_1 \times M$ routing links are provided to each group of M output ports. If there are more than $L_1 \times M$ cells in one cell time slot destined for the same output group, the excess cells will be discarded and lost. However, we can engineer L_1 (called the group expansion ratio) such that the probability of cell loss due to the competition for the $L_1 \times M$ links is lower than that due to the buffer overflow at the output port or bit errors occurring in the cell header. Performance study in Section 9.2.2 shows that the larger M is, the smaller L_1 needs to be to achieve the same cell loss probability. For instance, for a group size of one output port, which is the case in the second stage (MGN2), L_2 needs to be at least 12 to have a cell loss probability of 10^{-10}. But for a group size of 32 output ports, which is the case in the first stage (MGN1), L_1 just needs to be 2 to have the same cell loss probability. Cells from input ports are properly routed in MGN1 to one of the K groups; they are then further routed to a proper output port through the MGN2. Up

to L_2 cells can arrive simultaneously at each output port. An output buffer is used to store these cells and send out one cell at each cell time slot. Cells that originate from the same traffic source can be arbitrarily routed onto any of the $L_1 \times M$ routing links and their cell sequence is still retained.

Now let us consider a multicast situation where a cell is replicated into multiple copies in MGN1, MGN2, or both, and these copies are sent to multiple outputs. Figure 9.14 shows an example to illustrate how a cell is replicated in the MGNs and duplicated in the CD. Suppose a cell arrives at an input port i and is to be multicast to four output ports: #1, #M, #$(M + 1)$, and #N. The cell is first broadcast to all K groups in MGN1, but only the groups, #1, #2, and #K accept the cell. Note that only one copy of the cell will appear in each group, and the replicated cell can appear at any of the $L_1 \times M$ links. The copy of the cell at the output of group #1 is again replicated into two copies at MGN2. There are, in total, four replicated cells that are created after the MGN2. When each replicated cell arrives at the OPC, it can be further duplicated into multiple copies by the CD as needed. Each duplicated copy at the OPC is updated with a new VCI obtained from a translation table at the OPC before it is sent to the network. For instance, two copies are generated at output port #1 and three copies at output port #$(M + 1)$. The reason for using the CD is to reduce the output port buffer size by storing only one copy of the multicast cell at each output port instead of storing multiple copies that are generated from the same traffic source and multicast to multiple virtual circuits on an output port. Also note that since there are no buffers in both MGN1 and MGN2, the replicated cells from either MGN are aligned in time. However, the final duplicated cells at the output ports may not be aligned in time because they may have different queuing delays in the output buffers.

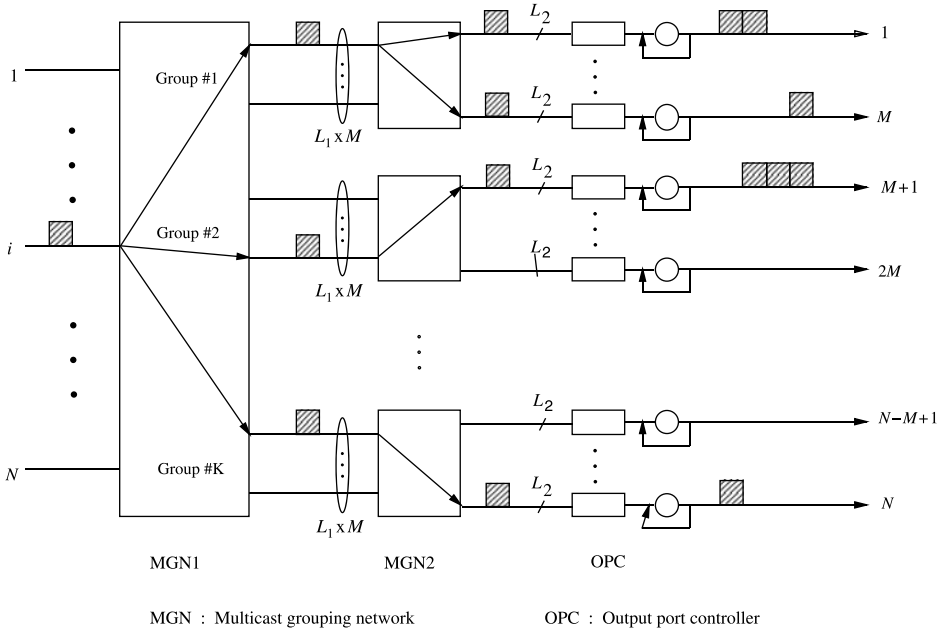


Figure 9.14 Example of replicating cells for a multicast connection in the MOBAS (©1995 IEEE).

9.3.2 Multicast Grouping Network (MGN)

Figure 9.15 shows a modular structure for the MGN at the first or the second stage. The MGN consists of K switch modules for the first stage or M for the second stage. Each switch module contains a switch element (SWE) array, a number of multicast pattern maskers (MPM), and an address broadcaster (AB). The AB generates dummy cells that have the same destination address as the output. This enables cell switching to be done in a distributed manner and permits the SWE not to store output group address information, which simplifies the circuit complexity of the SWE significantly and results in higher VLSI integration density. Since the structure and operation for MGN1 and MGN2 are identical, only MGN1 is described.

Each switch module in MGN1 has N horizontal input lines and $L_1 \times M$ vertical routing links, where $M = N/K$. These routing links are shared by the cells that are destined for the same output group of a switch module. Each input line is connected to all switch modules, allowing a cell from each input line to be broadcast to all K switch modules.

The routing information carried in front of each arriving cell is a multicast pattern, which is a bit map of all the outputs in the MGN. Each bit indicates if the cell is to be sent to the associated output group. For instance, let us consider a multicast switch with 1024 inputs and 1024 outputs and the number of groups in MGN1 and MGN2, K and M, are both chosen to be 32. Thus, the multicast pattern in both MGN1 and MGN2 has 32 bits. For a unicast cell, the multicast pattern is basically a flattened output address (i.e., a decoded output address) in which only one bit is set to '1' and all the other 31 bits are set to '0'. For a multicast cell, there is more than one bit in the multicast pattern set to '1'. For instance, if

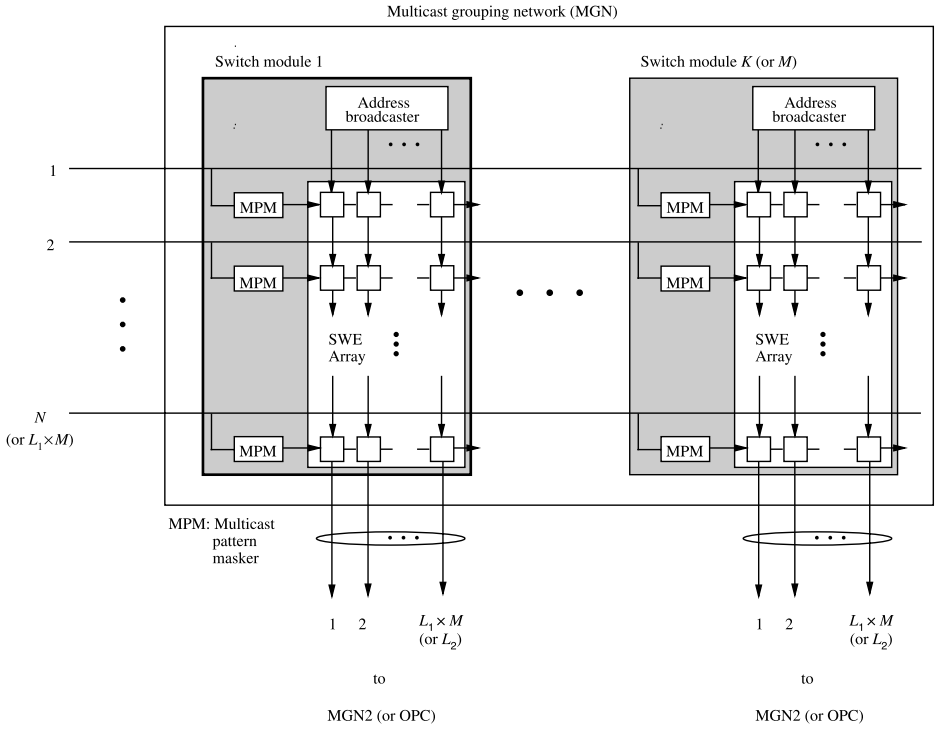


Figure 9.15 Multicast grouping network (MGN) (©1995 IEEE).

a cell, X, is multicast to switch modules i and j, the ith and jth bits in the multicast pattern are set to '1'.

The MPM performs a logic AND function for the multicast pattern with a fixed 32-bit pattern in which only the ith bit, corresponding to switch module i, is set to '1' and all the other 31 bits are set to '0'. So, after cell X passes through the MPM in switch module i, its multicast pattern becomes a flattened output address where only the ith bit is set to '1'.

Each empty cell that is transmitted from the address broadcaster (AB) is attached, in the front, a flattened output address with only one bit set to '1'. For example, empty cells from the AB in switch module i have only the ith bit set to '1' in their flattened address. Cells from horizontal inputs will be properly routed to different switch modules based on the result of matching their multicast patterns with empty cells' flattened addresses. For cell X, since its ith and jth bits in the multicast pattern are both set to '1', it matches with the flattened addresses of empty cells from the ABs in switch modules i and j. Thus, cell X will be routed to the output of these two switch modules.

The SWE has two states, cross state and toggled state, as shown in Figure 9.16. The state of the SWE depends on the comparison result of the flattened addresses and the priority fields in cell headers. The priority is used for the cell contention resolution. Normally, the SWE is at cross state, that is, cells from the north side are routed to the south side, and cells from the west side are routed to the east side. When the flattened address of the cell from the west (FA_w) is matched with the flattened address of the cell from the north (FA_n), and when the west's priority level (P_w) is higher than the north's (P_n), the SWE's state is toggled; the cell from the west side is routed to the south side, and the cell from the north is routed to the east. In other words, any unmatched or lower-priority (including the same priority) cells from the west side are always routed to the east side. Each SWE introduces a 1-bit delay as the bit stream of cells passes it in either direction. Cells from MPMs and AB

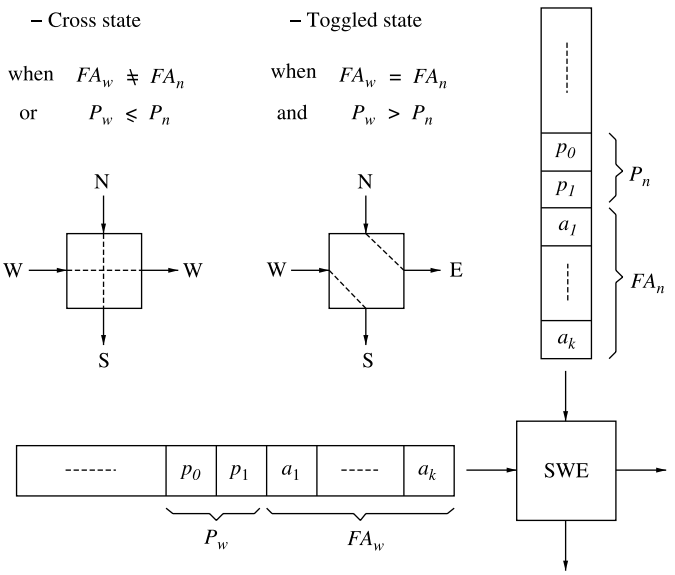


Figure 9.16 Switching condition of the switch element (SWE).

are skewed by one bit before they are sent to each SWE array, due to the timing alignment requirement.

Figure 9.17 shows an example of how cells are routed in a switch module. Cells U, V, W, X, Y, and Z arrive at inputs 1 to 6, respectively, and are to be routed in switch module #3. In the cell header, there is a 3-bit multicast pattern ($m_3\ m_2\ m_1$) and a 2-bit priority filed ($p_1\ p_0$). If a cell is to be sent to an output of this switch module, its m_3 bit will be set to '1'. Among these six cells, cells U, V, and X are for unicast where only one bit in the multicast pattern is set to '1'. The other three cells are for multicast, where more than one bit in the multicast pattern is set to '1'. It is assumed that a smaller priority value has a higher priority level. For instance, cell Z has the highest priority level ('00') and empty cells transmitted from the address broadcaster have the lowest priority level ('11'). The MPM performs a logic AND function for each cell's multicast pattern with a fixed pattern of '100'. For instance, after cell W passes through the MPM, its multicast pattern ('110') becomes '100' ($a_3\ a_2\ a_1$), which has only one bit set to '1' and is denoted as a flattened address. When cells are routed in the SWE array, their routing paths are determined by the state of SWEs, which are controlled according to the rules in Figure 9.16. Since cells V and X are not destined for this group, the SWEs they pass remain in a cross state. Consequently, they are routed to the right side of the module and are discarded. Since there are only three routing links

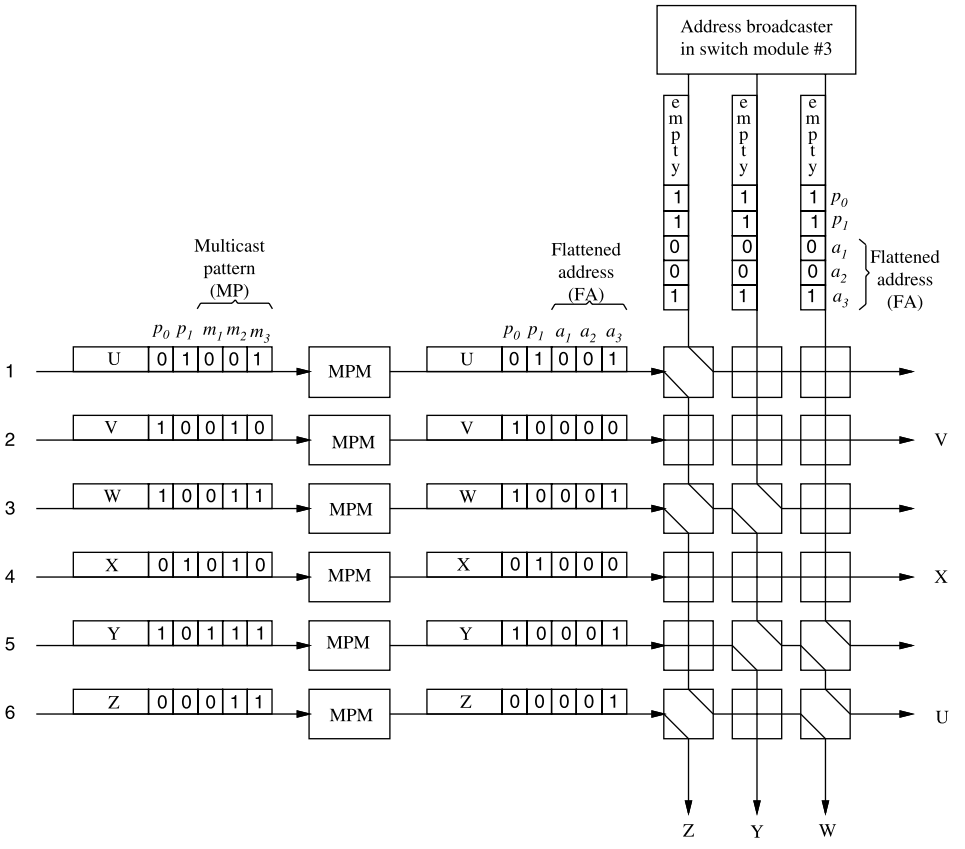


Figure 9.17 Example of routing a multicast cell.

in this example, while there are four cells destined to this switch module, the one with the lowest priority (i.e., cell U) loses the contention to the other three and is discarded.

Since the crossbar structure inherits the characteristics of identical and short interconnection wires between switch elements, the timing alignment for the signals at each SWE is much easier than that of other types of interconnection network, such as the binary network, the Clos network, and so on. The unequal length of the interconnection wires increases the difficulty of synchronizing the signals, and, consequently limits the switch fabric's size, such as the Batcher-banyan switch. The SWEs in the switch modules only communicate locally with their neighbors, as do the chips that contain a two-dimensional SWE array. The switch chips do not need to drive long wires to other chips on the same printed circuit board. Note that synchronization of data signals at each SWE is only required in each individual switch module but not in the entire switch fabric.

9.4 APPENDIX

Let us consider an ATM switch shown in Figure 9.10 and assume that cells arrive independently from different input ports and are uniformly delivered to all output ports. The variables used are defined as follows.

- N Number of a switch's input ports or output ports.
- M Number of output ports that are in the same group.
- L Group expansion ratio.
- ρ Offered load of each input port, or the average number of cells that arrive at the input port in each cell time slot.
- ρ/N Average number of cells from each input port destined for an output port in each cell time slot.
- $\rho M/N$ Number of cells from each input port destined for an output group in each cell time slot.
- P_k Probability of k cells arriving at an output group in each cell time slot.
- λ Average number of cells from all input ports that are destined for an output group in each cell time slot.
- λ' Average number for cells from all input ports that have arrived at an output group in each cell time slot.

The P_k is given by the following binomial probability,

$$P_k = \binom{N}{k} \left(\frac{\rho M}{N}\right)^k \left(1 - \frac{\rho M}{N}\right)^{N-k} \qquad k = 0, 1, \ldots, N.$$

$$\lambda = \sum_{k=1}^{N} (kP_k) = N\left(\frac{\rho M}{N}\right) = \rho M$$

$$\lambda' = \sum_{k=1}^{LM} kP_k + \sum_{k=LM+1}^{N} (LM)\, P_k$$

$$= \lambda - \sum_{k=LM+1}^{N} kP_k + \sum_{k=LM+1}^{N} (LM)P_k$$

$$= \lambda - \sum_{k=LM+1}^{N} (k - LM)P_k.$$

Since, at most $L \times M$ cells are sent to each output group in each cell time slot, the excess cells will be discarded and lost. The cell loss probability is:

$$\begin{aligned} P\,(\text{cell loss}) &= \frac{\lambda - \lambda'}{\lambda} \\ &= \frac{1}{\lambda} \sum_{k=LM+1}^{N} (k - LM)P_k \\ &= \frac{1}{\rho M} \sum_{k=LM+1}^{N} (k - LM) \\ &\quad \cdot \binom{N}{k} \left(\frac{\rho M}{N}\right)^k \left(1 - \frac{\rho M}{N}\right)^{N-k}. \end{aligned} \tag{9.8}$$

As $N \to \infty$,

$$P_k = \frac{(\rho M)^k\, e^{-\rho M}}{k!}$$

$$\begin{aligned} P(\text{cell loss}) &= \frac{1}{\rho M} \sum_{k=LM+1}^{\infty} (k - LM) \frac{(\rho M)^k\, e^{-\rho M}}{k!} \\ &= \sum_{k=LM+1}^{\infty} \frac{(k - LM)}{\rho M} \frac{(\rho M)^k\, e^{-\rho M}}{k!} \\ &= \sum_{k=LM+1}^{\infty} \frac{(\rho M)^{k-1}\, e^{-\rho M}}{(k-1)!} \\ &\quad - \frac{L}{\rho} \sum_{k=LM+1}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!} \\ &= \sum_{k=LM}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!} - \frac{L}{\rho} \sum_{k=LM+1}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!} \\ &= \frac{(\rho M)^{LM}\, e^{-\rho M}}{(LM)!} + \sum_{k=LM+1}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!} \\ &\quad - \frac{L}{\rho} \sum_{k=LM+1}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!} \end{aligned}$$

$$
\begin{aligned}
&= \left(1 - \frac{L}{\rho}\right)\left(\sum_{k=LM+1}^{\infty} \frac{(\rho M)^k\, e^{-\rho M}}{k!}\right) \\
&\quad + \frac{(\rho M)^{LM}\, e^{-\rho M}}{(LM)!} \\
&= \left(1 - \frac{L}{\rho}\right)\left(1 - \sum_{k=0}^{LM} \frac{(\rho M)^k\, e^{-\rho M}}{k!}\right) \\
&\quad + \frac{(\rho M)^{LM}\, e^{-\rho M}}{(LM)!}
\end{aligned} \tag{9.9}
$$

REFERENCES

[1] Y.-S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: A simple, modular architecture for high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 8, pp. 1274–1283 (Oct. 1987).

[2] W. Hoeffding, "On the distribution of the number of successes in independent trials," *Ann. Math. Statist.*, vol. 27, pp. 713–721 (1956).

[3] A. Pattavina, "Multichannel bandwidth allocation in a broadband packet switch," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1489–1499 (Dec. 1988).

[4] M. G. Hluchyj and M. J. Karol, "Queueing in high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1587–1597 (Dec. 1988).

[5] S. C. Liew and K. W. Lu, "Performance analysis of asymmetric packet switch modules with channel grouping," in *Proc. IEEE INFOCOM'90*, San Francisco, California, pp. 668–676 (June 1990).

[6] Y. Oie, M. Murata, K. Kubota, and H. Miyahara, "Effect of speedup in nonblocking packet switch," in *Proc. ICC'89*, Boston, Massachusetts, pp. 410–415 (June 1989).

[7] K. Y. Eng, M. J. Karol, and Y.-S. Yeh, "A growable packet (ATM) switch architecture: Design principles and applications," *IEEE Transactions on Communications*, vol. 40, no. 2, pp. 423–430 (Feb. 1992).

[8] H. J. Chao, "A recursive modular terabit/second ATM switch," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1161–1172 (Oct. 1991).