PERFORMANCE ANALYSIS OF VIRTUAL PATH OVER LARGE-SCALE ATM SWITCHES

By

Tang Oo

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Kong

December 1997

Abstract

A quasi-static routing scheme, called path switching, implemented in the three-stage Clos network has been proposed in [34]. It uses periodical connection patterns at the central stage, input queueing at the first stage and output queueing at the last stage. Traffics are multiplexed on a virtual path in input modules, which consists of all the virtual connections from an input module to an output module. The throughput is limited by the first stage, if space-division switch module is used, and can be made arbitrarily close to 100% with large number of central modules. However, the loss probability due to output contention will also increase with the number of central modules. We propose a simple virtual path scheduling scheme to achieve high throughput while lowering loss probability. The key idea of our method is to schedule the arrivals at the last stage of virtual paths such that only a limited number of contenders will be allowed at any output at any time. The scheduling is fulfilled by logical partition and proper route assignment of virtual path, which will be discussed fully in chapter 2. Compared with other methods, the scheduling scheme is more flexible to manage the virtual path.

The performance of virtual path under scheduling scheme, including maximum throughput and concentration loss, is evaluated with the assumption of

independent and uniform traffic. The concentration loss will be reduced by the partition of virtual path. However, the maximum throughput will be degraded if the input stage is implemented by space-division switch. The maximum throughput is obtained by simulation with various parameter settings. Several schemes such as look-ahead, input smoothing and complex buffer management are proposed to improve the throughput. If memory switch is used at input stage, the throughput could be as high as 100% even with bursty traffic. However, the multiplexing gain will decrease when the virtual path is split, because less sources share buffer and bandwidth on a partitioned virtual path. The loss of multiplexing gain is estimated by assuming on-off sources at the input module.

The path switch is both input and output buffered, and cells can be dropped at both stages. Two switching mechanisms, queue loss and backpressure mode, are assumed to study the buffer dimensioning and cell loss probability due to buffer overflow. The effect of backpressure upon the maximum throughput is obtained by simulation where the buffer is dedicated to each input and output port. The cell loss probability is compared under queue loss and backpressure mode with different buffer allocation among input and output.

大規模ATM交換機上的虛路徑性能分析

摘要： 文獻[34]中提出的虛路徑交換是一種實現于三級Clos網絡上的准靜態交換方案。其中間級采用周期連接方式，第一級采用輸入緩沖，第三級采用輸出緩沖。從一個輸入模塊到一個輸出模塊的所有虛電路組成了一條虛路徑，這些虛電路在輸入級被复接到虛路徑上。如果第一級采用空分交換，流量會受到限制。如果有足夠的中間級模塊，流量能接近100%，但是輸出級由於碰撞產生的信元丟失率會隨之上升。我們提出一個簡單的虛路徑調度方案，可以減低踫撞丟失率，同時保持大的流量。方案的關鍵是如何安排到達最後一級的信元，使得僅有有限數目的踫撞能夠同時發生。這種調度由虛路徑的邏輯劃分和正確的路由分配完成。与其它方法相比較，這個方案可以更加靈活地管理虛路徑。

針對于不相關和均勻的業務，我們分析了虛路徑的性能，包括最大流量和踫撞丟失率。踫撞丟失率能夠被虛路徑的劃分減低，但是，最大流量會受到損失。不同參數的最大流量可由仿真得到，并且可以利用以下幾種方案對之加以改善，例如：前視，輸入端口擴展和內存管理等。如果第一級采用存儲交換，流量可以達到100%。然而，當虛路徑被分細時，复用增益會被降低，因爲在一條被分細的虛路徑上，共享內存和帶寬的信源數目減少了。

對于輸入信號爲開-關信號的情況，我們估計了其复用增益的損失。因爲虛路徑交換機有輸入和輸出緩存，信元可能在輸入和輸出端被丟失。兩種交換模式：排隊丟失和反饋等待被用來研究內存的設置和溢出概率。通過仿真，我們得到了反饋等待對最大流量產生的影響。當輸入和輸出緩存采用不同設置時，我們比較了這兩種模式下的信元丟失率。

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Asynchronous Transfer Mode(ATM) is being developed by the ITU as part of the Broadband Integrated Services Digital Networks(B-ISDN) switching technology for future mixed telephone and data networks. B-ISDN is designed to provide subscriber communication services over a wide range of bit rates from a few megabits per second to several gigabits per second. In an ATM network, data is fragmented into cells before being sent on the transmission links. The motivation of cell-based network is to: (a) support multiple types of services; (b) reduce the number of transmission networks; (c) provide easier support for multicasting; and (d) offer a better multiplexing scheme than ISDN for higher utilization of network sources. ATM operates in a connection-oriented mode. A logical/virtual connection is set up between end points and necessary network resources(bandwidth and buffer) are reserved along the route. Each connection is characterized by a Virtual Channel Identifier(VCI) assigned at call setup. Since

there may be large number of simultaneous connections between two end-points, a semi-permanent connection can be established between end-points to allow efficient and simple management of available resources. This virtual connection is known as virtual path and is identified by the Virtual Path Identifier(VPI). Cells of a source are identified by both VCI and VPI in their header, and are multiplexed with cells of other sources on a virtual path which may traverse several physical links. The basic concept of virtual channel and virtual path is illustrated in figure 1.1. Even the state of individual virtual circuits may change quickly from time to time, the state of virtual path will keep quasi-static because of the superposition of traffic. This makes it possible that the reconfiguration of virtual path is relatively less frequent.

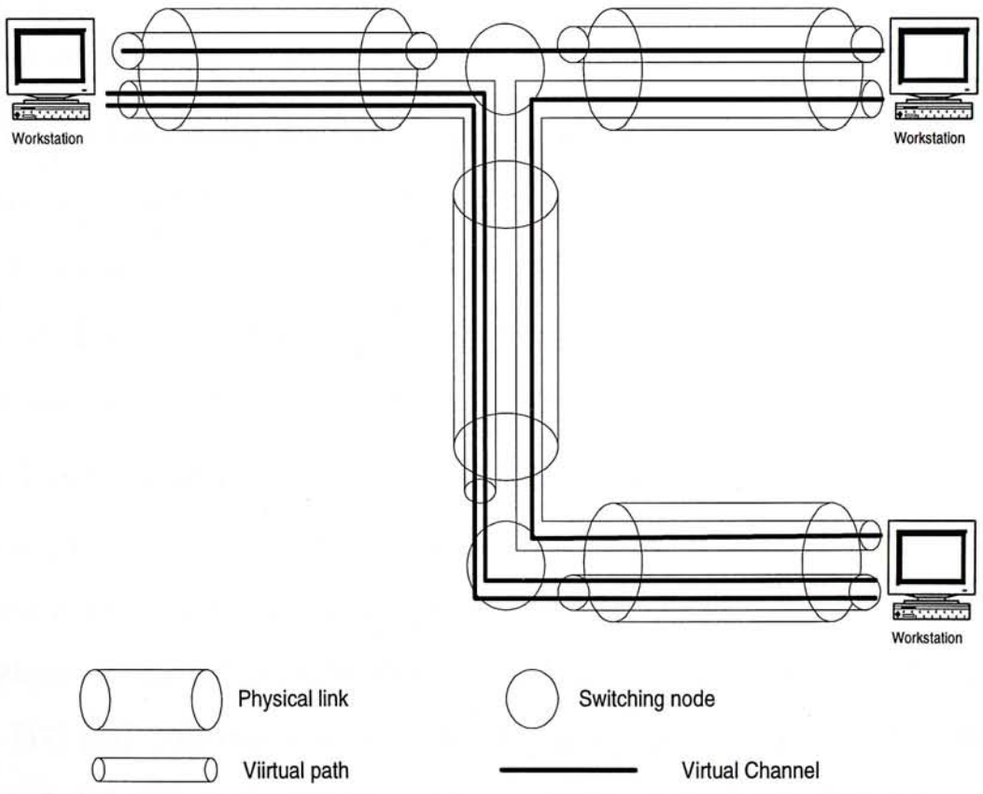

Figure 1.1: The Concept of Virtual Channel and Virtual Path

At intermediate switching nodes, cells are stored and forwarded to individual outlet according to the local routing information established at call setup. The Quality of Service(QOS) of each connection, such as cell loss rate, maximum cell transfer delay and peak-to-peak cell delay variation, must be satisfied at switches. It may happen that many cells simultaneously arrive at a buffer, causing congestion or buffer overflow, or many cells are destined to the same output link, causing output contention. Those cells congested at the buffer will suffer from queueing delay; and those which can not be stored or have lost contention will be dropped. The typical value for cell loss probability ranges from 10^{-8} to 10^{-11} [51]. These requirements challenge the performance of switches in terms of throughput and cell loss probability. In addition, the switch must be high-speed and large-scale to keep up with the command of ever-developing services and networks.

A lot of switch architectures have been studied, which can be classified into two classes: space-division switch and shared-buffering memory switch. The space-division switch is made up of cross points(2×2 switching elements), and can be further divided into three types by queueing discipline: input queueing, output queueing and input-output queueing.

An input queueing switch architecture is shown in figure 1.2. A separate buffer is placed at each input port and cells wait at the buffer for access to the output port. If the input buffer is served first-in first-out(FIFO), then the throughput is limited to 0.586 due to the head of line blocking under uniform traffic [17] [23]. An example is shown in figure 1.2. At input port 1, the first cell can not be cleared because it loses contention with the first cell at input port 3. Then the next cell destined for output 2 is blocked by it even that the

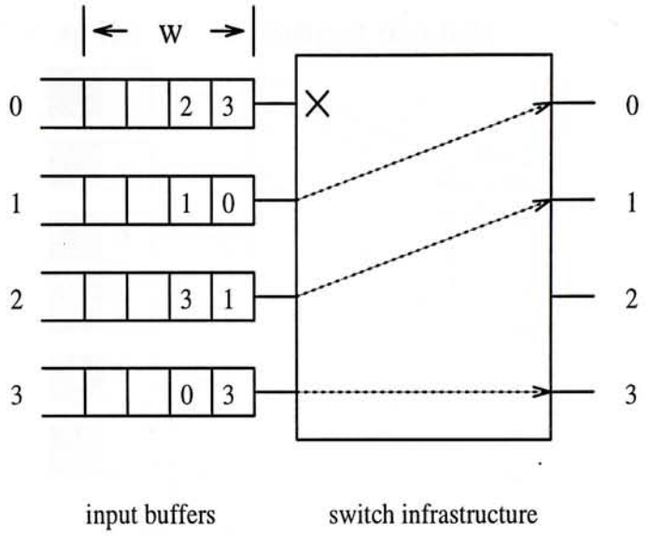

Figure 1.2: The Input Queueing Switch and Look-ahead Scheme

output 2 is idle at this time slot. The throughput can be increased by relaxing the strict first-in first-out discipline of input buffers, e.g., incorporating a look-ahead contention resolution scheme. During a time slot, the first w cells in each input queue will sequentially contend for idle output ports, till a cell in the queue is selected or the contention resolution process repeats w times. The parameter w is the look-ahead window size. If the look-ahead contention resolution scheme is used, the second cell of input 1 to output port 2 will not be blocked and can be selected to transmit at the look-ahead step 2. The analysis and simulation results of throughput as a function of w can be found in [17] [33].

The output queueing switch, such as the well-known knockout switch [24] and Batcher-R-banyan network with output expansion(Starlite) [33] [44], gives the best performance in terms of throughput and delay. In a knockout switch, for example, each output port can accept R simultaneous arrivals where R is the group size. It is possible that more than R cells contend the same output port. In that case, the excess ones will be dropped immediately. This is known

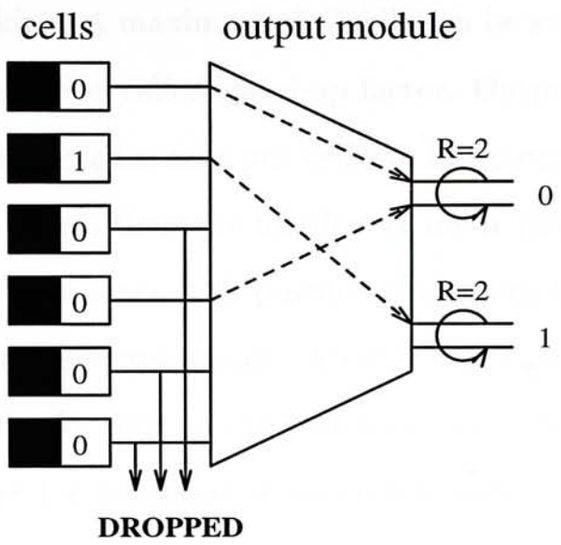

Figure 1.3: Output Contention

as output contention. An example is shown in figure 1.3, in which five cells contend for port 0. Since the group size is equal to two, three of the cells are dropped. One output port can be dedicated a buffer, or several output ports can share a buffer to achieve higher efficiency and lower overflow probability [43].

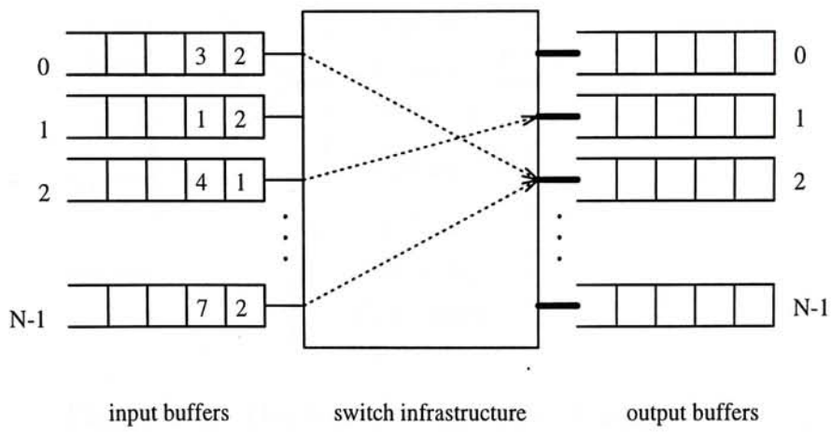

Figure 1.4: The Input-output Queueing Switch

The input-output queueing switch shown in figure 1.4 has been studied in [7] [21] [40] [50]. If the switch operates at S times the input link rate, more than one cells can be cleared from an input port in one time slot to alleviate

the head of line blocking. A maximum of S cells can be switched to one output port at a time slot and S is called speed-up factor. Output buffer is needed to accommodate arrivals, because only one cell can be transmitted to output link in a time slot. If S is less than the number of input ports, input queues are needed to avoid cell loss, since it is possible that more than S cells at input buffers are destined to one output port. Arrivals that can not be delivered will wait at the input buffer for retry in the next time slot instead of being dropped. As illustrated in figure 1.4, two head-of-line cells to output port 2 can be cleared at one time slot with speed-up factor of 2, and excess ones have to wait at the input buffer. Likewise, we can place input buffer to an output-expansion switch to store cells which can not be accepted at output port. Also there may be some sharing among input or output buffer to lower cell loss rate. The input-output queueing switch can reach high throughput with speed-up or output expansion.

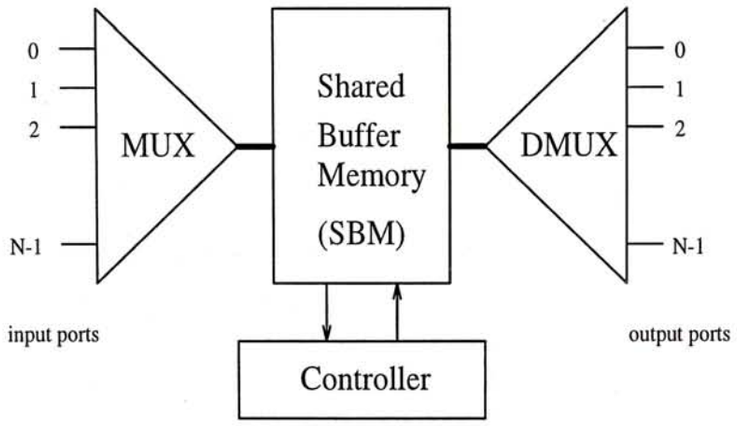

Figure 1.5: The Shared-buffering Memory Switch

The architecture of shared-buffering memory switch is illustrated in figure 1.5 [10] [14] [54] [62]. Incoming ATM cells are time-multiplexed in the MUX block and stored in the shared buffer memory(SBM). Physically all stored ATM cells share the whole memory, but a separate queue is formed logically for each output

port, thereby realizing the self-routing function. Every cell is written into its corresponding logical queue and read out from the SBM, demultiplexed in the DMUX block to output port. Since a separate queue is formed for an output port, there is no head of line blocking, so the throughput can be as high as 100%. The buffer is shared by all output ports, thereby achieving efficient buffer utilization and small cell loss ratio. However, the access speed of memory is proportional to the switch size and transmission speed, since the memory must be read and written by all input and output ports during a time slot. Thus the operating speed is the bottleneck for a large-scale and high-speed memory switch.

Because the physical size of switch module is constrained by the VLSI technology and the processing speed of a central controller, they can be interconnected to construct large-scale, multistage switch network such as Clos network [8], Benes network [4] and Cantor network [5] etc. Due to the stochastic characteristics of traffic, routing in a switch may change at every time slot. The routing assignment is on a slot-by-slot basis according to the global information of arrivals at the switch. Therefore the processing speed of central controller is a bottleneck which severely restricts the growth of switch size and speed. In addition, the Quality of Service(QOS) of virtual connections can not be guaranteed in most of switches. On the other hand, the opposite of dynamic routing is static routing such as circuit switch, in which the routing pattern of switch is not changed on the fly. However, the efficiency of resource utilization is low, because peak rate is reserved and dedicated to a source so that there is no multiplexing. Especially under multimedia environment in which the sources are of multirate, the non-blocking condition is stringent by static routing in circuit

switch. A distributed routing scheme–path switching has been proposed by Lee [34] to handle the above issues.

1.2 The Concept of Cross-Path Switching

The cross-path switch proposed in [34] is a large-scale ATM switch architecture which adopts quasi-static routing scheme in the three-stage Clos network. As shown in figure 1.6, the three-stage Clos network has M modules at the central stage and K modules at each outer stage. The size of each input (output) module is $N \times M$ ($M \times N$), where $M/N \geq 1$ is defined as the expansion factor. The input(output) links of central modules are called central links from now on. At any time, a central module can be assigned to an input-output module pair only once, and there are up to M alternative routes between any input-output module pair. Instead of routing all arrivals with a central controller on the fly, the routing could be implemented in a distributed manner over three stages of the Clos network.

The cross-path switching uses predetermined, periodical connection patterns in the central stage, input queueing in the first stage, and output queueing in the last stage. The connection patterns of central modules are repeated in each frame, which consists of F consecutive time slots. A virtual path between an input module and an output module comprises all virtual circuits connecting them. The scheduling of path switching involves capacity assignment and route assignment of all virtual paths between input and output modules. Since a maximum of M cells can be switched from an input module by central stage in a slot, the total free capacity is M. The required service rate of virtual path form

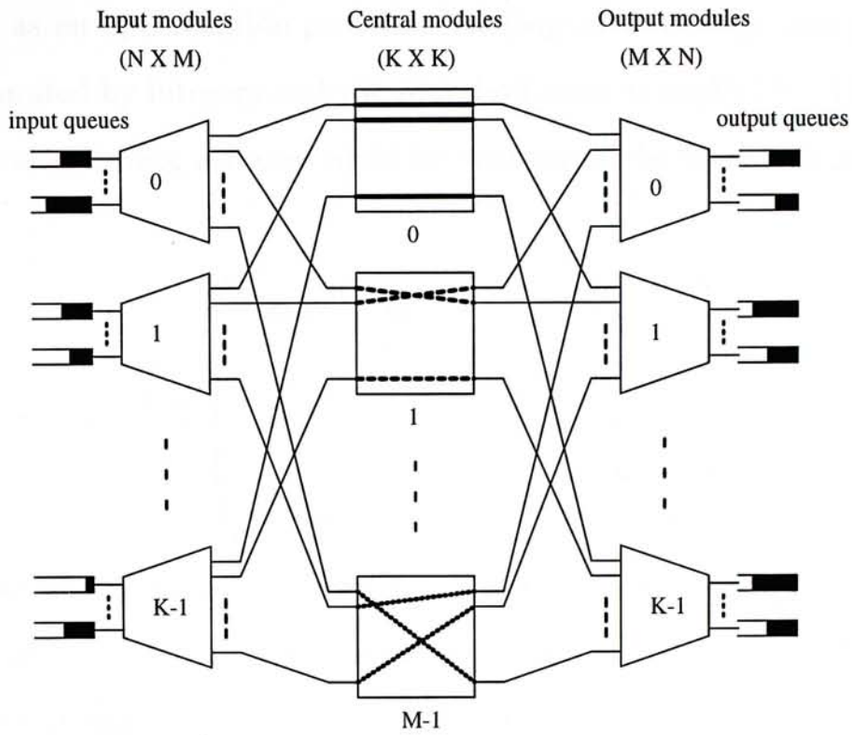

Figure 1.6: The Three-Stage Clos Network

input module i to output module k, $0 < \lambda_{i,k} \leq M$ cells/slot, is determined by the stochastic characteristics of the aggregate traffic multiplexed on it and their QOS, which will be discussed later in chapter 4. The capacity assignment is to find $C_{i,k}$ subject to $C_{i,k} \geq \lambda_{i,k}$ and $\sum_{i=0}^{K-1} C_{i,k} = \sum_{k=0}^{K-1} C_{i,k} = M$ by a certain criterion. The assigned capacity of virtual path, $C_{i,k}$, is defined as the average number of cells that can be delivered by central stage at a time slot, or the average number of central modules assigned to the virtual path at a slot. However, $C_{i,k}$ may not be integer after capacity assignment. Since the connection pattern of virtual paths is repeated in frames, the service discipline is like weighted round robin with period F. If $e_{i,k}$, the number of tokens during a frame is known, $C_{i,k}$, the average service rate of virtual path is equal to $e_{i,k}/F$. Then we only need to find $e_{i,k}$ subject to $e_{i,k}/F \geq \lambda_{i,k}$ and $\sum_{i=0}^{K-1} e_{i,k} = \sum_{k=0}^{K-1} e_{i,k} = FM$, which can be

formulated as an optimization problem. As long as F is large enough, $e_{i,k}$ can be approximated by integers and the round-off error is negligible. The capacity of virtual paths during a frame could be written in the form of a matrix E as follows:

$$E = \begin{pmatrix} e_{0,0} & e_{0,1} & \cdots & e_{0,K-1} \\ e_{1,0} & e_{1,1} & \cdots & e_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ e_{K-1,0} & e_{K-1,1} & \cdots & e_{K-1,K-1} \end{pmatrix}$$

Each element of matrix E denotes the number of central modules assigned to the virtual path. Because capacity is reserved for each virtual path, the QOS at path level is guaranteed.

The route assignment is to determine the connection pattern of all the central modules in each time slot after capacity assignment, i.e., the number of central modules assigned to a virtual path is known. If the Clos network at F time slots are put together to construct one large Clos network with FM central modules, the route assignment is equivalent to determine the routing pattern of FM central modules at one instant given the input-output pair connections. It is fulfilled by the edge-coloring of a regular bipartite graph with degree FM [35]. This point is illustrated in figure 1.7 which represents the connection patterns of central modules in figure 1.6. Nodes on the left denote the input modules and nodes on the right, the output modules. The number of edges joining two nodes is equal to the capacity of the virtual path during a frame. All the adjacent FM edges at a node are colored with FM distinct colors, each of which represents a central module at a certain time slot by time-space interleaving principle [34]. Color $0, 1, \ldots, M-1$ represent central modules at time slot 0; color $M, M+$

$1, \ldots, 2M-1$ represent time slot 1; ...; color $(F-1)M, (F-1)M+1, \ldots, FM-1$ represent time slot $F-1$. Once the color of each edge is known, the connection pattern of central modules at each time slot is determined by the time-space interleaving principle.

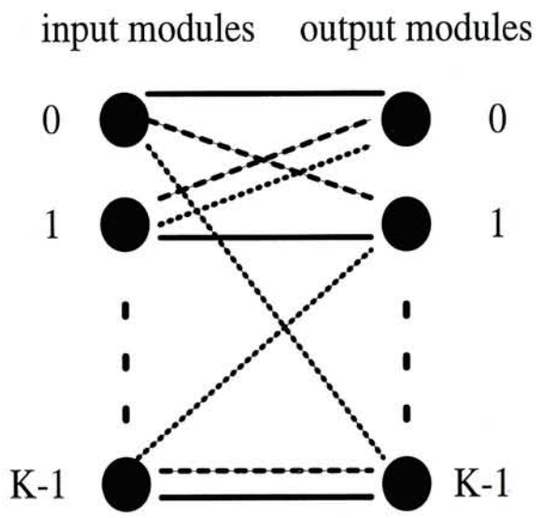

Figure 1.7: The Corresponding Bipartite Graph of Route Assignment

Knowing the connection pattern of central modules, the input modules will select matched cells waiting at input buffer and deliver them to the central stage, where cells will be switched to their destined output module by predetermined routes. The output module will handle the contention among arrivals and route them to individual output port. Therefore the routing is distributed among all the input and output modules, without the involvement of central controller. This makes large-scale ATM switch built on modules possible, since routing is not the bottleneck any more.

1.3 Contribution and Organization of Thesis

Up to now, the "width" of virtual path is defined by the physical parameters of Clos network, e.g., N, M and K. Once these parameters are fixed, it is impossible to manage the virtual path flexibly. However, the restriction is relaxed by logical partition of virtual path and the "width" of those sub-paths could be variable. This is the virtual path scheduling scheme proposed in the next chapter. The performance of path switch including throughput and concentration loss probability, has been studied in [34]. A trade-off between throughput and concentration loss rate is revealed that they will both increase with the number of central modules. In the next chapter, it will be illustrated that by proper route assignment of partitioned virtual path, the issue is solved by the virtual path scheduling scheme which can lower concentration loss rate while keeping throughput high. It is flexible to meet different cell loss requirement while the implementation is rather simple.

With the virtual path scheduling scheme, the improvement of concentration loss is analyzed by knockout principle. The maximum throughput is obtained numerically and several methods to improve it are discussed in chapter 3. These results could be used for optimization of switch design. With different parameter settings, the cell loss probability at input and output buffer is obtained by simulation. Given the number of buffer budget, a proper allocation among input and output should be performed according to the feedback strategy and maximum throughput. The buffer dimensioning under backpressure and queue loss mode, are compared in chapter 3.

If the switch size is fixed, the larger is N, the smaller is K, and the number

of virtual path per input module, which is equal to K, is smaller too. With larger input(output) module size, the efficiency of resource utilization is improved because more virtual circuits are multiplexed on the virtual path and share capacity. It seems that the module size should be as large as possible to achieve high efficiency. However, how much is the multiplexing gain improved by increasing module size? It is possible that the multiplexing gain rise quickly with the module size within a certain range. Beyond the range, the improvement is marginal. If this is true, the module size should not be as large as possible, because the cost rises sharply with the size. Even there may not be an optimal solution, it is still of significance to evaluate the multiplexing gain of virtual path, which is presented in chapter 4.

Chapter 2

The Virtual Path Scheduling Scheme

2.1 The Trade-off Between Throughput and Concentration Loss

The modules at the first stage are input-buffered switches, which will store the arrivals that can not be cleared immediately. The throughput is mainly limited at the first stage due to the head of line blocking. To alleviate the head of line blocking, the look-ahead arbitration is performed to select those packets whose addresses match with the connection pattern at that time slot. The first w cells at input buffer will contend for idle output ports sequentially during a contention resolution cycle. It is possible that no matched cell can be found after searching the whole window in all the input buffers of an input module, so the central links assigned to the virtual path are wasted. Obviously, the deeper

is the look-ahead search, the higher is the throughput. When the window size w tends to infinity, the throughput approaches 100%. However, the window size is limited by the speed of processor and the improvement of throughput is trivial after w is beyond a certain value [17].

The maximum throughput of the first stage is also a monotonic increasing function of the expansion factor M/N. Because the arbitration of cells at the first stag is based on their output module, the number of virtual paths of each input module equals to the number of output modules under uniform traffic. For constant K, more routes could be assigned to a virtual path if more central modules were provided. This is equivalent to channel grouping method which expands the number of output ports of a physical address. The throughput surges up since the head-of-line blocking is alleviated and more cells destined for the same output module can be delivered at the same time. A full discussion of this point and the specific analysis could be found in [43].

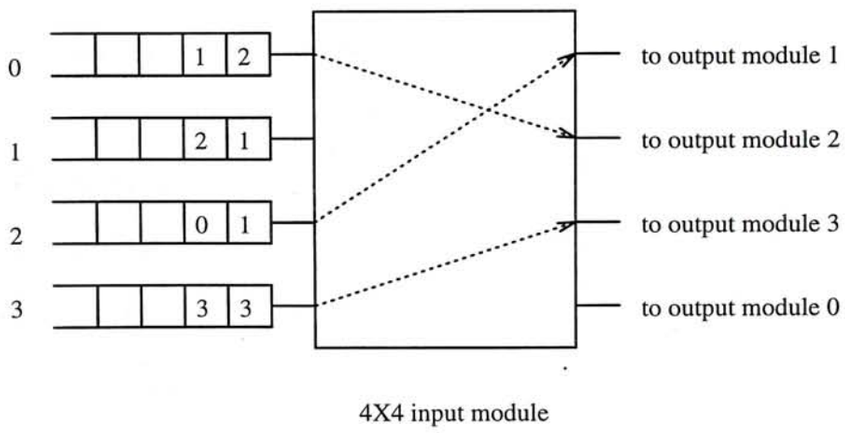

Figure 2.1: Lower Throughput with 4 Central Modules

An input module is shown in figure 2.1, with size of 4×4. If the number of output modules, K, is also 4, then there is only one central module assigned to a virtual path in one time slot under homogeneous traffic. Maximum of 1

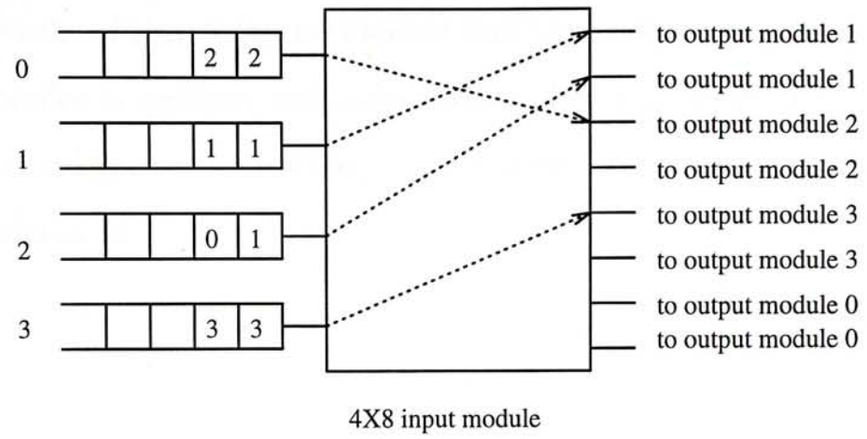

Figure 2.2: Higher Throughput with 8 Central Modules

cell can be delivered to one output module through the path, so the first cell at input link 1 has to wait for retry because it loses contention with input link 2. If the input module size is altered to 4×8 as in figure 2.2, then two central modules will be assigned to a virtual path at one time slot, so that both the two head cells can be carried on the path simultaneously. The throughput of the first stage can be arbitrarily close to 100% with large number of central modules. However, the loss probability due to contention at the last stage may also increase with M. This trade-off has been discussed in [34], which is fully revealed in the follows.

Modules in the last stage are output buffering with group size R, the maximum number of cells that can be accepted by an output port in one time slot. Since cells in input modules are selected independently, it is possible that a number of cells with the same destination address find their way to the output port simultaneously. Each output module will resolve the contentions among cells destining for the same output port, and those losing contention will be dropped on the floor. In the worst case, there are M contenders at the last stage, and $M - R$ of them would be dropped.

For the sake of simplicity, we assume that the traffic loading on each link of central modules is uniform and independent. Let ρ_{out} be the loading on each central link and ρ_{in} be the loading on each input link of an input module, then they are related by:

$$\rho_{out} = \frac{N}{M} \cdot \rho_{in} \tag{2.1}$$

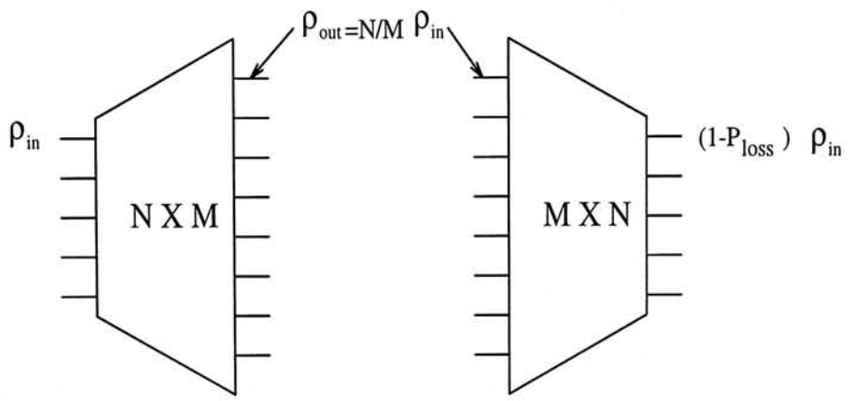

Figure 2.3: The Relation Between Loading at Input Links and Central Links

According to the Knockout principle [24], the loss probability P_{loss} is given by

$$P_{loss} = \frac{1}{\rho_{in}} \cdot \sum_{l=R+1}^{M} (l-R) \binom{M}{l} (\frac{\rho_{out}}{N})^l (1-\frac{\rho_{out}}{N})^{M-l} \tag{2.2}$$

$$= \frac{1}{\rho_{in}} \cdot \sum_{l=R+1}^{M} (l-R) \binom{M}{l} (\frac{\rho_{in}}{M})^l (1-\frac{\rho_{in}}{M})^{M-l} \tag{2.3}$$

Given that the input loading of 0.8 and group size of 8, the loss probability as function of the expansion factor and module size is shown in figure 2.4. It can be observed that P_{loss} increases with the module size quickly. for the same M/N, the loss probability varies in a wide range with different N. For each value of N, the loss probability rises sharply with respect to M/N, when M/N

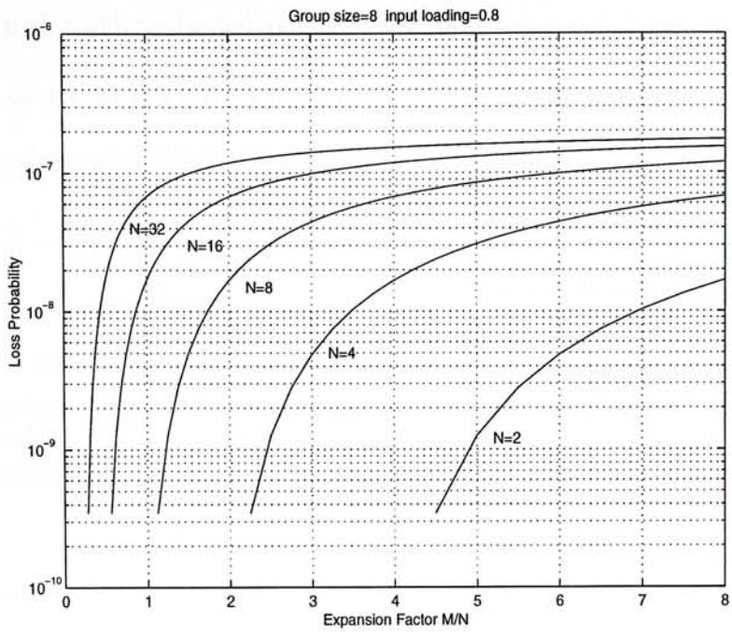

Figure 2.4: The Loss Probability vs. M/N and N with R=8

is less than a certain threshold; beyond the threshold, then P_{loss} rises slowly. For example, P_{loss} is 5×10^{-9} as $M = 12$ and $N = 8$, while it rises to 10^{-7} if $M = 64$ and $N = 32$. From the above analysis, it seems that there are only two approaches to lower the output contention: either reduce N and M/N(i.e. smaller module size) or enlarge group size R. However, for a constant switch size, reducing N would significantly increase K, the number of input(output) modules , and consequently the central module size will be large. The number of virtual path per module is also rising with K, which will degrade the throughput [34] and result in poor statistical multiplexing gain of virtual paths(it will be analyzed later in chapter 4). The latter is not economical because it would bring greater complexity to output modules. Raising R by 1 means to place one more $N \times N$ banyan network in a Batcher-banyan knockout switch architecture shown in figure 2.5, or increase the buffer access speed of a memory switch by 1. Furthermore, once the switch is built, M, N and R can't be altered any

longer. A virtual path scheduling scheme is proposed here to reduce output contentions. It could satisfy different cell loss requirements with great flexibility while achieving desirable high throughput.

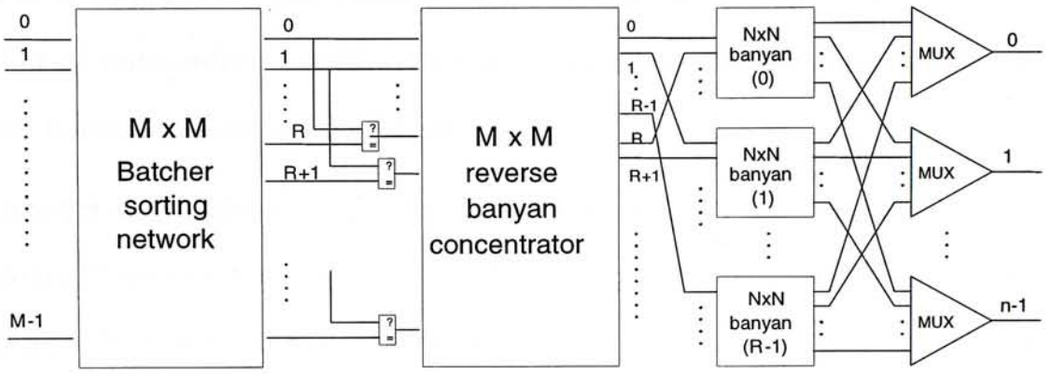

Figure 2.5: The Architecture of the Batcher-R-banyan Knockout Switch

2.2 Partition of Virtual Paths

It is well known that the output contention is due to unscheduled concurrent arrivals. Contention resolution could be performed at input- queueing switch, where blocked cells have to wait at input buffer for re-entry so that no contention occurs at output ports. But the case is different for output queueing, which allows cells with the same destination to arrive simultaneously. Path switching is the combination of input queueing and output queueing, with predetermined and repeated route assignments of every virtual path. Input modules choose suitable cells according to their destination. Excess cells that current route can't carry have to wait for re-entry at later time slots. In this sense, the

virtual path is able to schedule cells, i.e., decide whether to deliver them or not. However, the current virtual path is so "wide" that a number of contenders can pass through it at the same time. The key idea of the virtual path scheduling scheme is to partition the virtual path into sub-paths such that only a limited number of contenders would arrive at an output port simultaneously and the output contention could be alleviated.

In order to partition a virtual path, we logically divide an output module into G "virtual" modules, each of them consists of N/G output ports, assuming that M/G and N/G are integers. Those output ports in the same virtual module will be called an output "cluster". Thus, a virtual path is defined as the connection between an input module (physical) and an output virtual module. That is, we split one original virtual path into G sub-paths. $G = 1$ and $G = N$ are two extreme cases. $G = 1$ is the case of the original path switch without division; $G = N$ means that a cluster comprises only one output port. We assume that the capacity of central stage is equally distributed among G clusters, such that each of them is assigned M/G central modules in every time slot(This is fulfilled by the route assignment presented in the next section). As a result, arrivals at the last stage are scheduled and the number of contenders at each output port is limited to M/G. The idea is illustrated in figure 2.6, in which an 6×2 output module with $R = 2$ is split into two 3×1 virtual ones. The number of contenders is limited to three, and consequently maximum of one cell per output port would be dropped at any time, compared to four without partition.

Loadings from an input module to a cluster are multiplexed on the virtual path and served by assigned rate, as illustrated in figure 2.7. The connection patterns of central modules are known and repeated in a cyclic manner. Each

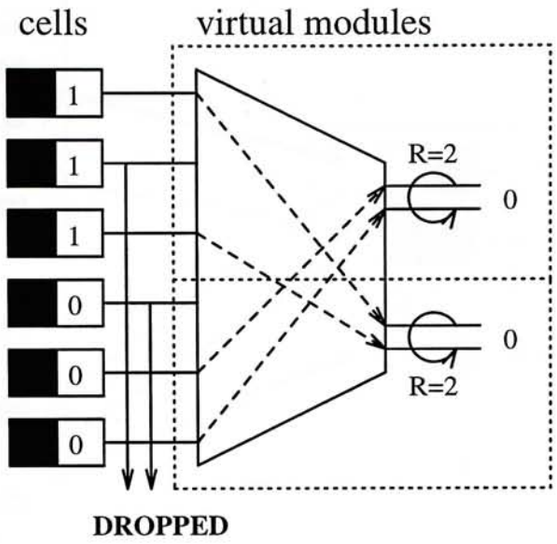

Figure 2.6: Limited Contenders With Partition

input module selects those cells with desired cluster number which matches the connection pattern of central stage.

2.3 The Capacity and Route Assignment of Virtual Paths

The capacity and route assignments of virtual paths follow the same procedure as that in the original path switch [34]. Capacity is assigned to each virtual path according to the aggregate traffic statistics from the input module to the cluster and their required QOS(Quality of Service). After capacity assignment, the number of central modules granted to virtual paths during a frame is known. the details will be postponed to chapter 4. There is an invisible border across the virtual paths which lead to the same physical module by logical partition. The capacity of a virtual path can not be used by others, even they are connected to one output module. However, the border can be broken if we need "wider" path

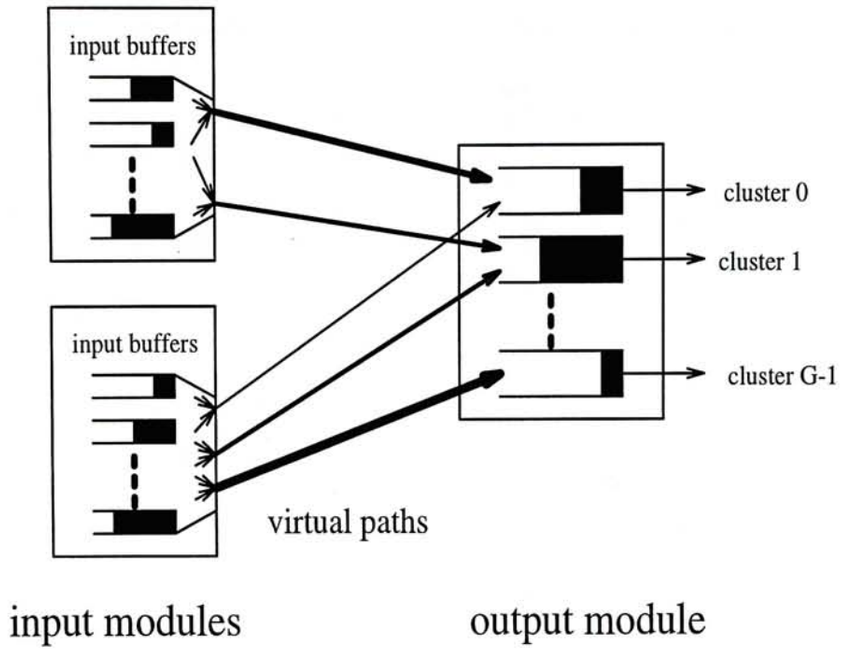

Figure 2.7: Partition of Virtual Paths

to carry traffic. In this sense, the management of virtual path is rather flexible.

The route assignment determines the connection patterns of central modules at each time slot. It is fulfilled by coloring a bipartite graph shown in figure 2.8. The bipartite graph needs not to be symmetric any longer, which relaxes the constraints imposed by physical parameters. In figure 2.8, there are KG small nodes on the right, each of them represents a cluster and G small nodes enclosed by a circle form an output module. Input modules are indicated by the nodes on the left, whose degree is FM, as mentioned before. Notice that there are FM/G adjacent edges on each cluster, if capacity of central stage is equally distributed among clusters. Once the coloring is complete, each central module will know the virtual paths which it is assigned to at any time slot and establish proper connections for the virtual paths.

However, in addition to the edge-coloring with FM colors, the scheduling of routes imposes extra constraint. Colors assigned to a cluster must be evenly

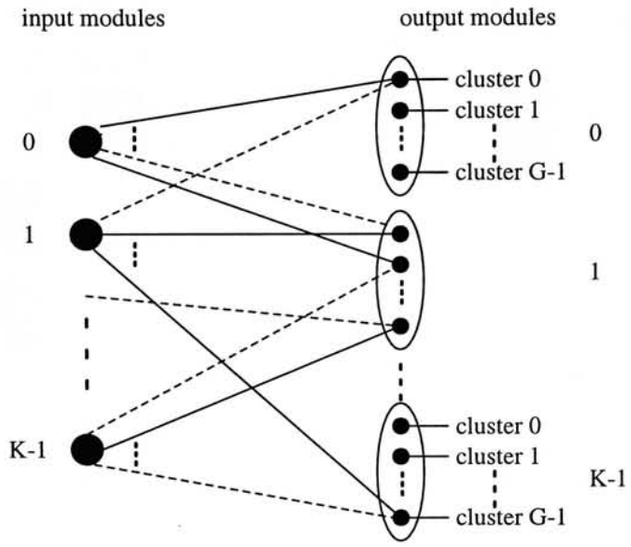

Figure 2.8: The Bipartite Graph of Route Assignment to Clusters

distributed over the F time slots, so that only M/G central modules are assigned to a cluster at any time. The FM colors can be divided into F sets, set $0 = \{0, 1, \ldots, M-1\}$, set $1 = \{M, M+1, \ldots, 2M-1\}$, ..., set $F-1 = \{(F-1)M, (F-1)M+1, \ldots, FM-1\}$. Set i denotes the colors corresponding to time slot i according to the time-space interleaving principle. M/G colors must be elaborately chosen from each set for a cluster so that there are M/G central modules assigned in a time slot and consequently FM/G during a frame.

A parallel algorithm, which does route assignment in a $N \times N$ Benes network [35], can be used for coloring bipartite graphs. Given input-output connections, the status of all the switching elements of Benes network can be calculated with time complexity of $O(\log^2 N)$. Since the connection patterns of M central modules of size $K \times K$ need to be determined, there are KFM connections during a frame. It will be shown in the follows that the connection patterns of M central modules will be determined by the route assignment in an $KFM \times KFM$ Benes network.

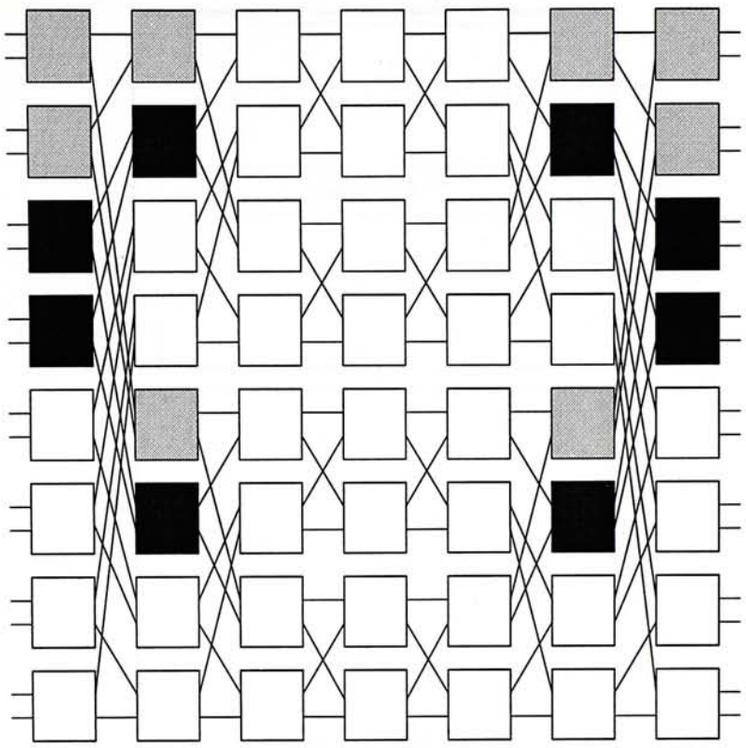

Figure 2.9: The Architecture of the Benes Network

A Benes network is constructed recursively with 2×2 switching elements as in figure 2.9. An $N \times N$ Benes network can be viewed as constructed by two $N/2 \times N/2$ Benes subnetworks, which can be further divided into four $N/4 \times N/4$ Benes subnetworks. So the input links at the first stage can be routed through two subnetworks, then 4 subnetworks at stage 2 and so on. Then the Benes network could be viewed as consisting of central subnetworks sandwiched by two out stages as shown in figure 2.10.

Property 1. *The Benes network is isomorphic to a Clos network, by rearranging some switching elements. Assuming that F, M, N, K are all power of 2, a $KFM \times KFM$ Benes network can be converted to a Clos network with FM central modules of size $K \times K$ and K input(output) modules of size $FM \times FM$* [35].

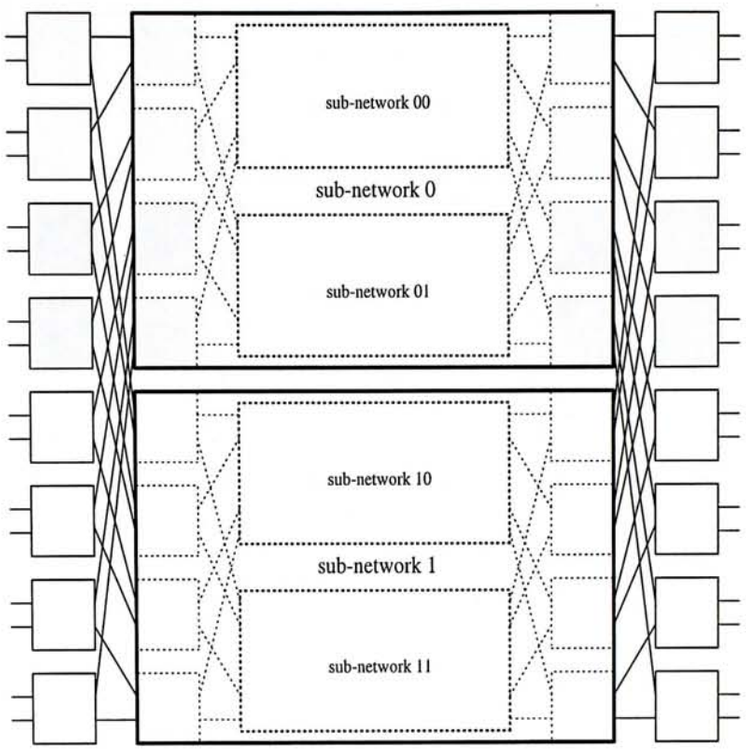

Figure 2.10: Divide a Benes Network into Sub-networks

If the switching elements of two filling patterns in figure 2.9 are inter-changed together with their connected links, a Clos network will be resulted in, as shown in figure 2.11. From top to bottom, the FM central modules are sequentially numbered from 0 to $FM-1$, the number of colors assigned to them. By the time-space interleaving principle, the top M ones denote color set 0 which represents central modules at time slot 0; the second M ones denote color set 1 which is time slot 1; and so on. The number of connections established from input to output modules is equal to the capacity of the virtual path. To calculate the routes of these connections, we only need to know the routing information of switching elements of the isomorphic Benes network. Given the KFM input-output connections, the status of each switching element of the Benes network in figure 2.9 can be determined by the parallel algorithm. By converting the

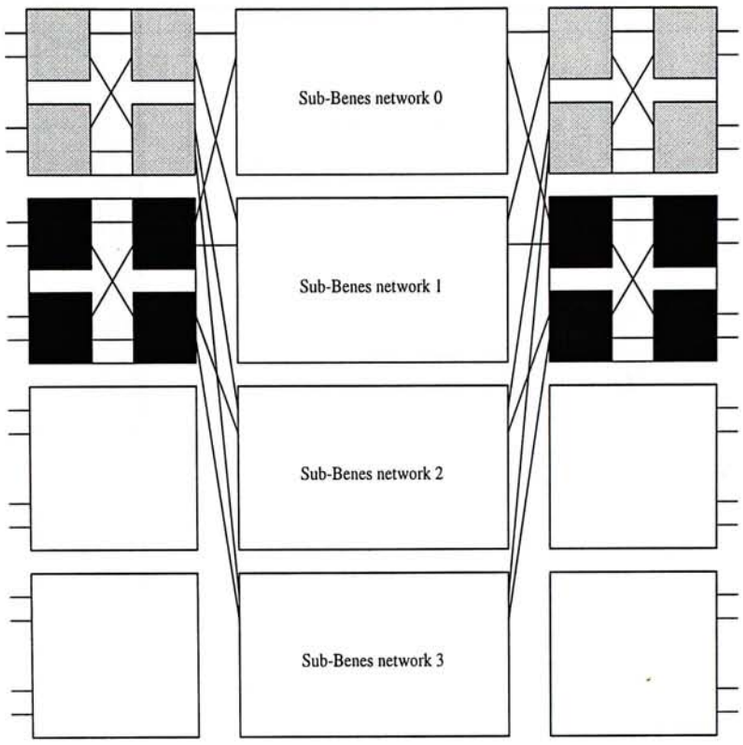

Figure 2.11: Convert A Benes Network into A Clos Network

Benes network into a Clos network in figure 2.11, the connection patterns of FM central modules can be obtained, which is actually the routing information of M central modules at every time slot. At this step, the edges of the bipartite graph can be colored.

To schedule the colors of a cluster over F time slots, another property of routing in Benes network is utilized.

Property 2. *Beginning with port zero, every F inputs(outputs) of one $FMK \times FMK$ Benes network are grouped into a "bunch" so that input(output) ports are partitioned into MK bunches. F Connections within a bunch will be scheduled in F different time slots.*

At any element of a Benes network, one of the two input links is routed through either of the two subnetworks at next stage. This is the constraint that

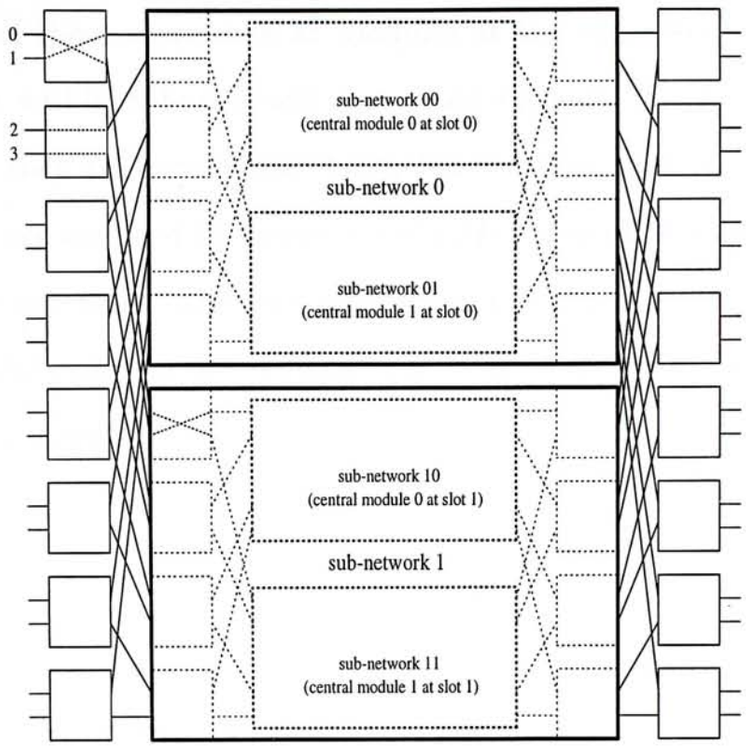

Figure 2.12: The Routing Constraint in Benes Network

routing must abide by. Due to the construction principle of Benes network, the F connections within a bunch must be routed through F subnetworks at stage $log_2 F$. For example, the first four input links in figure 2.12 are routed to 2 subnetworks first, with link 0, 2 to subnetwork 0 and link 1, 3 to subnetwork 1. At the second stage, subnetwork 0 is divided into two smaller subnetworks 00 and 01; link 0 is routed through subnetwork 00 and link 2 through subnetwork 01. The same is the routing of link 1 and 3. By property 1, the $KFM \times KFM$ Benes network can also be rearranged into a Clos network with F central modules of size $KM \times KM$ and KM input(output) modules of size $F \times F$, as is shown in figure 2.10. The F central modules are numbered with $0 \sim F-1$ from top to bottom, which can be viewed as the number of F color sets (time slot). One $KM \times KM$ central module can be further divided into M smaller $K \times K$

ones, so each of them represents M modules at the same time slot. Since the F connections within a bunch will be routed through the F distinct central modules, then they are evenly placed into F time slots, with one in each. When M/G bunches are assigned to a cluster, only M/G central modules are assigned to a cluster at any time. After the connection pattern of F central modules in figure 2.12 is determined, the routing of FM $K \times K$ central modules of path switch will be known.

Chapter 3

Performance Analysis and Simulation Results

Assuming uniform and independent traffic, the cell loss rate due to contention is calculated theoretically as that in [24]. The throughput with respect to various partition is hard to analyze theoretically, so the result is obtained by simulation. The maximum throughput with look-ahead scheme will be presented in section 2, which can be estimated by an approximate formula. In section 3, the input-smoothing scheme is assumed, which will improve the throughput. To reveal the cell loss probability at input buffer and output buffer, two mechanisms, queue loss and backpressure, are compared in section 5.

3.1 The Improvement of Concentration Loss

Under above assumption, cells on a virtual path have the same probability to be destined to each of N/G output ports within a cluster. Similar to equations 2.3,

the cell loss probability can be computed as follows.

$$\rho_{out} = \frac{N/G}{M/G} \cdot \rho_{in} = \frac{N}{M} \cdot \rho_{in} \tag{3.1}$$

$$P_{loss} = \frac{1}{\rho_{in}} \cdot \sum_{l=R+1}^{M/G} (l-R) \binom{M/G}{l} (\frac{\rho_{out}}{N/G})^l (1 - \frac{\rho_{out}}{N/G})^{M/G-l} \tag{3.2}$$

$$= \frac{1}{\rho_{in}} \cdot \sum_{l=R+1}^{M/G} (l-R) \binom{M/G}{l} (\frac{\rho_{in}}{M/G})^l (1 - \frac{\rho_{in}}{M/G})^{M/G-l} \tag{3.3}$$

It equals to the loss probability of an $M/G \times N/G$ module, since there are maximum of M/G concurrent contenders. If group size R is set equal to M/G, no cell would be knocked out. For a 1024×1024 switch, we choose $N = 32$ and $K = 32$. Given the offered load of 0.8, the cell loss probability with $R = 8$, $R = 9$ and $R = 10$ respectively, is shown in figure 3.1, figure 3.2 and figure 3.3 with respect to M/N and G, compared with that of no splitting(i.e. $G = 1$).

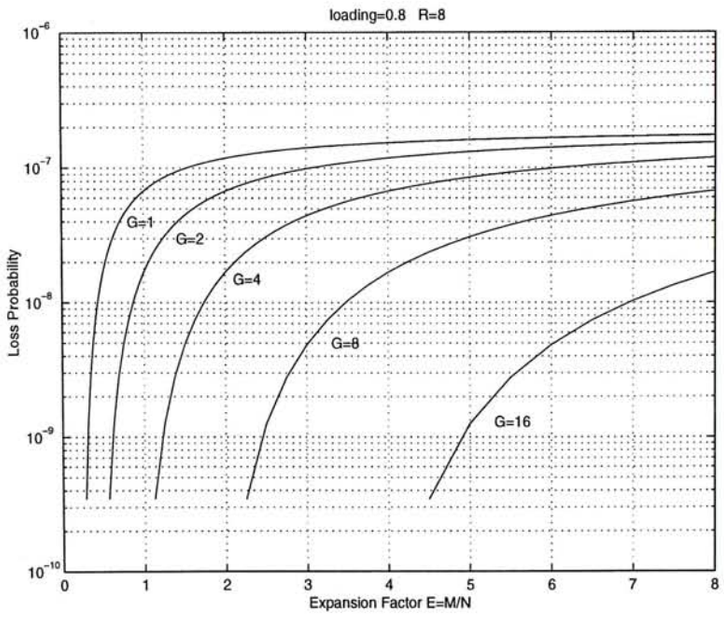

Figure 3.1: The Loss Probability vs. M/N for Various G with R=8

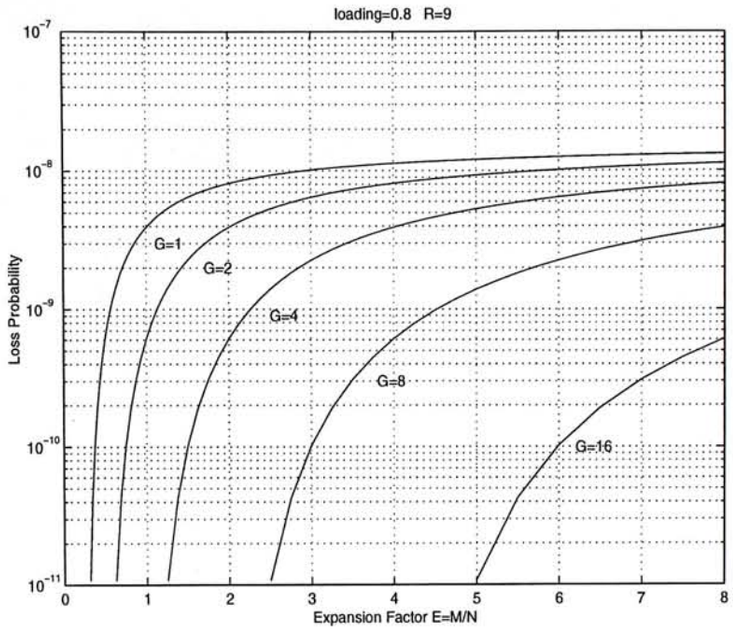

Figure 3.2: The Loss Probability vs. M/N for Various G with R=9

Given the required value of P_{loss}, several pairs of $(M/N, G)$ could be chosen. For instance, pairs of (1.2, 4), (2.4, 8), (4.9, 16) could achieve P_{loss} of 10^{-9} when $R = 8$. No cells loss would occur if any pair of (1, 4), (2, 8) and (4,16) is chosen, which is the case that M/G equals to R. It is shown that P_{loss} decreased sharply when M/G is close to R. However, the values of G and M/N also depend on the complexity of hardware and the throughput. The throughput would suffer from degradation when G is large for constant M/N, as illustrated in the follows.

3.2 The Throughput with Look-ahead Scheme

If the input modules are space-division switch, the throughput will suffer from degradation with partition of virtual path due to the head of line blocking. The degradation of throughput is due to the fact that the number of virtual paths of each input module is expanded by splitting. Desired cells are selected and

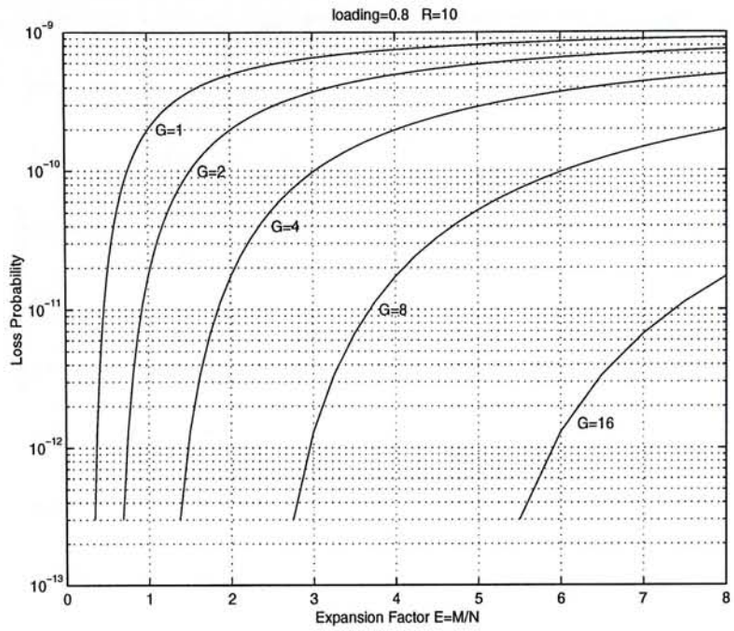

Figure 3.3: The Loss Probability vs. M/N for Various G with R=10

delivered according to their cluster number at the first stage. Under uniform traffic, cells at the first stage are destined for KG clusters, compared with K in the original path switching. As there are more destinations, less central modules could be dedicated to a virtual path on the average for constant M. This is equivalent to reducing the number of ports within a group in channel-grouping scheme. This limit of throughput is imposed by input-queueing.

The maximum throughput as a function of M/N is obtained by simulation when $w = 8$, which is denoted by circles in figure 3.4. From the simulation result, the empirical formula of the maximum throughput, ρ_{max} is estimated to be:

$$\rho_{max} = exp\{-\frac{0.288\sqrt{D}}{E} \cdot exp(-a(\frac{E}{\sqrt{D}-b})^2)\} \qquad for\ D > 2 \qquad (3.4)$$

where $D = KG/N$, $E = M/N$, and $a = 0.44$, $b = 1$ for $D = 4, 8$; $a = 0.3$, $b = 0$ for $D = 16$. The empirical formula is also plotted in figure 3.4, which

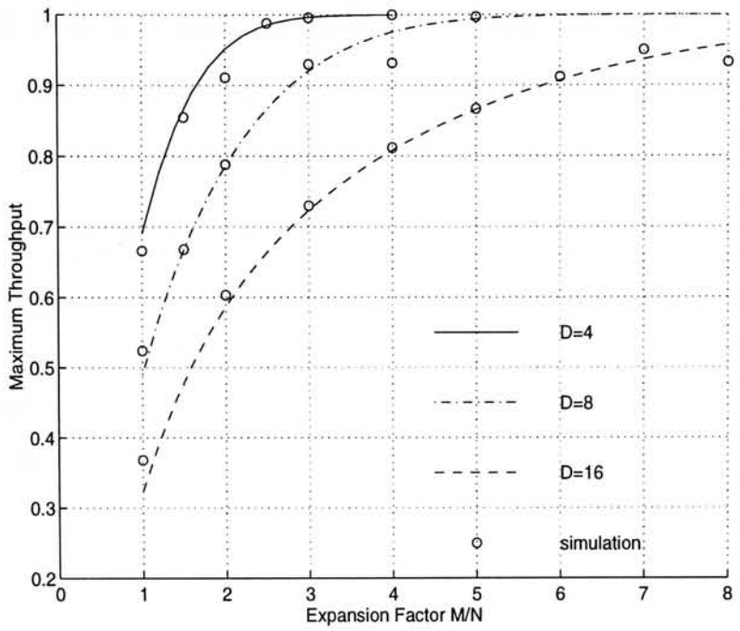

Figure 3.4: The Maximum Throughput vs. M/N for Various D

matches the simulation very well. It seems that the maximum throughput ρ_{max} is quite sensitive to the change of E and D, but insensitive to the change of a or b. This empirical formula can be used to investigate the the trade-off between performance and complexity of the switch, and to estimate the optimal design parameters for given throughput and cell loss probability requirements. We have observed the following from simulation results:

(1) The maximum throughput ρ_{max} surges up rapidly when M/N is less than $\frac{D}{2}$, but decreases with respect to D. We can't explain the noticeable discrepancy when $M/N = \frac{D}{2}$. It can be seen that for pairs of $(M/N, D)$ which achieve the same P_{loss}, those of larger M/N produce higher throughput. The reason is that the ascending rate of throughput with respect to M/N is greater than its descending rate with respect to D, so that large value of M/N and D could maintain high throughput.

(2) The throughput improvement is marginal when M/N is greater than $\frac{D}{2}$.

In addition, the switch is more complex with large value of M/N. Therefore, only moderate value of M/N and D should be chosen. Several ways to keep high throughput are presented below.

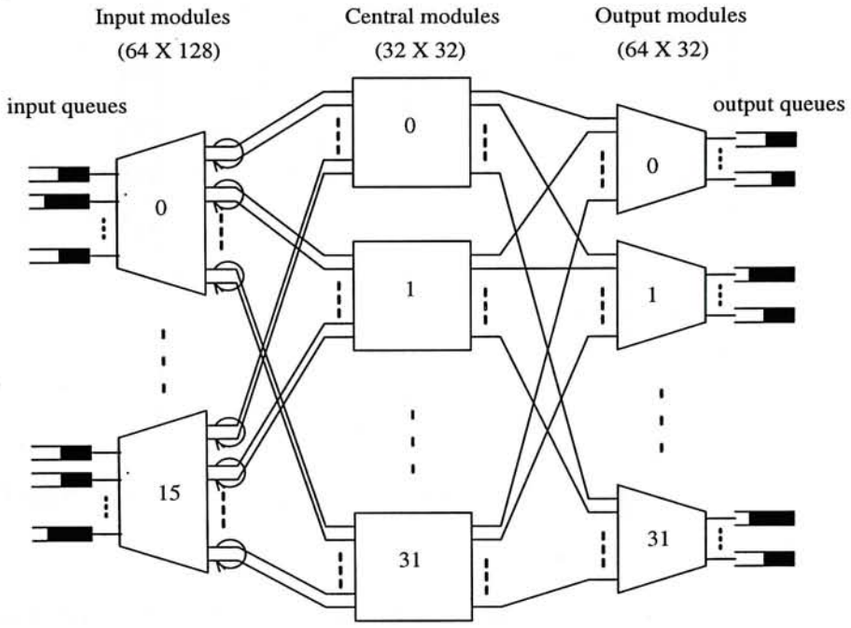

Figure 3.5: The Integration of Input Modules in Clos Network

For fixed M/N and G, larger N(i.e. smaller D) will result in both higher throughput and multiplexing gain [43]. It is due to two reasons: 1. the probability of finding a matched cell for an idle central link is higher if searching in more input ports; 2. More sources share the capacity of virtual path, which will improve the multiplexing gain. Thus, several input modules could be combined as a large one so that D is reduced, and consequently the throughput could be higher. For example, if $N = 32$ and $K = 32$ are chosen to construct a 1024×1024 switch, the maximum throughput is 81% when $M/N = 2$, $G = 8(D = 8)$. If every two 32×64 input modules are integrated into one 64×128 module with G kept unchanged, as in figure 3.5, then D is only 4 so that the throughput could reach 91% without cell loss. However, the size of module is constrained by VLSI

technique and look-ahead scheme must be performed faster for a large module.

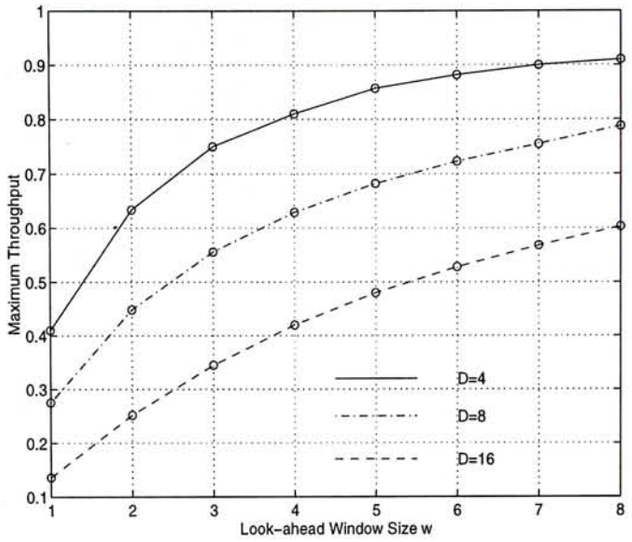

Figure 3.6: The Maximum Throughput vs. w for Various D

Increasing the look-ahead window size w would also be helpful, especially when D is large. Results of simulation with $M/N = 2$ are shown in figure 3.6.

3.3 The Throughput with Input Smoothing Scheme

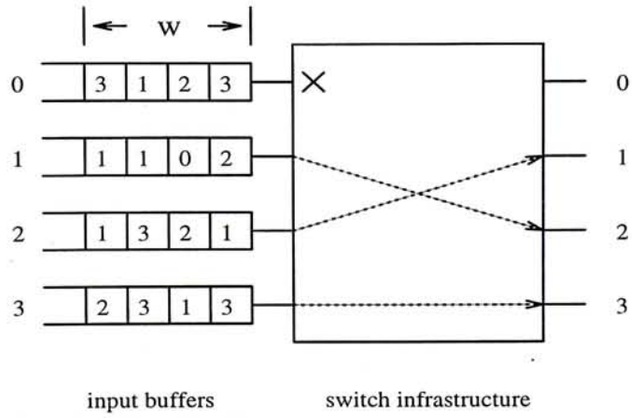

Figure 3.7: Idle Output Port with Look-ahead Scheme

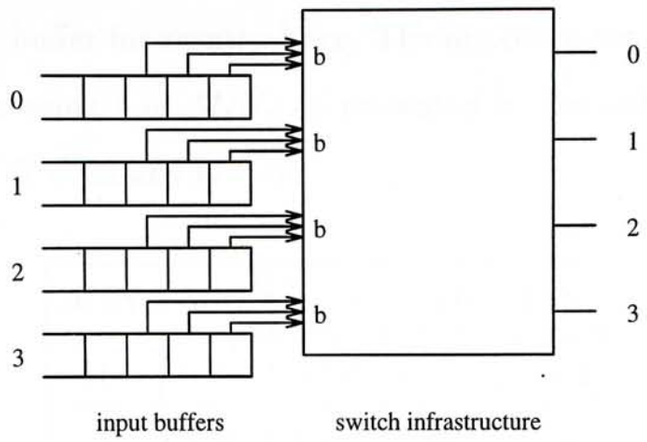

Figure 3.8: The Input Smoothing Scheme

It is proposed in look-ahead scheme that an input port can send at most one cell during a contention resolution phase, so maximum of N cells can be cleared from an input module and $M - N$ central links will be idle at any time. Besides, after all the losing input ports are searched to depth of w, it is possible that cells destined to some remaining idle output ports can not be found at those losing input ports within the whole window. Thus the bandwidth of those output ports will not be used even if there are matched cells at those winning input ports. For instance, in figure 3.7, input port 1 is selected to transmit the first cell to output port 2 during the contention resolution phase, so it can not send any more cells at current time slot. At the end of contention resolution phase, output port 0 is still idle, because there is no cell to output port 0 at all the other input ports apart from port 1. Even the second cell at input port 1 is destined to output port 0, the output port 0 has to be idle. If an input port is allowed to transmit more than one cell, the throughput of input queueing switch will be improved. This idea, called input smoothing, was proposed in [17]. As shown in figure 3.8, cells at an input port are demultiplexed into b input links and Nb cells simultaneously contend for output ports. Those losing contention

will wait at input buffer for reentry later. The improvement of the throughput is marked by increasing b or M/N, as presented in the following table when choosing $K = 32$, $N = 32$ and $G = 8$:

M/N	b=2	b=4	b=6	b=8
1	0.317	0.486	0.589	0.655
2	0.507	0.850	0.987	0.996
3	0.636	0.994	0.999	0.999

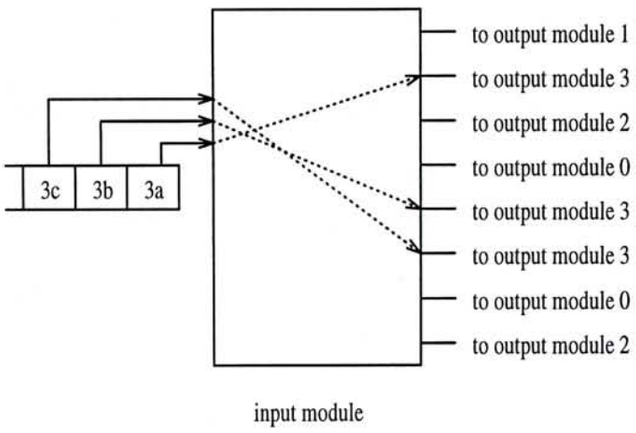

Figure 3.9: The Resequencing of Cells

Of course, the complexity of input module with input-smoothing is much higher compared with look-ahead scheme. If it is constructed by batcher-banyan modular architecture, the complexity is discussed in [33] and [34]. Another problem of resequencing appears if multiple cells can be sent from an input port simultaneously. It is possible that more than one cells of a virtual circuit at an input port can be sent to central modules and further arrive at their output port at the same time. To maintain the sequence of cells of a virtual circuit, the first one is always routed to the upmost central module available, and the successors

are routed to lower central modules sequentially. So the central modules keep the sequence information. Suppose that three central modules are assigned to output module 3 at a time slot, as shown in figure 3.9. The three cells, 3a, 3b and 3c of a connection are stored at the input FIFO buffer. If all of them win the access to the central modules, 3a is always routed to the upmost central module, 3b the second and 3c the lowest. When they arrive at the output port, the output port can identify the one appearing at upper central module as the predecessor and the one at lower central module as the successor. Then they will be stored at the output buffer with their original sequence.

3.4 The Throughput with Bursty Source

For independent traffic such as Bernoulli source, the throughput can be improved by look-ahead or input smoothing. However, this may not be true for correlated source. Suppose that cells arrive at input port in bursts; within each burst, cells are destined to the same output port. If these bursty arrivals are stored in FIFO buffer, it is highly possible that several consecutive ones have the same destination. If the first one of them fails in the contention, it is most likely the rest of them can not be selected to transmit so that the look-ahead or input smoothing is almost useless [40]. The bursty characteristics of traffic will degrade throughput a lot, which depends on the burst length [43].

An approach to maintain high throughput of input queueing was proposed in [27], [45] etc. It is suggested that each FIFO queue in input link is divided into logical queues according to their destinations and a matching procedure is performed at input buffer to select winning cells. However, the number of logical

queues at an input port has to equal to the number of output ports. For a large-scale switch, this central controller may be a bottleneck because the number of logical queues at an input port is vast. In path switch, cells are accommodated in logical queues according to their clusters, thus the number of logical queues is reduced and the above scheme may be feasible. The head of line blocking could be completely eliminated if each input keeps $D = KG$ logical queues destined for D addresses. Or a shared-buffering memory switch could do the same job. Almost 100% throughput could be achieved with a moderate value of M even with bursty traffic.

3.5 Buffer Dimensioning and The Cell Loss Probability Due to Buffer Overflow

Owing to the finite input and output buffer size, arrivals at the first and last stage seeing a full buffer will be lost. Intuitively, more cells will be lost at input buffer if the throughput of switch is low; otherwise, most of the loss will take place at output buffer. The input queueing and output queueing switch are the extreme cases where buffer overflows at either input or output. The path switch is input-output buffered, which requires a proper buffer allocation to input and output. We assume uniform and independent Bernoulli process at input buffer so that the arrivals at output port form Poisson process [17]. Each input and output port is dedicated a buffer and there is no sharing. To evaluate the cell loss probability, two different switching mechanisms can be considered:

- Queue Loss(QL), in which cell loss takes place at both the input and output buffer. The first stage can switch cells to the last stage whenever the bandwidth is available, without the knowledge of buffer space at output port. At both input and output buffer, excess cells that can not be accommodated will be dropped.

- Backpressure(BP), in which the available space of output buffers is signaled back to input ports. The number of cells transferred to output port can not exceed the available vacancy, even if there is idle bandwidth. In this case, cells that can not be switched are stored at input buffer unless it is full, and no cell loss occurs at output buffer.

Since the switch capacity is expensive, we should protect cells which have passed through switch from being collapsed. If some cells have to be dropped due to heavy traffic and limited storage space, we should choose to drop those less important, or abandon them at input buffer, instead of losing at output buffer. The switch capacity is wasted if a number of cells are lost at output buffer after they are switched to output port. Backpressure could keep cell from loss at output buffer, at the cost of lower throughput [21] [50]. The throughput will be lower than that under QL mode because cells may be blocked by a full output buffer, and consequently idle bandwidth will be resulted in. In path switch, however, the capacity of central stage is shared by all the traffic on the virtual path. If one output buffer is full, cells destined to other output ports on the virtual path could be selected to transmit. Then we can expect that the throughput with backpressure will not degrade much.

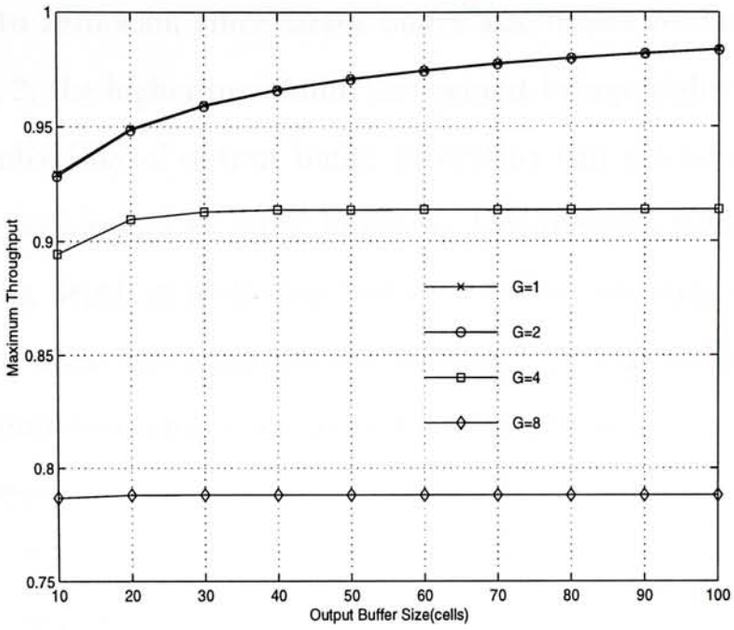

Figure 3.10: The Maximum Throughput under Backpressure Mode

We are interested in evaluating the maximum throughput and cell loss probability due to buffer overflow under these two modes. Each input port is dedicated a buffer of size B_i and output port B_o. With $K = 32, N = 32, M = 64, R = 8, w = 8$ and look-ahead scheme, the maximum throughput under QL mode has been shown in figure 3.4. The throughput with $G = 1, 2$ is very close to 100%, which is not shown in the figure. The maximum throughput under BP mode as the function of output buffer size is obtained by simulation, shown in figure 3.10 with the same parameters. The throughput with $G = 4, 8$ under BP mode is almost identical to that under QL mode, but there is a noticeable gap when $G = 1, 2$. Because the maximum throughput with $G = 8$ is close to 80% under QL mode, the chance of output buffer saturation is negligible at this load when buffer size is greater than 10. In this case, the backpressure signal seldom occurs so that the first stage will not be affected. When $G = 4$, the backpressure can not be neglected if buffer size is around 10, so the throughput is a bit lower. But

the gap tends to zero soon since larger buffer size makes backpressure ineffective. As $G = 1, 2$, the higher maximum throughput brings higher load at output buffer. The probability of output buffer saturation can not be neglected, then more cells will be blocked at input buffer by backpressure signal. Due to the limited searching depth of look-ahead scheme, other cells may not be selected to transfer even when idle bandwidth is available. As the buffer increases, the throughput is improved and tends to that under QL mode since a large buffer makes BP useless.

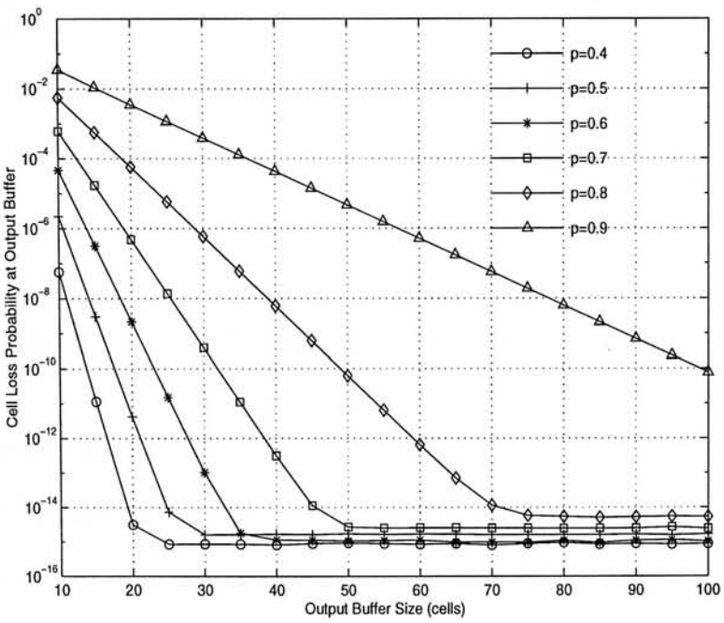

Figure 3.11: The Cell Loss Probability at Output Buffer

The loss probability at output buffer under QL mode, ϵ_o, could be evaluated by the M/D/1/N model [17](see appendix of this chapter) and is plotted in figure 3.11 with various traffic loading at the last stage. The loss probability at input buffer, ϵ_{iql}, is obtained by simulation. Due to the limited random numbers that computer can generate, the accurate value of loss probability is above 10^{-7}. The total loss probability in the switch, ϵ_{ql}, is equal to $\epsilon_{iql} + (1 - \epsilon_{iql})\epsilon_o \approx \epsilon_{iql} + \epsilon_o$.

Since there is no loss at output buffer under BP mode, the cell loss probability, ϵ_{bp}, is equal to the loss at input buffer denoted by ϵ_{ibp}, which is also got by simulation.

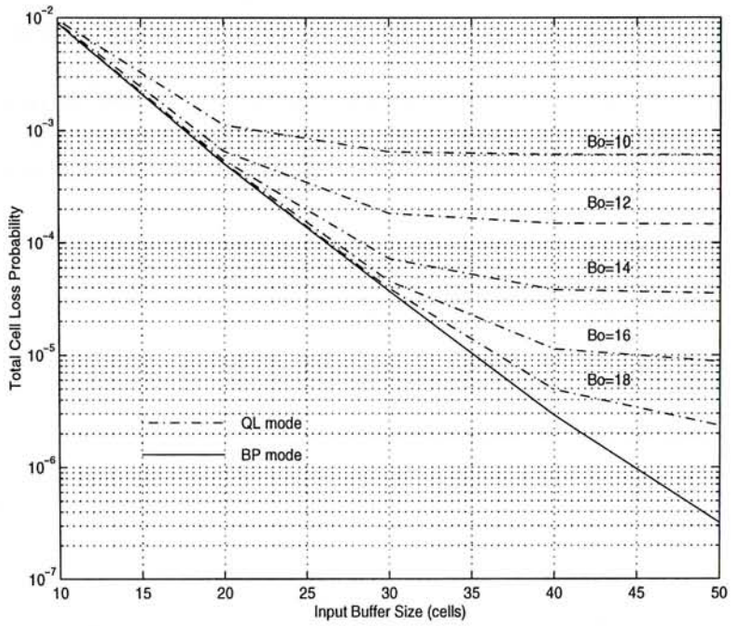

Figure 3.12: The Total Cell Loss Probability vs. Input Buffer Size

First we choose $K = 32, N = 32, M = 64, R = 8, w = 8$ and $G = 8$. Because the maximum throughput under BP mode is not affected by backpressure as mentioned above, the cell loss at input buffer is also identical to that under QL mode. Given input load of 0.7, the total cell loss probability under two modes as a function of input buffer size is plotted in figure 3.12. ϵ_{bp}, the solid line, is obtained with B_o=10, since larger B_o is ineffective. Under BP mode, it is effective to place a large buffer at input and ϵ_{bp} drops quickly with B_i if B_o is greater than 10. However, Under QL mode, ϵ_{ql} gets saturated soon when B_o is small. It is of no use to allocate a large buffer to input since the cell loss is dominated by ϵ_o. It can be observed that ϵ_{bp} is always lower than ϵ_{ql} given identical buffer budget, because no output loss occurs. For the same loss

requirement, the BP mode needs less buffer than QL mode. For instance, total buffer size of 45($B_o = 10$, $B_i = 35$) is needed under BP mode, given loss rate of 10^{-5}; but under QL mode, total of 55($B_o = 18$, $B_i = 37$) is required.

The reason is illustrated in [40] that there are some buffer-sharing effects in the input-buffered scheme. Under QL mode with high throughput, arrivals from all inputs to an output port can easily congest the dedicated buffer, so most of losses occur at output buffer. Under BP mode, however, the arrivals are distributed across several input buffers if output buffer gets congested. Therefore, an input buffer does not overflow easily. It is consistent with the intuitive explanation that the QL mode generally requires a larger output buffer to guarantee a certain loss performance.

Now we choose $K = 32, N = 32, M = 64, R = 8, w = 8$ and $G = 2$, so that the maximum throughput will change with the output buffer size B_o under backpressure. In this case, the throughput under QL mode is very lose to 100%, so the performance is like that of output queueing switch. Given the input load of 0.9, ϵ_{iql} is less than 10^{-9} if input buffer size is larger than 10. Compared with ϵ_o, ϵ_{iql} could be neglected in the range interested so that we could consider that ϵ_{ql} does not change with input buffer size. However, the story is different for BP mode, where the throughput will degrade and input buffer plays an important role in loss performance. The total cell loss probability with respect to output buffer size are compared in figure 3.13. ϵ_{ql} , the dotted-dash line, is obtained with B_i=10.

It shows that cell loss under both modes decreases with B_o, but ϵ_{bp} drops more sharply. With the same total buffer size, the BP mode does not always performs better unless buffer is properly allocated among input and output. For cell

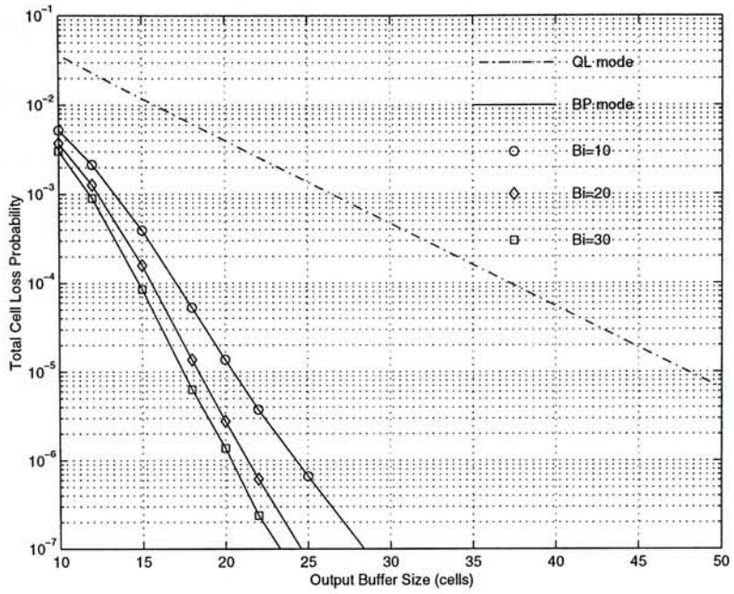

Figure 3.13: The Total Cell Loss Probability vs. Output Buffer Size

loss requirement of 10^{-3}, buffer size of 42 under BP mode is needed if B_i=30. Whereas, only 37 is required under QL mode(B_i = 10, B_o = 27). With this buffer setting under BP mode, the improvement of ϵ_{bp} by input buffer-sharing can not compensate the negative effect of lower throughput. Consequently, the QL mode with a large output buffer will performs better. However, if fixing B_i to be 10, the BP mode always performs better. Although the throughput is lower under BP mode, the effect of buffer-sharing with this setting overruns the negative effect of longer service time. Then the proper buffer allocation among input and output is important under BP mode. It is noticed that the performance under BP mode is improved much more quickly by increasing output buffer size than placing the same amount at input buffer. For instance, the loss rate is about 3.7×10^{-3} with B_i of 20 and B_o of 10; however, it drops sharply to around 1.4×10^{-5} with the same buffer budget but 10 at input and 20 at output. Though a larger input buffer would reduce the cell loss, the improvement of the

maximum throughput offsets the effect of larger input buffer if the same amount is placed at output. Then we can conclude that in this case, more budget should be given to output buffer if the amount of memory is kept constant.

Appendix: The Loss Probability at Output Buffer

Let the load at an output be p, then the probability that a cell to an tagged output port appears at a central link is $\frac{p}{M/G}$, because maximum of M/G central links can carry cells to an output port. The arrival process at output can be viewed as Poisson process with parameter p, if M/G is large enough [17]. Defining the random variable A as the number of arrivals at an output port in a time slot, we have:

$$a_k = Pr\{A = k\} = \binom{M/G}{k}(\frac{p}{M/G})^k(1 - \frac{p}{M/G})^{M/G-k} k = 0, 1, \ldots, R-1 \quad (3.5)$$

$$a_R = Pr\{A = R\} = \sum_{k=R}^{M/G} \binom{M/G}{k}(\frac{p}{M/G})^k(1 - \frac{p}{M/G})^{M/G-k} \quad (3.6)$$

where R is the output group size. Remember that maximum of R cells can be read into the buffer at a time slot and excess ones are knocked out.

Let Q_m be the output queue length at the end of the mth time slot, and A_m denote the arrivals during the time slot. Since there are at most R batch arrivals at an output port, the queue length could be modeled by a M/D/1/N queue with output buffer size B_o:

$$Q_m = min[max(0, Q_{m-1} - 1) + A_m, B_o] \quad (3.7)$$

This is a finite, discrete-time Markov chain with state transition matrix P

$$P = \begin{pmatrix} a_0 & a_1 & \cdots & a_R & 0 & \cdots & 0 & 0 & 0 \\ a_0 & a_1 & \cdots & a_R & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_0 & \cdots & a_{R-1} & a_R & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & a_0 & a_1 & 1 - \sum_{i=0}^{1} a_i \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & a_0 & 1 - a_0 \end{pmatrix}$$

Denote the steady state probability by $q_n = Pr\{Q = n\}$ and in vector form $q = (q_0, q_1, \ldots, q_{B_o})$, where Q is the queue length in steady state,. It can be solved by the balance equation $q(I - Q) = 0$ and $qe = 1$. The cell loss probability at output buffer, ϵ_o, can be calculated by following equation

$$\epsilon_o = \frac{E[number\ of\ lost\ cells\ at\ output\ buffer]}{offered\ load\ at\ output} \tag{3.8}$$

$$= \frac{1}{p} \sum_{n=B_o-R+1}^{B_o} \sum_{k=B_o-n+1}^{R} q_n a_k (k + n - B_o) \tag{3.9}$$

Chapter 4

Capacity Assignment and Evaluation of Multiplexing Gain

4.1 Principle of Capacity Assignment

Capacity assignment of ATM services has been discussed in [13] [16] [18] [19] [20] [22] [37] [52] etc, based on prediction or measurement of traffic. If traffic characteristics can be modeled accurately, the prediction method can be used to do the capacity assignment. The ATM network has to support multiple classes of services such as data, voice and video with widely different traffic characteristics and QOS requirements. The users must declare their traffic parameters and the expected QOS, which can be mapped to appropriate service classes. The ATM services are classified into five classes , including CBR(Constant Bit Rate), real-time VBR(Variable Bit Rate), non real-time VBR, ABR(Available Bit Rate), and UBR(Unspecific Bit Rate) service. The traffic parameters are specified by PCR(peak cell rate), SCR(sustainable cell rate), MBS(maximum

burst size), MCR(minimum cell rate), and CDVT(cell delay variation tolerance), which are detailed in the Traffic Management Specification of ATM Forum [2]. The QOS parameters of interests are 1.CDV(peak-to-peak cell delay variation) and CTD(maximum cell transfer delay) 2. CLR(cell loss ratio), defined by ATM Forum. The attributes of service classes in terms of traffic parameters and QOS are presented in the following table, which characterize a class completely.

service class	traffic parameters	QOS
CBR	PCR, and CDVT	CLR, CTD and CDV
real-time VBR	PCR, SCR MBS, and CDVT	CLR, CTD and CDV
non real-time VBR	PCR, SCR MBS, and CDVT	CLR
ABR	PCR, CDVT, and MCR	Minimum Cell Rate
UBR	PCR, CDVT, and MCR	non

At call setup, necessary resources including capacity and buffer, must be reserved along the path. If enough resource could be reserved at all the nodes along the path, the connection can be admitted to the network, otherwise it is rejected. So it is very important to determine the required capacity and buffer at each node to guarantee the promised QOS while maximizing the resource utilization. Most of ATM services are bursty, such as video streams and TCP/IP datagram which generate large amount traffic in a short time, but little or no at the rest of duration. Since traffic arrive in bursts, the amount of bandwidth required by these services varies with time during the connection. One of the simplest way to manage all the traffic streams is based on peak allocation. A

bandwidth equivalent to the declared peak bit rate is allocated to each connection to ensure the high deterministic QOS. However, under this policy, most of the bandwidth will be significantly under-utilized because of the large variance in the traffic streams. In this case, no multiplexing gain is taken advantage and the resource of network is wasted.

Another way is to allocate capacity based on the effective bandwidth, which is the minimum capacity required to guarantee the QOS when a source is served alone. For instance, the effective bandwidth of an on-off source can be calculated once its traffic descriptors, QOS and available buffer size are known, as detailed in [12] [16]. The allocated bandwidth of aggregate traffic is equal to the sum of effective bandwidth of all the multiplexed sources. It is still conservative because the multiplexing gain among sources is not taken into account.

A more efficient approach to manage network resource is to allocate bandwidth to aggregate bursty traffic with close characteristics and QOS requirements. In this case, the capacity allocated to a group of bursty traffic streams is lower than the sum of their effective bandwidth. For simplicity, we assume that connections within a service class possess identical traffic attributes, so that capacity is assigned to the aggregate traffic of the class. The QOS is guaranteed for the class instead of individual connections, but each connection can also get satisfactory QOS if proper scheduling scheme is applied at cell level [63].

4.2 The Model of Virtual Path

If the input modules are shared-buffering memory switch, the throughput will be as high as 100%, since there is no head of line blocking. As depicted in figure 4.1,

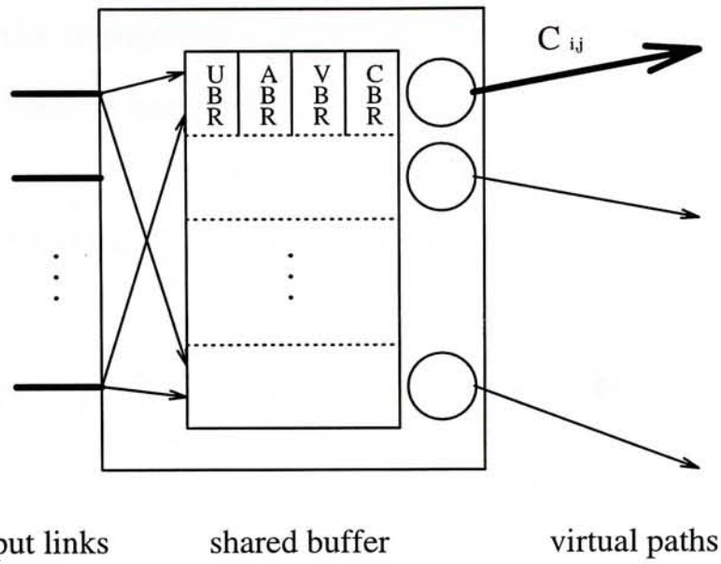

Figure 4.1: The Model of Virtual Path

there are KG logical queues in one input module, each of them accommodating cells destined to a cluster. Arrivals at all the input ports are multiplexed subject to their destination in individual logical queues served by allocated rate. With work-conserving service discipline, a virtual path will make use of its capacity whenever there is backlog in its buffer. To make analysis simple, each logical queue is viewed as a dedicated buffer, which is further partitioned based on classes. No buffer sharing is assumed among virtual path or classes. The capacity assignment is not trivial, because the multiplexing must be taken into account. Since the characteristics and QOS of service classes are different, the capacity of virtual path will be calculated based on each class. Supposing that there are n_k connections of class k(CBR, VBR, ABR and UBR) feeding into a logical queue with service rate C_k for each class, the capacity of virtual path is equal to the sum of C_k. Despite that the capacity of virtual path is logically partitioned among classes, it can be used by other sources if a certain class does not make use of its allocated bandwidth with a work-conserving scheduling scheme at cell

level. Although the throughput will not degrade for shared-buffering memory switch, the multiplexing gain still changes with the "width" of virtual paths. In successing sections, the capacity assignment based on services classes will be presented and the multiplexing gain will be estimated.

4.3 Capacity Assignment for CBR Service

CBR services are characterized by PCR with constant interarrival time. It is common to assume the aggregate CBR traffic should be allocated equal to the sum of their peak rate. However, if the CTD and CLR requirements are stringent, the required bandwidth will be greater than the overall peak rate [53]. The waiting time distribution of the aggregate CBR connections can be analyzed via the $\sum D_i/D/1$ queue. To avoid complex root-finding computation, Dron, Ramamurthy and Sengupta developed an efficient approximation for the waiting time distribution, which significantly reduces the computation time while being very accurate [11].

Suppose there are n CBR connections with peak rate R bits/sec, multiplexed on a virtual path with service rate C_{cbr} bits/sec. From [11], the probability that the waiting time of an arrival cell is greater than t can be calculated as follows:

$$Prob\{waiting\ time\ of\ an\ arrival > t\} = exp\{-2[\frac{t^2{C_{cbr}}^2}{nb^2} + \frac{(C_{cbr} - nR)t}{b}]\} \tag{4.1}$$

Where b is the number of bits in an ATM cell(424 bits).

In order to guarantee $Prob\{CTD > t_{max}\} < \epsilon_2$, it is sufficient to ensure $Prob\{waiting\ time\ of\ an\ arrival > t_{max}\} < \epsilon_2$, which can be computed by

equation 4.1. Given ϵ_2 and t_{max}, the required capacity C_{cbr}^{delay} to meet the delay constraint can be calculated by equation 4.1 as

$$C_{cbr}^{delay} = \frac{2n(\alpha b/t_{max} + nR)}{n + \sqrt{n^2(1 + 4Rt_{max}/b) + 4\alpha n}} \tag{4.2}$$

where $\alpha = -1.15 \log_{10} \epsilon_2$

To compute the cell loss ratio, assume buffer of size B_{cbr} is allocated to all the n CBR connections. Arrivals seeing a full buffer will be lost. Approximating the cell loss ratio by the buffer overflow probability, we can use the relation between the queue length seen by arrival and the waiting time experienced. When there is backlog in the buffer, the service rate must be C_{cbr}, so it will take time of B_{cbr}/C_{cbr} to drain out a full buffer.

$$queue\ length\ seen\ by\ arrival \approx waiting\ time\ experienced * C_{cbr} \tag{4.3}$$

$$\begin{aligned} CLR &\approx Prob\{queue\ length\ on\ arrival > B_{cbr}\} && (4.4)\\ &= Prob\{waiting\ time\ experienced\ by\ arrival > \frac{B_{cbr}}{C_{cbr}}\} && (4.5)\\ &= exp\{-2[\frac{{B_{cbr}}^2}{n} + (1 - \frac{nR}{C_{cbr}})B_{cbr}]\} && (4.6) \end{aligned}$$

Given cell loss rate ϵ_1 and buffer size B_{cbr}, from equation 4.6, the bandwidth required to meet the cell loss constraint is:

$$C_{cbr}^{loss} = \frac{nR}{1 + B_{cbr}/n - \beta/B_{cbr}} \tag{4.7}$$

where $\beta = -1.15 \log_{10} \epsilon_1$

From the above analysis, the capacity to support n CBR connections should guarantee both the CTD and CLR, so it is

$$C_{cbr} = max\{nR, C_{cbr}^{delay}, C_{cbr}^{loss}\} \tag{4.8}$$

4.4 Capacity Assignment for Real-time VBR Service

For real-time VBR sources, the QOS is specified by the Cell Loss Ratio(CLR), Maximum Cell Transfer Delay(MaxCTD) and the Cell Delay Variation(CDV). Allocating a large buffer to a real time source is of no significance, since the delay constraints are tight and cells experienced queueing delay beyond the constraints will be discarded. Only the case of small buffer for real-time source will be considered. Most of real-time VBR sources are very bursty, such as video stream which produces traffic from time to time but not constantly. Thus the real-time traffic could be modeled by exponential on-off source very well, which alternates between "on" and "off" states. When it is on, traffic stream of peak rate R is generated; when it is off, no traffic is generated. The period of on state as well as off state is exponentially distributed with average length of $1/\mu$ and $1/\lambda$ respectively. The utilization of a source, ρ, i.e. the probability that a source is on, is equal to $\lambda/(\lambda+\mu)$; and the average traffic rate of a source $r = R\lambda/(\lambda+\mu)$.

Assume n identical real-time VBR sources are multiplexed on a virtual path with service rate C_{rtvbr}. The buffer size allocated to each source is B_s, and the capacity per source is denoted by $C_s = C_{rtvbr}/n$. The buffer overflow probability

can be estimated by following equation [18] [20]:

$$Prob\{queuelength > nB_S\} = \frac{R}{\sqrt{2\pi n C_s(R-C_s)} ln \frac{C_s(1-\rho)}{\rho(R-C_s)}} exp\{-n[I(C_s)+H\sqrt{B_s}]\} \tag{4.9}$$

$$I(C_s) = \frac{C_s}{R} ln \frac{C_s/R}{\rho} + (1 - \frac{C_s}{R}) ln \frac{1-C_s/R}{1-\rho} \tag{4.10}$$

$$H = \sqrt{\frac{1}{R}\{\lambda(1-\frac{C_s}{R}) + \mu\frac{C_s}{R} ln \frac{\mu C_s}{\lambda(R-C_s)} - 2[\mu\frac{C_s}{R} - \lambda(1-\frac{C_s}{R})]\}} \tag{4.11}$$

where $1 < C_s < R$

It is noticed that the buffer overflow probability is not improved much by the presence of a small buffer compared to the average burst size. For example, if the peak rate of each source is 1Mbit/s and the average on period is 2 seconds, then the average burst size is 2Mbits. A buffer of 1000 bits is small and negligible to improve the buffer overflow. So the buffer overflow probability can always be approximated by a zero buffer [20].

By approximating cell loss ratio with buffer overflow probability ϵ_1, the capacity per source C_s^{loss} can be calculated from equation 4.9 numerically, which is not time consuming.

Given the delay constraint $Prob\{CTD > t_{max}\} < \epsilon_2$, the required capacity C_s^{delay} could be approximated by the relation between queue length seen by arrival and the waiting time in the preceding section.

$$\epsilon_2 = Prob\{waiting\ time\ experienced\ by\ arrival > t_{max}\} \tag{4.12}$$

$$\approx \quad Prob\{queue\ length\ on\ arrival > nC_s t_{max}\} \tag{4.13}$$

$$= \quad \frac{R}{\sqrt{2\pi n C_s (R - C_s) ln \frac{C_s(1-\rho)}{\rho(R-C_s)}}} exp\{-n[I(C_s) + H\sqrt{C_s t_{max}}\ ]\} \tag{4.14}$$

After above steps, the capacity for real-time VBR connections should be the maximum of nC_s^{delay} and nC_s^{loss}.

4.5 Capacity Assignment for Non Real-time VBR Service

Since there is no time constraint on non real-time VBR services, a large buffer can be used to store bursts that arrive faster than that can be transmitted. The accurate analysis was done by anick etc [1]; however, it involves rather complex root-finding solutions. Two computationally simple approximation of bandwidth allocation for single or aggregate connections are proposed. One is the effective bandwidth or equivalent capacity based on fluid-flow model [12] [16]; the other is Gaussian approximation [16]. The former calculates the effective bandwidth of a single source based on the flow-fluid model with bit rate modulated by the state of an underlying Markov chain. The sum of effective bandwidth is then assigned to the aggregate traffic subject to the linearity and additivity, as revealed in [6] [12] [28]. The effective bandwidth is considered to be conservative since the multiplexing is ignored. However, in the case of large buffer, high utilization of source, or small number of multiplexed connections, the multiplexing could be negligible, as stated in [16] and verified in the next section where the multiplexing gain is evaluated. When a lot of sources are aggregated or the utilization is low, the Gaussian approximation would perform better, because the multiplexing is

no longer negligible.

The model assumed here is the same as that of real-time VBR source. The effective bandwidth of a single source is approximated in [16] as:

$$C_e = \frac{R}{2} + \frac{1}{2\alpha\beta(1-\rho)}\{\sqrt{[\alpha\beta(1-\rho)R - B_s]^2 + 4B_s\alpha\beta\rho(1-\rho)R} - B_s\} \quad (4.15)$$

where $\alpha = ln(1/\epsilon)$, $\epsilon = Prob\{buffer \ \ overflow\}$ and $\beta = 1/\mu$, the average burst length. The capacity required by n multiplexed connections is the sum of their effective bandwidth.

When statistical multiplexing is of significance, the Gaussian approximation would be rather accurate by assuming a Gaussian distribution for the aggregate traffic rate. Let m_i and σ_i^2 be the mean and variance of VBR source i, then the mean m and the variance σ^2 of the superposition of n processes should be $\sum_{i=1}^{n} m_i$ and $\sum_{i=1}^{n} \sigma_i^2$ respectively. Given the cell loss probability ϵ, the required capacity can be approximated by [16]:

$$C_{nrtvbr} = m + \zeta\sigma \quad (4.16)$$

where $\zeta \approx 1.8 - 0.46\log_{10}(\sqrt{2\pi}m\epsilon/\sigma)$

4.6 Capacity Matrix

After above procedure, the capacity for CBR, real-time VBR and non real-time VBR services on a virtual path have been determined. The sum of them is the required capacity of the virtual path to support their QOSs, which can be represented by following $K \times GK$ matrix T:

$$T = \begin{pmatrix} \lambda_{0,0,0} & \lambda_{0,1,0} & \cdots & \lambda_{0,G-1,0} & \cdots & \lambda_{0,G-1,K-1} \\ \lambda_{1,0,0} & \lambda_{1,1,0} & \cdots & \lambda_{1,G-1,0} & \cdots & \lambda_{1,G-1,K-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \lambda_{K-1,0,0} & \lambda_{K-1,1,0} & \cdots & \lambda_{K-1,G-1,0} & \cdots & \lambda_{K-1,G-1,K-1} \end{pmatrix}$$

where K is the number of input(output) modules and $\lambda_{i,j,k}$ is the required capacity from input module I_i to the jth cluster of output module O_k. It is possible that the sum of a row or a column is less than M, the capacity of central stage. Then we need to allocate the residual bandwidth to virtual paths by certain criteria, and these extra bandwidth can be used by ABR and UBR services. Finally, the following $K \times GK$ capacity matrix C can be found:

$$C = \begin{pmatrix} c_{0,0,0} & c_{0,1,0} & \cdots & c_{0,G-1,0} & \cdots & c_{0,G-1,K-1} \\ c_{1,0,0} & c_{1,1,0} & \cdots & c_{1,G-1,0} & \cdots & c_{1,G-1,K-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{K-1,0,0} & c_{K-1,1,0} & \cdots & c_{K-1,G-1,0} & \cdots & c_{K-1,G-1,K-1} \end{pmatrix}$$

$$\sum_{j=0}^{G-1} c_{i,j,k} = M/G$$

$$\sum_{i=0}^{K-1} \sum_{j=0}^{G-1} c_{i,j,k} = M$$

$$\sum_{k=0}^{K-1} \sum_{j=0}^{G-1} c_{i,j,k} = M$$

where $c_{i,j,k}$ is the capacity allocated to that virtual path, which must equal to or be larger than $\lambda_{i,j,k}$. $c_{i,j,k}$ may be obtained by optimizing some objective functions such as average delay of virtual path. Multiply C by F—the frame size and do some roundoff, so that elements in matrix E are integers.

$$E = F \cdot C = \begin{pmatrix} e_{0,0,0} & e_{0,1,0} & \cdots & e_{0,G-1,0} & \cdots & e_{0,G-1,K-1} \\ e_{1,0,0} & e_{1,1,0} & \cdots & e_{1,G-1,0} & \cdots & e_{1,G-1,K-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{K-1,0,0} & e_{K-1,1,0} & \cdots & e_{K-1,G-1,0} & \cdots & e_{K-1,G-1,K-1} \end{pmatrix}$$

Since total capacity is FM and equally assigned to G groups, $e_{i,j,k}$ must satisfy the following restrictions:

$$\sum_{j=0}^{G-1} e_{i,j,k} = FM/G$$

$$\sum_{i=0}^{K-1} \sum_{j=0}^{G-1} e_{i,j,k} = FM$$

$$\sum_{k=0}^{K-1} \sum_{j=0}^{G-1} e_{i,j,k} = FM$$

4.7 The Evaluation of Multiplexing Gain of Input Stage

In ATM networks, connections(virtual circuits) are multiplexed on virtual path, then traffic on virtual paths are multiplexed on physical link. The multiplexing improves the utilization by exploiting the fact that when sources are bursty, some sources generate low or no traffic much of the time. The statistical characteristics of traffic makes it possible that Quality of Service(QOS) can be guaranteed while maintaining sufficient utilization. The multiplexing gain can be defined as the ratio between the effective bandwidth of a source when it is transmitted alone, and the required bandwidth of a source when it is multiplexed with others. The required bandwidth of aggregate traffic depends on the number

of multiplexed sources, buffer size and QOS. In path switch, traffics are statistically multiplexed on the virtual paths, the evaluation of multiplexing gain is important for dimensioning and call admission control. In this section, the multiplexing gain which changes with the number of sources on a path, will be evaluated for homogeneous real-time and non real-time VBR traffics.

To analysis the multiplexing gain in input module, exponential on-off sources are assumed. The traffic streams generated by all sources are independent and identically distributed(i.i.d). Traffics are uniformly distributed on all the virtual path and sources on a virtual path share both buffer and capacity.

Assuming N_s identical sources per input module, the number of clusters is KG, so there are $n_s = N_s/KG$ sources per cluster. For simplicity, the peak rate R and the average burst length $1/\mu$ are normalized to 1. The buffer size per source and the required capacity per source is denoted by B_s and C_s respectively. Thus the n_s sources are multiplexed in a buffer of size $n_s B_s$ with service rate of $n_s C_s$.

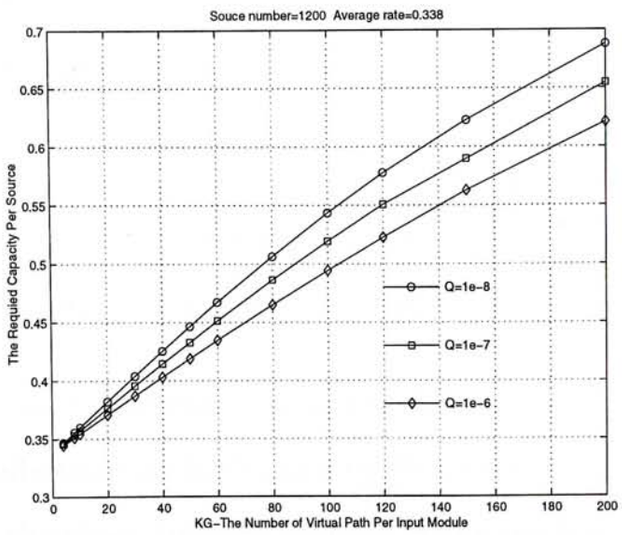

Figure 4.2: The Required Bandwidth per Non Real-time Source with $\rho = 0.338$

For non real-time VBR sources, a large buffer can be allocated to avoid overflow, though a long period of queueing delay will result. Given the buffer overflow probability ϵ, the required C_s can be calculated accurately by equation 52 in [1]. When $N_s = 1200$, $\lambda = 0.51$, $B_s = 1$, the relation between C_s and KG is shown in figure 4.2, given $\epsilon = 10^{-6}, 10^{-7}$ and 10^{-8} respectively. The effective bandwidth of a single source is estimated to be 0.93, 0.94, 0.9473 respectively by equation 4.15, which is much higher than the average rate of 0.3377. The capacity for the aggregate traffic is much less than the effective bandwidth which is close to peak rate. It is verified that the effective bandwidth is too conservative in this case, especially when the number of source on a virtual path is large, because the utilization of source is low(0.3377).

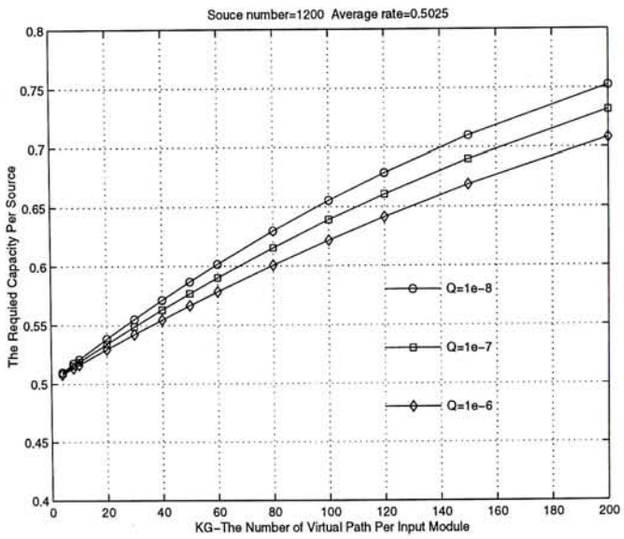

Figure 4.3: The Required Bandwidth per Non Real-time Source with $\rho = 0.5$

When $\lambda = 1.01$ so the average rate is 0.5, the required capacity is presented in figure 4.3. The effective bandwidth of a single source is estimated to be 0.9329, 0.9418, 0.9487 for the three different overflow requirements.

It is shown that the required capacity per source is almost a linear increasing

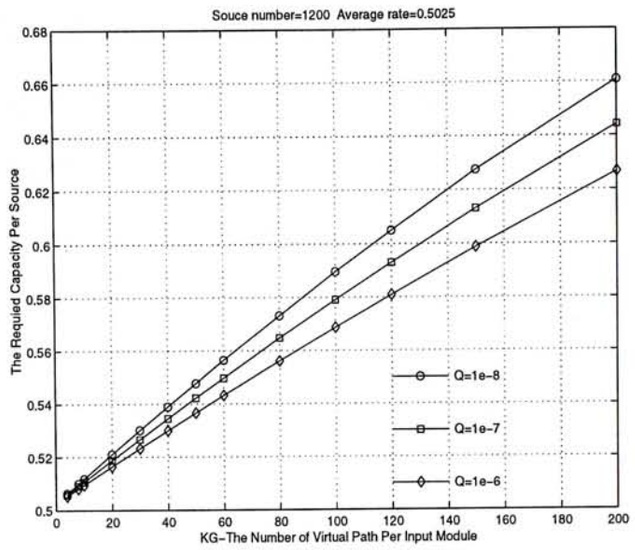

Figure 4.4: The Required Bandwidth per Non Real-time Source with $B_s = 2$

function as the number of virtual path. When a large number of sources are multiplexed on a virtual path, the shared buffer size is large so that the bursty traffic is smoothed out. The superposition of a lot of bursty sources is much smoother than the single one according to the central limit theorem. When a lot of sources are multiplexed on a path, the required capacity is a bit more than the average rate, compared with the effective bandwidth of a single source which is close to peak rate. The capacity assignment based on effective bandwidth is then too conservative which will admit less sources to the switch. The path switch can take advantage of multiplexing gain quite well, resulting in efficient utilization of resources.

When the virtual path is getting narrower, the number of sources on a path gets smaller and the required capacity gradually increases. This is because the shared buffer size is smaller and the probability that most of sources are generating traffic simultaneously is much greater. In this case, the multiplexing gain is no longer significant so that the utilization efficiency of resources is degraded.

However, the loss of multiplexing gain is not serious, with high utilization of source or a large buffer. The multiplexing gain with average rate 0.5 in figure 4.3 is lower than that in figure 4.2 with average rate 0.3377. Under this situation, the probability that a source is "on" is higher, which makes it harder to take advantage of the bursty characteristics. In extreme case where all sources generate traffic at the peak rate constantly, there is no multiplexing at all. If the buffer is large enough to smooth out the burst, the multiplexing among sources is not significant due to the same reason. As illustrated in figure 4.4, the loss of multiplexing gain is less than 30% if buffer size of 2 is allocated to a source($B_s = 2$).

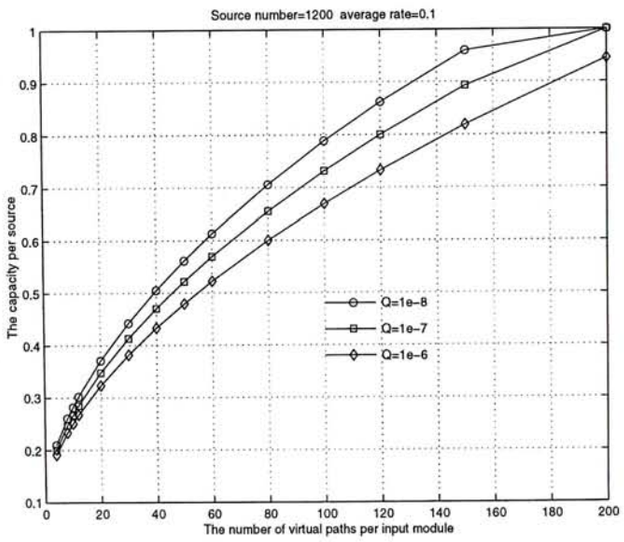

Figure 4.5: The Required Bandwidth Per Real-time Source

For real time VBR sources, the buffer overflow probability can always be approximated by a zero buffer. Given $N_s = 1200$, $\lambda = 0.111$, $B_s = 0$, $\epsilon = 10^{-6}, 10^{-7}$ and 10^{-8} respectively, the relation between C_s and KG is shown in figure 4.5. The required capacity varies in a wide range, from 0.2 to nearly peak rate with stringent QOS, and it increases almost exponentially with the number

of virtual paths. When a large number of real-time sources are multiplexed on a virtual path, the aggregate traffic gets smoother so that the probability of the aggregate traffic rate exceeds the service rate is small. Whereas, when only a few sources share capacity on a virtual path , zero or small buffer can not smooth out the bursty traffic, so that the capacity must be large enough to guarantee the tight delay constraint. Great loss of multiplexing gain and consequently quite poor efficiency of resource will result in, if the virtual path is too "narrow". When there are small number real-time sources multiplexed on virtual path, it is better keep the virtual path "wide" enough to achieve high efficiency. Usually, the real-time services can tolerate relative high cell loss rate, such as voice and video. And coding technology will recover the lost cells or errors. Non real-time services such as data communication, are more susceptible to the cell loss. So narrow virtual path is suggested for non real-time traffic and wide one for real-time traffic, though the concentration loss will be greater for real-time traffic. In the extreme case, there is no partition for real-time traffic. At input module, cells of non real-time services will be selected by their cluster number while those of real-time services will selected by their output module number. At output port, priority will be given to cells of non real-time services if contention occurs.

Chapter 5

Discussions and Conclusions

We have proposed a virtual path scheduling scheme to manipulate virtual paths flexibly and alleviate the output contention at cross-path switch. The definition of virtual path is no longer confined by the physical parameters of switch architecture and the bipartite graph can be asymmetric. Since the scheduling scheme is implemented by upper layer software, it has the flexibility to alter G in order to achieve various performance. Once the hardware of switch is built, only G needs to be changed to meet diverse requests. Splitting of the original virtual path is shown to be able to lower cell loss probability. In reality, the traffic loading is time-variant, and traffic characteristics of VCs are various. It is known that output contention is not sensitive to input loading [24]. When traffic loading is heavy, small value of G should be adopted to keep high throughput and multiplexing gain, since the cell loss is mainly due to buffer overflow instead of contention. Under light traffic loading, G should be large enough to limit the cell loss due to contention as the maximum throughput is not a major concern.

Compared with the space-division switch, memory switch is suggested at

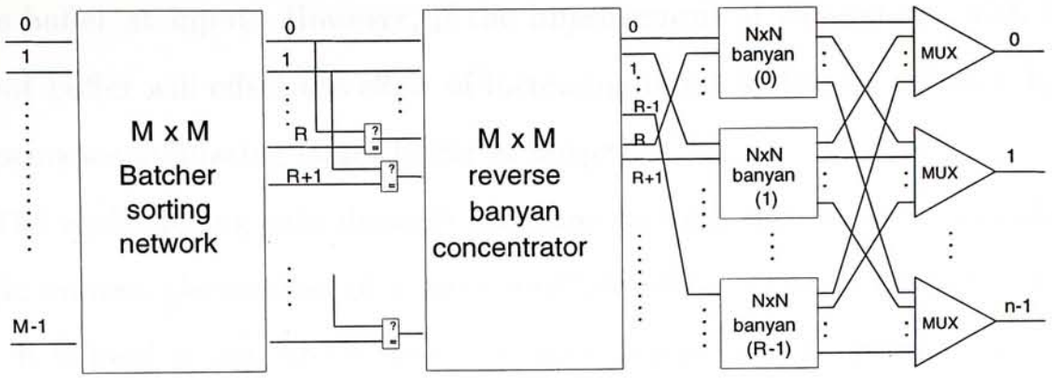

Figure 5.1: The Architecture of the Batcher-R-banyan Knockout Switch

input stage to keep high throughput and make cell scheduling more feasible. No special contention resolution scheme or concentrator would be needed if the last stage is implemented by a Batcher-R-banyan knockout switch in figure 5.1 [6] and R is set to M/G, because there will be no gaps between active cells after the sorting procedure. If a multi-buffered memory switch is used as output module, the memory access speed is reduce from M to M/G so that the cost is lower. Thus, the implementation of cross-path switch can be greatly simplified.

With dedicated buffer at input and output port, the maximum throughput and cell loss probability are compared under queue loss mode(QL) and backpressure mode(BP). The buffer dimensioning under these two modes among input and output are studied with different settings. The BP mode always performs better than QL mode in terms of cell loss probability, if buffer is properly allocated among input and output. When maximum throughput is low, more buffer should be given to input port under BP mode, because the backpressure effect is not sensitive to output buffer size so that cell loss will be improved by placing

more buffer at input. However, if the improvement of throughput with large output buffer will offset the effect of increasing input buffer, we obtain a better performance by placing larger buffer at output.

The multiplexing gain depends on many factors, such as the attributes of traffic sources, the number of sources multiplexed on a virtual path, buffer size etc. It is hard to say which choice of switch parameters is optimal. However, given the traffic at an instant, the multiplexing gain can be evaluated for the purpose of system administration. The multiplexing gain may be degraded a lot for real-time traffic if only a small number of sources are multiplexed on a virtual path; in that case, we should keep a wide path for real-time traffic.

Bibliography

[1] D. Anick, D. Mitra and M. M. Sondhi, "Stochastic Theory of a Data-Handling System With Multiple Sources", Bell System Tech. J., 61, (1981) 1871-1894

[2] The ATM Forum "Traffic Management specification", Version 4.0, 1995

[3] K. E. Batcher, "Sorting Networks and Their Applications", Proc. 1968 Spring Joint Comput. Conf.

[4] V. E. Benes, "Mathematical Theory of Connecting Networks and Telephone Traffic", Academic Press, New York, 1965

[5] D. G. Cantor, "On Non-Blocking Switching Networks", Networks, Vol. 1, No. 4, Winter 1971, 367–377

[6] Cheng-Shang Chang and Joy A. Thomas, "Effective Bandwidth in High-Speed Digital Networks", IEEE JSAC, Vol 13, No 6, August 1995, 109–1100

[7] Jeane S-C Chen and Thomas E. Stern, "Throughput Analysis, Optimal Buffer Allocation, and Traffic Imbalance Study of a Generic Nonblocking Packet Switch", IEEE JSAC, Vol 9, No. 3, April 1991, 439-449

[8] C. Clos, "A Study of Non-Blocking Switching Networks", Bell Syst. Tech. J., Vol. 32, March 1953, 406–424

[9] Imrich Chlamtac, Andras Farago and Tao Zhang, "Optimizing the System of Virtual Paths", IEEE Trans. Networking, Vol. 2, No. 6, December 1994, 581–587

[10] Michel Devault, Jean Yves Cochennec and Michel Servel, "The "Prelude" ATD Experiment: Assessments and Future Prospects", IEEE JSAC, Vol. 6, No. 9, December 1988, 1528–1537

[11] Lisa G. Dron, Gopalakrishnan Ramamurthy and Bhaskar Sengupta, "Delay Analysis of Continuous Bit Rate Traffic Over an ATM Network", IEEE JSAC, Vol 9, No 3, April 1991, 402–407

[12] Anwar Elwalid and Debasis Mitra, "Effective Bandwidth of General Markovian Traffic Source and Admission Control of High Speed Networks", IEEE/ACM Trans. Networking, 1 (1993) 329–343

[13] Anwar Elwalid, Debasis Mitra and Tober H. Wentworth, "A New Approach for Allocating Buffers and Bandwidth ot Heterogeneous, Regulated Traffic in an ATM Node", IEEE JSAC, Vol. 13, No. 6, August 1995, 1115–1127

[14] Noboru Endo, Takahiko Kozaki etc, "Shared Buffer Memory Switch for an ATM Exchange", IEEE Trans. Comm., Vol. 4, No. 1, January 1993, 237–245

[15] Andras Farago, Soren Blaabjerg, Laszlo Ast, Geaz Gordos and Tamas Henk, "A New Degree of Freedom in ATM Network Dimensioning: Optimizing

the Logical Configuration", IEEE JSAC, Vol. 13, No. 7, September 1995, 1199–1205

[16] Roch Guerin, Hamid Ahmadi and Mahmoud Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks", IEEE JSAC, Vol 9, No 7, September 1991, 968–981

[17] Michael G. Hluchyj and Mark J. Karol, "Queueing in High-Performance Packet Switching", IEEE JSAC 6 (1988) 1587–1597

[18] Ivy Hsu and Jean Walrand, "Admission Control for ATM Networks", IMA Workshop on Stochastic Networks, Minneapolis, Minnesota, March 1994

[19] J. Y. Hui, "Resource Allocation for Broadband Networks", IEEE JSAC, Vol. 6, No. 9, December 1988, 358–368

[20] Ivy Hsu and Jean Walrand, "Admission Control for Multi-Class ATM Traffic with Overflow Constraints", Computer Networks and ISDN Systems Journal for the Special Issue on High Speed Networks and Application

[21] Ilias Iliadis and Wolfgang E. Denzel, "Analysis of Packet Switches with Input and Output Queueing", IEEE Trans. Comm. Vol 41, No. 5, May 1993, 731-740

[22] Sugih Jamin, Peter B. Danzig, Scott J. Shenker and Lixia Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks", IEEE Trans. Networking, Vol 5, No 1, February 1997, 56–69

[23] Mark J. Karol, Michael G. Hluchyj and Samuel P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch", IEEE Trans. Commun. 35 (1987) 1347–1356

[24] Yu-Shuan Yeh, Michael G. Hluchyj and Anthony S. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching", IEEE JSAC 5 (1987) 1274–1283

[25] Kai Y. Eng, Mark J. Karol and Y.S.Yeh, "A Growable Packet(ATM) Switch Architecture: Design Principles and Applications", Conf. Rec. Globecom'89, 32.2.1–32.2.7

[26] Mark J. Karol and Chih-Lin I, "Performance Analysis of a Growable Architecture for Broadband Packet(ATM) Switching", Globecom'89, 32.4.1–32.4.8

[27] Mark J. Karol, Kai Y. Eng and Hitoshi Obara, "Improving the Performance of Input-Queued ATM Packet Switches", Proc. INFOCOM'92, 110–115

[28] George Kesidis, Jean Walrand and C-S Chang, "Effective Bandwidth for Multiclass Markov Fluids and other ATM Sources", IEEE Trans. Networking, Vol 1, No 4, August 1993, 424–428

[29] Harufusa Kondoh, Hiromi Notani etc, "A 622-Mb/s 8×8 ATM Switch Chip Set with Shared Multibuffer Architecture", IEEE JSSC, Vol. 28, No. 7, July 1993, 808–815

[30] Cheuk-Hung Lam, "Virtual Path Traffic Management of Cross-Path Switch", PhD Thesis, Department of Information Engineering, The Chinese University of Hong Kong, 1997

[31] Myung Jong Lee and San-Qi Li, "Performance of a Nonblocking Space-division Packet Switch in a Time Variant Nonuniform Traffic Environment", IEEE Trans. Comm. Vol 39, No. 10, October 1991, 1515-1524

[32] Myung J. Lee and David S. Ahn, "Cell Loss Analysis and Design Trade-offs of Nonblocking ATM Switches with Nonuniform Traffic", IEEE Trans. Networking, Vol 3, No. 2 April 1995, 199-210

[33] Tony T. Lee, "A Modular Architecture for Very Large Packet Switches", IEEE Trans. Commun. 38 (1990) 1097–1106

[34] Tony T. Lee and Cheuk H. Lam, "Path Switching—A Quasi–Static Routing Scheme for Large–scale ATM Packet Switches", IEEE JSAC 5 1997, 914-924

[35] Tony T. Lee and Soung-Yue Liew, "Parallel Routing Algorithms in Benes–Clos Networks", Proc. INFOCOM'96, 279–286

[36] Oo Tang and Tony T. Lee, "Virtual Path Scheduling for Large-Scale ATM Switches", Proceeding of the 5th International Conference on Telecommunication Systems, Nashville, Tennessee, USA, March 1997

[37] Hyong W. Lee and Jon W. Mark, "Capacity Allocation in Statistical Multiplexing of ATM Sources", IEEE Trans. Networking, Vol. 3, No. 2, April 1995, 139–151

[38] San Qi Li, "Nonuniform Traffic Analysis on a Nonblocking Space-Division Packet Switch", IEEE Trans. Comm., Vol. 38, No 7, July 1990, 1085–1096

[39] San Qi Li, "Performance of a Nonblocking Space-Division Packet Switch With correlated Input Traffic", IEEE Trans. Comm., Vol 40, No 1, Jan 1992, 97–108

[40] Soung-Chang Liew, "Performance of Various Input-buffered and Output-buffered ATM Switch Design Principles Under Bursty Traffic: Simulation Study", IEEE Trans. Comm., Vol. 42, No. 2/3/4, 1994, 1371–1379

[41] Soung-Chang Liew and Tony T. Lee, "Principles of Broadband Switching and Networks", Lecture Notes, Chinese University of Hong Kong (1996)

[42] Soung-Chang Liew and Kevin W, Lu, "A 3-Stage Interconnection Structure for Very Large Packet Switches", Conf. Rec, ICC'90, 316.7.1–316.7.7

[43] Soung-Chang Liew and Kevin W. Lu, "Comparison of Buffering Strategies for Asymmetric Packet Switch Modules", IEEE JSAC, Vol. 9, April 1991, 428–438

[44] Soung-Chang Liew and Kevin W. Lu, "A 3-Stage Interconnection Structure for Very large Packet Switches", conf. Rec., ICC'90 316.7.1-316.7.7, 1990

[45] Nick McKeown, Venkat Anantharam and Jean Walrand, "Achieving 100% throughput in an Input-Queued Switch", Proc. INFOCOM'96, 296–302

[46] D. Medhi, "Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-Based Dynamically Reconfigurable Wide-Area ATM Networks", IEEE Trans. Networking, Vol. 3, No. 6, December 1995, 809–818

[47] Riccardo Melen and Johathan S, Turner, "Nonblocking Networks for Fast Packet Switching", IEEE Trans. Comm. Vol. 41, No. 2, Feb 1993, 362–369

[48] Debasis Mitra, John A. Morrison and K.G. Ramakrishnan, "ATM Network Design and Optimization: A Multirate Loss Network Framework", IEEE Trans, Networking, Vol. 4, No. 4, August 1996, 531–543

[49] Craig Partridge, "Gigabit Networking", Addison-Wesley Publishing Company, 1994

[50] Achille Pattavina and Giacomo Bruzzi, "Analysis of Input and Output Queueing for Nonblocking ATM Switches", IEEE Trans. Networking, Vol 1, No. 3, June 1993, 314-327

[51] Martin de Prycker, "Asynchronous Transfer Mode Solution for Broadband ISDN", Ellis Horwood, 1995

[52] G. Ramamurthy and Qiang Ren, "Multi-Class Connection Admission Control Policy for High Speed ATM Switches", INFOCOM'97,

[53] J. W. Roberts, "Performance Evaluation and Design of Multi-service Networks", COST 224 project, 1992

[54] Yoshito Sakurai, Nobuhiko Ido, Shinobu Gohara and Noboru Endo, "Large-Scale ATM Multistage Switching Network with Shared Buffer Memory Switches", IEEE Communications Magazine, January 1991, 90–96

[55] Youichi Sato and Ken-ichi Sato, "Evaluation of Statistical Cell Multiplexing Effects and Path Capacity Design in ATM Networks", IEICE Tans. Comm, Vol. E75-B, No. 7, July 1992, 642–648

[56] Ken-ichi Sato, Satoru Ohta and Ikuo tokiazwa, "Broadband ATM Network Architecture Based on virtual Paths", IEEE Trans. Comm., Vol. 38, No. 8, August 1990, 1212–1222

[57] Alain Simonian and Jacky Gulber, "Large Deviations Approximation for Fluid Queues Fed by a Large Number of On/Off Sources", IEEE JSAC, Vol 13, No. 6, August 1995, 10171027

[58] Kotikalapudi Sriram and Ward Whitt, "Characterizing Superposition Arrival Processes in Packet Multipexers for Voice and Data", IEEE JSAC, vol. 4, No. 6, September 1986, 833–846

[59] Philip W. Tse and Moshe Zukerman, "Evaluation of Multiplexing Gaing",

[60] G. de Veciana, G. Kesidis and J. Walrand, "Resource Management in Wide-Area ATM networks Using Effective Bandwidths", IEEE JSAC 13 (1995) 1081–1090

[61] R. J. Wilson, "Introduction to Graph Theory"(Academic Press, New York, 1972)

[62] Hideaki Yamanaka, Hirotaka Saito etc, "Scalable Shard-Buffering ATM Switch with a Versatile Searchable Queue", IEEE JSAC, Vol. 15, No. 5, June 1997, 773–784

[63] Hui Zhang, "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks," IEEE Proceedings, October 1995.