Path Switching over

Multirate Benes Network

By

MUI Sze Wai

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
July 2003

Acknowledgement

I would like to take this opportunity to express my deepest gratitude towards my supervisor, Prof. Tony T. Lee, for his continuous support, encouragement and invaluable advice. His deep insight and broad vision on research broadened my outlook and taught me a lot.

Without my friends in Broadband Communication Laboratory, the last two years would lack of vitality. I would like to thank Mr. Wai-hung Man, for his comments and continuous discussion on my research work; Mr. Lui-fuk Lam, Mr. Man-ting Choy, Mr. Yun Deng, Mr. Tsz-chung Wong and Mr. Liang Zhang for their various kinds of help during these two years. I would like to thank them for enriching my life in CUHK and making this two-year research period a fruitful and rewarding experience.

Finally, I am grateful to my family for their selfless love and support.

摘要

Benes 網絡在高速數據包交換網絡的發展一直受到限制，問題出現在分派非阻塞路徑上。很多熟悉的順序分派路線算法，就例如在爲電路交換而設計的環繞算法中，只可以以較低的速度重新分配交換配置。在數據包交換網絡中，交換單位需要在每一個時隙改變它們的狀態。對數據包交換網絡來說，大部分現存的并行算法都不實際，因爲它們都假定每個輸入端口都繁忙。

最近提出一種新的用於大型數據包交換網絡的路由方案名爲路徑交換。路徑交換是一種介於靜態路由和動態路由之間的方案。它將一種預定的週期連接模式用於中央階段。

本篇文章中，我們提出運用路徑交換於 Benes 網絡的路徑分配中。因爲連接模式是重複使用的，我們就可以避免實時尋

找路徑，從而消除 Benes 網絡在高速數據包交換時的瓶頸。通過每個數據會話的到達及服務曲線，我們研究這種網絡的性能並且得到會話延遲、輸入輸出端口積壓的上限。我們證明這種網絡可以比原有的 Benes 網絡更有效率地處理均勻及多速率的數據流。

Abstract

The development of Benes network in high-speed packet switching is limited by the problem of assigning nonblocking routes. Most known sequential route assignment algorithms, such as the looping algorithm, are designed for circuit switching systems where the switching configuration can be rearranged at a relatively low speed. In packet switching, switching elements may need to change their states every time slot. Most existing parallel algorithms are also not practical for packet switching as they assume that every input port is busy.

A novel routing scheme called path switching was proposed for large-scale packet switches. Path switching is a compromise of static routing and dynamic routing schemes. It uses a predetermined periodical connection pattern in the central stage.

In this thesis, we propose to use path switching for route assignment in Benes network. As the connection pattern is used repeatedly, we can

avoid path hunting on the fly, which is the bottleneck when Benes network is operated in high-speed packet switching. We then study the performance of the path-switched Benes network at each data session by using arrival and service curves and obtain the upper bound on session delay, input and output backlog. We show that path-switched Benes network can handle uniform and multirate traffic more effectively than original Benes network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the last decade, there has been a growing interest in communication networks that are capable of serving applications with widely varying characteristics. This motivates the study of multirate interconnection networks. In particular, such networks are being designed to support connections with different bandwidth requirements, including voice, data, video and multimedia traffic streams. The data rates vary from a few bits per second to hundreds of megabits per second [6]. These multirate networks carry information in multiplexed format, with each connection consumes only an arbitrary fraction of the link bandwidth. Typically, the information is carried in the form of independent blocks called packets or cells.

1.1 Evolution of Multirate Networks

Initially both multirate circuit switching and connection-oriented packet switching were considered as potential technologies for broadband networks, but connection-oriented packet switching, in the form of Asynchronous Transfer Mode (ATM) technology was soon chosen as the most promising technical solution. The development of a theory of multirate interconnection networks stem from the necessity of modeling a new generation of switches for broadband services. While the first studies were motivated by multirate circuit switches [5], the advent of ATM re-oriented the research towards packet switching.

1.2 Some Results from Previous Work

In [11], it establishes the nonblocking conditions for multirate traffic with different types of networks and compares their complexity. The networks being investigated are Clos, Cantor and Benes and banyan networks. The result shows that nonblocking operation can be obtained for multirate traffic with essentially the same complexity as in the classical theory of nonblocking network. It also points out that both the Benes network and the banyan network appear to be the most attractive for larger packet size.

The simulation results in [7] also illustrates that the Benes network performs well under multirate traffic. A 64 × 64 Benes network can achieve nearly 80% throughput under multirate traffic with mean bandwidth of each connection equal to 0.3 and the blocking probability is about 10^{-4} when offered load is 0.5.

Banyan networks and Benes networks both belong to multistage interconnection networks (MIN) with a single path between any input-output pair in banyan networks while multipaths between inputs and outputs in Benes network. Multipath MINs have two potential advantages: the traffic distribution may be kept more uniform throughout the MIN to minimize internal conflicts, and the MIN is more fault tolerant.

In banyan networks, the routes of two packets destined to different outputs might conflict before the last switching stage. This condition is called internal blocking, in which only one of the two packets contending an outgoing link can be passed to the next stage, while the other packet is being dropped. Thus, the overall throughput is reduced. A nonblocking condition for the banyan network was found in [13] and later proved in [8] and [20]. To ensure nonblocking operation of the

banyan network, the set of input packets to the network must be concentrated and with monotonic output destinations. Therefore, a sorting network, e.g. Batcher bitonic sorting network, is always added before the banyan network to sort the address of the packets.

Both Batcher-banyan network and Benes network may experience output contention if two packets are destined to the same output address. It must solved by buffering. One approach is to buffer the packets at the input of the network. If two or more packets are destined to the same output address, only one packet is allowed to past while the other packets are queueing in the buffer. Nevertheless, most packet scheduling algorithm assume switches are output buffered. Moreover, it cannot achieve high throughput due to the head-of-line (HOL) blocking, where the packets behind a delayed packet cannot pass through the network. The HOL blocking problem can be reduced by using look-ahead contention resolution scheme [12]. The throughput of the switch is found to increase monotonically with an increase in window size.

It is clear that output buffering is the most preferable approach for achieving optimal throughput. Unfortunately, banyan networks cannot realize output buffering straightforwardly because they deliver at most

one packet per time interval to any output. To realize output queueing, we should deliver multiple packets per time interval to the same output, for example in Benes network.

1.3 Multirate Traffic on Benes Network

We have seen that Benes network is a promising candidate for handling multirate traffic and much work has been done on rearrangably nonblocking and strictly nonblocking conditions on multirate Benes networks [11], [14].

Melen and Turner extended the problem of routing a set of connections through a Clos network to the multirate environment in [3]. They use the bipartite graph to represent the set of connections. The edges of the graph are assigned weights between 0 and 1 that represent the bandwidth used by each of the connection. In the weighted bipartite graph, the edges incident to a single vertex are allowed to have the same color, as long as the total weight of these edges does not exceed the speedup factor of the switch. This is illusrtated in Figure 1.1, where a set of multirate connections is listed at the top and the corresponding weighted graph is shown at left. It is important to point out that each edge corresponds to one connection and not to one physical link.

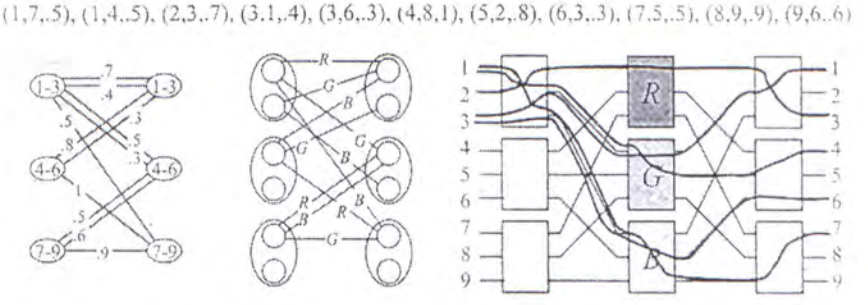


Figure 1.1: Simultaneous routing of multirate connections in 3-stage Clos network.

This weighted bipartite graph coloring problem can be converted to an ordinary graph coloring problem by splitting each of the vertices and associating different subsets of the edges incident to the vertex with different sub-vertices. In particular, the n heaviest edges are all assigned to the same sub-vertex, the next n heaviest edges are assigned to another sub-vertex, and so forth, as illustrated in the middle part of Figure 1.1. When the splitting process has been applied to all vertices, the resulting graph has at most n edges incident to each vertex and so can be colored in the ordinary way using just n colors.

The algorithm described above can also be used in $N \times N$ Benes networks if it is applied recursively at each of the first $\log_2 N$-1 and is refereed as the Balanced Vertex Splitting (BVS) algorithm [11]. However, there are still some cases that the sets of connections routed by BVS needed to be rearranged, e.g. route (1, 7) and (3, 6) at the same

time in Figure 1.1. In this thesis, we will propose a novel method to handle multirate traffic for Benes networks, that is path switching [1] over Benes network. In this model, a finite set of interconnection patterns is pre-calculated and used repeatedly, such that the capacity allocated to all the input-output pairs are satisfied with the connection requests in long term average.

1.4 Organization

The rest of the thesis is organized as follows. Chapter 2 gives some background knowledge about Benes network and path switching. The model of path switching over Benes network and the implementation details are presented in chapter 3. An example of multirate traffic over an 8×8 path-switched Benes network is also included. In chapter 4, we use a graphical model to study the arrival curve and service curve of a data session. We also show that the maximum delay, input backlog and output backlog can be easily calculated from these curves. We then establish the upper bounds on these quantities at the end of chapter 4. Chapter 5 compares the simulation results of our model with original Benes networks. Finally, we conclude the main work of the thesis and give some suggestions for future research.

Chapter 2

Background Knowledge on Benes Network and Path Switching

In this chapter, we will first have an introduction on Benes network, including its properties, construction and routing. Then we will discuss the pros and cons when Benes network is operated as large switch fabrics. The next subsection introduces the basic concept of path switching and describes the route scheduling of the Clos network. Finally, we provide an example on path switching over Clos network under nonuniform traffic.

2.1 Benes Network

The Benes network is rearrangeably nonblocking [4], there exist several alternative paths from any input port to any output port. If a complete list of input-output pairs is given for Benes network, it is always possible to find a set of routes to satisfy all requests, provided that no two input ports want to reach the same output port.

2.1.1 Construction of Large Switching Fabrics

To construct an $N \times N$ Benes network, we can use a recursive method. The network can be broken down into three stages consisting of 2×2 modules in the first and third stages and two $N/2 \times N/2$ modules in the middle. The $N/2 \times N/2$ modules can be again broken down, only 2×2 modules will remain in the end. Figure 2.1 shows a Benes network with $N = 16$.

Let us assume that $N = 2^n$ and let $f(k)$ be the number of stages in an $k \times k$ Benes network. By the recursive construction, we have

$$f(N) = f(\frac{N}{2}) + 2 \qquad (2.1)$$

$$
\begin{aligned}
f(2^n) &= f(2^{n-1}) + 2 \\
&= f(2^{n-2}) + 2(2) \\
&\vdots \\
&= f(2^{n-i}) + 2i \\
&\vdots \\
&= f(2) + 2(n-1) \\
&= 1 + 2(n-1) \\
&= 2n - 1
\end{aligned} \tag{2.2}
$$

Since each stage has $N/2$ modules, the total number of modules is

$$
\begin{aligned}
&\frac{N}{2} \times (2n-1) \\
&= Nn - \frac{N}{2} \\
&= N \log_2 N - \frac{N}{2}
\end{aligned} \tag{2.3}
$$

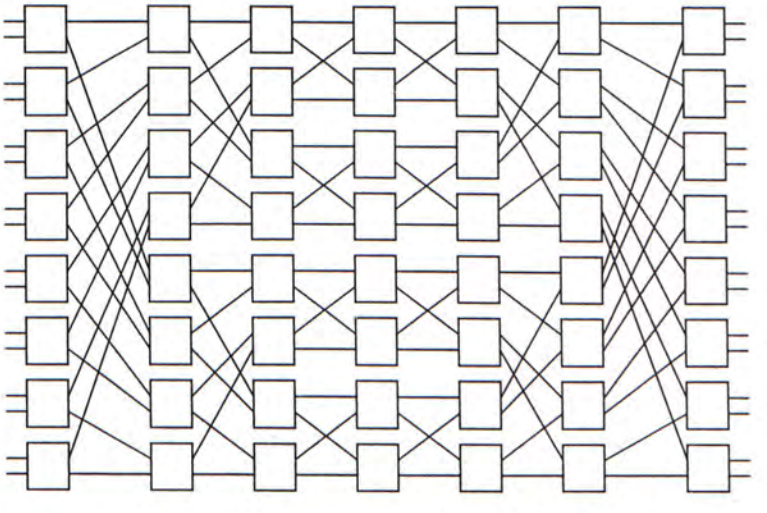


Figure 2.1 A Bene network with $N = 16$

The Benes network satisfies the $N\log N$ lower bound on the number of crosspoints required in a nonblocking switch.

2.1.2 Routing in Benes Network

The Benes network consists of two subnetworks between the first and third columns of 2 × 2 modules. To setup the path for an input and output pair, one way is to determine whether it should go through the upper or lower subnetwork. Since there is only one link form a 2 × 2 module to a subnetwork, the input and outputs sharing the same 2 × 2 module must go through different subnetwork. If the upper path is selected, then the other input-output pair share with the same output module must choose the lower path and vice versa. Once the set of paths have been determined at that level, we can go down to the next level of setting paths within each of the two $N/2 \times N/2$ subnetworks. The algorithm can be applied in a recursive manner until the paths of all the input-output pairs are determined. This is called the looping algorithm [16].

Let's illustrate the looping algorithm by an example. We want to setup paths for the following input-output pairs: (1,5), (2,8), (3,1), (4,6), (5,2), (6,7), (7,3), (8,4). We start the route by letting the first path from input 1 through the upper subnetwork and reaches output 5. Since output 6 shares the same module as output 5, the path from input 4 to output 6 must go through the lower subnetwork. By the same principle, the path

form input 3 to output 1 must go through the upper subnetwork. Performing this iteratively by satisfying the sharing properties at the input and output alternatively, we will close the loop by establishing a path from input 2 to output 8 through the lower subnetwork, as shown in Figure 2.2(a). The loop may not involve all the inputs and outputs. In this example, the path (7,3) and (8,4) have not been considered by the first loop. We can start the path setup procedure as above starting from the input that are not involved in the first loop. In this way, the connections from all input-output pairs will be established in the end. Figure 2.2(b) shows the routes after two iterations.

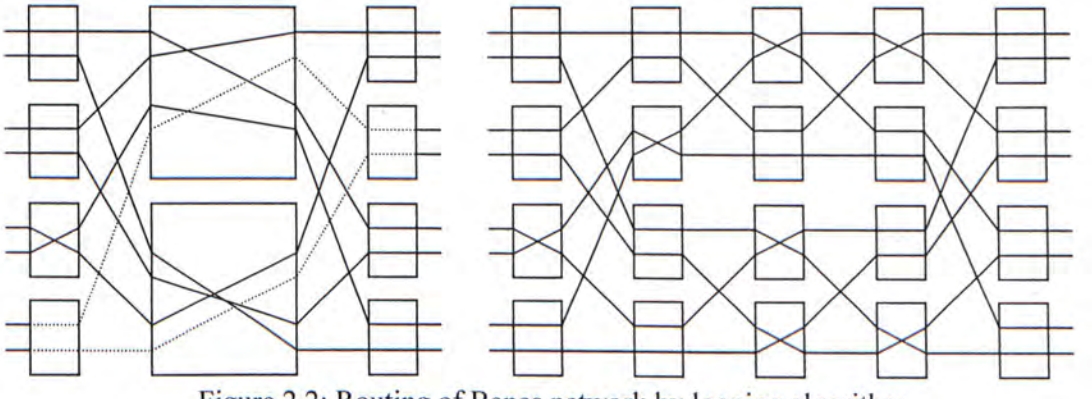


Figure 2.2: Routing of Benes network by looping algorithm.
(a) One iteration, (b) Two iterations.

The number of step required to determined the path in the first-level is N. Since each step depends on the step before, these steps cannot be executed concurrently. At the next level, $N/2$ steps are required. As there are two subnetworks, a total of N steps are required. The total

number of steps needed is $N\log_2 N$ since there are $\log_2 N$ levels. However, if we parallelize the path setup procedures in separate subnetworks, the time complexity is of order

$$N + \frac{N}{2} + \frac{N}{4} + \cdots + 2 = 2(N-1) \tag{2.4}$$

2.1.3 Performance when Operated as a Large Switch Fabric

The main advantage of the Benes Network is its low complexity compared to the other nonblocking networks. It can be rearrangably non-blocking while still maintain the number of crosspoints at the order of $N\log N$. However, the weakness of the Benes Network is lacking a fast control algorithm that can set up the paths between inputs and outputs. A central controller is also necessary for setup and changes the state of the crosspoints. When the size of the switch is doubled, the time needed for path setup and rearrangment is approximately doubled. This makes it less appealing for large fast-packet switching in which the paths taken by packets must be determined in a very short time interval.

2.2 Path Switching

The path switching is a quasi-static routing scheme [1], and it is a compromise of the static scheme and the dynamic scheme. The routing of path switching is based on the concept of virtual path within the Clos network. We consider that there is a virtual path between an input module and an output module, which comprises all virtual circuits interconnecting any incoming port and any outgoing port on this pair of modules.

2.2.1 Basic Concept of Path Switching

In the first part of the scheduling of path switching, the capacity required for each virtual path is determined by the traffic statistics among all pairs of input and output modules so that the QoS on these pairs can be satisfied. Then, a regular bipartite multigraph is generated according to the capacity assignment matrix. This regular bipartite multigraph can be decomposed into several smaller ones with degree equal to the number of central modules, by a time-space interleaving principle. A particular connection pattern in the middle stage of the Clos network can be represented by a regular bipartite multigraph if we consider each input module and each output module as a node. As a

result, these bipartite multigraphs generated can be used to setup the connections in the middle stage.

Unlike other schemes, which the scheduling and routing for all incoming packets are simultaneously processed by the central controller, the routing algorithm of path switching is implemented in a distributed manner over three different stages of the Clos network. The route assignments in the central modules are stored in the local memory. For every input module, the connection pattern of the central stage is known in every time slot. Each connection pattern specifies exactly how many packets can be delivered to a particular output module through which central modules in that time slot. Based on this routing information, each input module can select those packets queueing in the buffers according to their destinations and priorities. The output modules need to handle the output port contention problems. If the switch is operated repeatedly according to a set of connection patterns, then the capacity requirement on each virtual path can be satisfied in the long run, and the computation of route assignment on the fly can be avoided.

2.2.2 Capacity Allocation and Route Assignment

The scheduling of path switching consists of two steps, the capacity allocation and the route assignment. The capacity allocation is to find

the capacity $C_{ij} > \lambda_{ij}$ for each virtual path P_{ij} between input module I_i and output module O_j, where λ_{ij} denotes the aggregated bandwidth requirement of the virtual path P_{ij} in the unit of packets per time slot. The step can be carried out by optimizing some objective function subject to $\sum_i C_{ij} = \sum_j C_{ij} = m$, where m is the number of central modules. The choice of the objective function depends on the stochastic of the traffic on virtual paths and the quality of service requirements of the connections. If each virtual path is modeled as an independent $M/M/1$ queue with arrival rate λ_{ij} and service rate C_{ij} for all i, j; then the average delay for the packets from input module i to output module j is given by $T_{ij} = \frac{1}{C_{ij} - \lambda_{ij}}$. The objective function is minimization of the total weighted delay

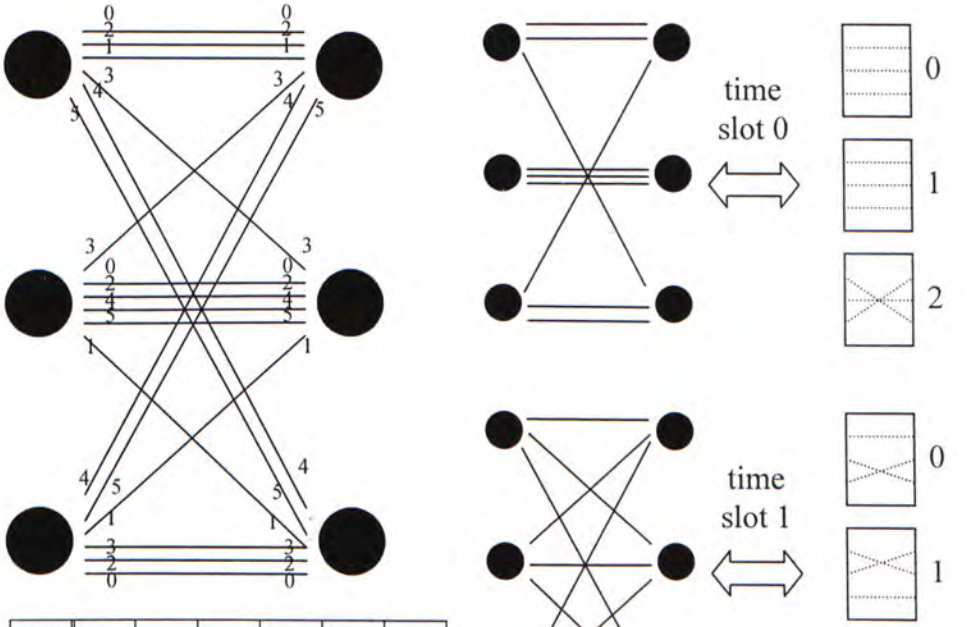
$$z = \sum_{i,j} \lambda_{ij} \cdot T_{ij} . \tag{2.5}$$

The next step is to multiply the capacity matrix $[C_{ij}]$ with a sufficient large integer f such that fC_{ij} are integers for all i, j. By considering each input and output module as a node, a regular bipartitie multigraph, called capacity graph is formed. An edge coloring of a bipartite multigraph is to assign m distinct colors to m edges of each node such that no two adjacent edges have the same color. It is well known that a regular bipartite multigraph with degree m is m-colorable. The capacity graph is with degree fm, so it can be edge colored by fm colors.

Any edge coloring of the capacity graph with degree fm is the superposition of the edge coloring of f regular bipartite multigraphs of degree m. Let $a \in \{0,1,\ldots, fm-1\}$ be the color number and

$$a = r \cdot f + t \tag{2.6}$$

where $r \in \{0,1,\ldots, m-1\}$ and $t \in \{0,1,\ldots, f-1\}$ are the quotient and the remainder of dividing a by f, i.e. $r = \left\lfloor \frac{a}{f} \right\rfloor$ and $t = a \bmod f$. The color assignment a of the edge between I_i and O_j indicates that the central module r has been assigned to a route from I_i to O_j in the tth time slot of every cycle. This time-space interleaving relation is illustrated in Figure 2.3, where $m = 3$ and $f = 2$.



a	0	1	2	3	4	5
r	0	0	1	1	2	2
t	0	1	0	1	0	1

Figure 2.3: Illustration of time-space interleaving principle

Consider a 9 × 9 3-stage Clos network with 3 modules in each stage. Let the rate matrix be

$$\lambda = \begin{bmatrix} 1.0 & 0.3 & 0.7 \\ 0.4 & 1.4 & 0.3 \\ 0.7 & 0.5 & 1.0 \end{bmatrix} \tag{2.7}$$

the capacity assignment matrix calculated by minimization of the total weight delay $\sum_{i,j} \frac{\lambda_{ij}}{C_{ij} - \lambda_{ij}}$ is

$$C = \begin{bmatrix} 1.4 & 0.5 & 1.1 \\ 0.6 & 1.8 & 0.6 \\ 1.0 & 0.7 & 1.3 \end{bmatrix} \approx \begin{bmatrix} \frac{3}{2} & \frac{1}{2} & \frac{2}{2} \\ \frac{1}{2} & \frac{4}{2} & \frac{1}{2} \\ \frac{2}{2} & \frac{1}{2} & \frac{3}{2} \end{bmatrix}. \tag{2.8}$$

The capacity assignment matrix is non-integer. However, the product $f \cdot C$ can be rounded off into integers, and the round off error is inversely proportional to f. The error can be arbitrarily small if the frame size f is sufficiently large. However, since the amount of routing information stored in the memory is linearly proportional to f, the access speed and the memory space of the input modules limits f. In this example, f is chose to be 2. The capacity graph and the time-space interleaving relation are shown in Figure 2.3.

Figure 2.4 gives the resultant connection pattern of the given rate matrix. It is easy to verify that the number of central modules assigned for each input-output pair satisfies the given matrix if the patterns are used repeatedly.

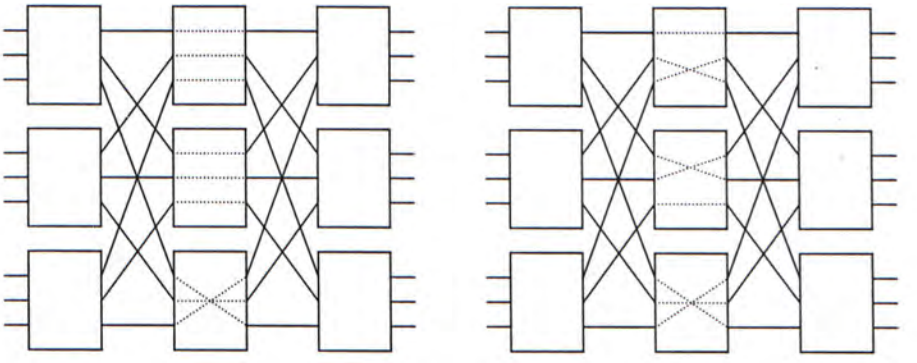


Figure 2.4: Route scheduling in central modules for the example.
(a) Time slot 0. (b) Time slot 1.

Chapter 3

Path Switching over Benes Network

The previous chapter showed that the Benes network lacks a fast routing algorithm. This makes it not suitable for fast-packing switching since the paths taken by the packets cannot be calculated on the fly. In order to solve this problem, we can develop a fast routing algorithm for Benes network. However, this solution cannot be achieved easily. Alternatively, we can find a method to prevent the path hunting on the fly, and this can be done with path switching. In path switching, all the paths between the inputs and outputs are found previously and used repeatedly, while the capacity allocated to all the connection requests can still be satisfied.

3.1 The Model of path-switched Benes Network

Path switching can be implemented to Benes network in two kinds of different manner. The first one is to use the similarity on the structure of the Benes network and the Clos network, and then directly apply the procedure of the path switching. This is called the module-to-module path-switched Benes network. The other one is to use the idea of path switching and then apply the scheme to each port of the Benes network. This is called the port-to-port path-switched Benes network. The module-to-module implementation is simpler, however, the port-to-port implementation can guarantees bandwidth requirement for each connection.

3.2 Module-to-Module Implementation

Let's consider the structure of an $N \times N$ Benes network, there are $2\log_2 N$-1 stages, each stage has $N/2$ modules. The first and last stage of the Benes network is just equivalent to the corresponding stage of the three-stage Clos network. For the central stages, if we only concentrate on the input and output ports of the two subnetworks, they are just the same as two $N/2 \times N/2$ modules. As a result, the $N \times N$ Benes network

can be modelled as a three-stage Clos network while the first and last stages have $N/2$ 2 × 2 modules and the middle stage has two $N/2 \times N/2$ modules. We can then directly use the method introduced in chapter 2. In the following sections, we will discuss the roles of different stages in the module-to-module path-switched Benes network.

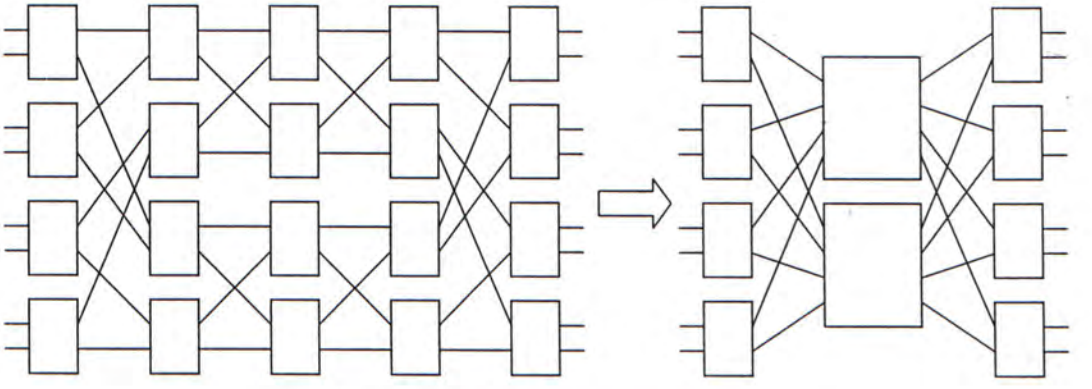


Figure 3.1: Model of the module-to-module path-switched Benes network. ($N = 8$)

3.2.1 The First Stage (Input Module)

Every module in the first stage stores a routing table in their local memory. This routing table records all the connection pattern of the middle stage. As a result, the input modules know which central module is connected to a particular output module in all the time slots. The look-ahead contention resolution scheme [12] is applied in our design, in which the contention resolution process during a time slot is divided into w cycles where w is the window size. In the first cycle, the packets at the heads of the input queues contend for those outputs that have not been seized in this time slot. In the second cycle, the scheme

looks at the second packets of the inputs that have lost contention in the first cycle to see if they are destined for any outputs that are as yet unclaimed. The process is repeated w times. This can improve the overall system throughput. The size of w depends on different applications, and the throughput is increased monotonically with an increase in w. Packets with different priorities can also be routed by the input module with a little bit change in the selection process. Under certain traffic constraints and regulation, the number of buffers of the input modules can be limited. This will be discussed in the next chapter.

3.2.2 The Middle Stage (Central Module)

A bipartite multigraph can be directly converted to a particular connection pattern in the middle stage. This connection pattern is end-to-end and is not suitable to be used in the middle stage directly. However, the middle stage is just another two Benes networks with half dimensions, so the connection pattern is equivalent to a complete list of input-output pairs without output contention. We can use the looping algorithm [16] to find out all the paths. Alternatively, the parallel algorithm for route assignments in Benes networks introduced in [18] can also be used. The time complexity of this algorithm is $O(\log^2 N)$ where N is the network size. Although this step is very time-consuming, it only needs to be done once and can be done in the call setup stage.

Once all the paths are found, they are stored in the local memory of the central modules and can be used repetitively.

3.2.3 The Last Stage (Output Module)

The input modules select the packet that the destination address matched with the desired output only. They do not care whether the other modules have the packet destined to the same output address. Therefore, each output module would have to handle the output port contention problems. The sizes of the output buffers directly affect the loss probability. Just the same as input modules, the number of buffers of the output modules can also be limited under certain traffic constraints.

The performance issues of the module-to-module implementation and the some simulation results are presented in chapter 4 and chapter 5 respectively.

3.3 Port-to-Port Implementation

Although all the procedures of the path switching can be directly applied on the module-to-module implementation, the bandwidth requirement of each connection cannot be guaranteed. In order to provide bandwidth guarantee for all the connections, the capacity allocation must be based on the port-to-port request matrix.

3.3.1 Uniform Traffic

The connection request matrix and the capacity assignment matrix of an $N \times N$ Benes network under uniform traffic is

$$\lambda = C = \begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix} \tag{3.1}$$

The frame size f can be chosen as the number of inputs N so that the product of frame and capacity matrix is a matrix with all elements equal to 1.

$$f \cdot C = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{3.2}$$

The route assignment can be easily found by the *Latin Square* given in the following table, where T_i is the ith time slot of a frame.

	O_0	O_1	O_2	$\cdots$	O_{N-1}
I_0	T_0	T_1	T_2	$\cdots$	T_{N-1}
I_1	T_{N-1}	T_0	T_1	$\cdots$	T_{N-2}
I_2	T_{N-2}	T_{N-1}	T_0	$\cdots$	T_{N-3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_{N-1}	T_1	T_2	T_3	$\cdots$	T_0

Table 3.1: Route assignment by *Latin Square* for uniform traffic

Table 3.2 shows the input-output pairs of an 8 × 8 Benes network in different time slots.

	output							
input	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

Table 3.2: Connection patterns for uniform traffic (8 × 8)

3.3.2 Multirate Traffic

In this section, we use an example to illustrate the implementation of port-to-port path-switched Benes network under multirate traffic. A set of connection request is given in the form of (x, y, r) where x is the input port, y is the output port and r is the fraction of bandwidth required by the connection.

(1, 4, 0.4), (1, 7, 0.4), (2, 3, 0.7), (3, 1, 0.4), (3, 4, 0.2), (3, 6, 0.1), (4, 8, 0.9), (5, 2, 0.7), (6, 3, 0.2), (6, 5, 0.5), (7, 1, 0.3), (7, 5, 0.5), (7, 7, 0.2), (8, 6, 0.8).

The rate matrix is

$$\lambda = \begin{bmatrix} 0 & 0 & 0 & 0.4 & 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0.2 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 \\ 0 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.5 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 & 0.4 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 \end{bmatrix} \tag{3.3}$$

Assume each virtual path as an independent $M/M/1$ queue with arrival rate λ_{ij} and service rate C_{ij}. Then the capacity assignment matrix is calculated by minimizing the objective function

$$z = \sum_{i,j} \frac{\lambda_{ij}}{C_{ij} - \lambda_{ij}} \tag{3.4}$$

subject to the following constraints:

$$\begin{cases} C_{ij} > \lambda_{ij} \\ \sum_i C_{ij} = 1 \\ \sum_j C_{ij} = 1 \end{cases} \tag{3.5}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0.6 & 0 & 0 & 0.4 & 0 \\ 0.2 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0.4 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.6 & 0 & 0.2 & 0 \\ 0.4 & 0 & 0 & 0 & 0.4 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 & 0 \end{bmatrix} \tag{3.6}$$

The resulting capacity assignment matrix is then multipled by an integer f so every element of the matrix can be rounded off into an integer.

Let f equals to 5, we have

$$f \cdot C = \begin{bmatrix} 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 \\ 1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 1 & 0 \end{bmatrix} \tag{3.7}$$

The remaining process is to form a regular bipartite multigraph by letting the input ports and the output ports as nodes and then decomposes it into f bipartite multigraphs each with degree equals to 1. Alternatively, we can use the following algorithm to obtain f matrices, the value 1 at row i and column j indicates that there is a connection between input port i and output port j.

1. *for frame f*
2. *for each unmarked row i*
   count the number of non-zero and unmarked element
3. *find the minimum count*
4. *mark the corresponding row i and column j*
5. T_{ij} = 1 (T_{ij} *is the connection pattern of this time slot)*
6. *repeat 2 - 5 until all rows are marked*
7. *C = C - T*
8. *repeat 1 - 7 for all the frame*

The input-output pairs of the Benes network in different time slots calculated by the above algorithm are shown in table 3.3.

	output				
input	T_0	T_1	T_2	T_3	T_4
1	4	4	4	7	7
2	1	3	3	3	3
3	6	1	1	4	4
4	8	8	8	8	8
5	2	2	2	2	2
6	3	5	7	5	5
7	5	7	5	1	1
8	7	6	6	6	6

Table 3.3: Connection patterns for the multirate traffic example

3.4 Closing remarks

The implementation of path switching is completely distributed. Although the computation of capacity assignment and route assignment by the central controller still needs global information, it is not a slot-by-slot process. The routing tables stored in the local memory of the input modules would be updated only if the traffic matrix changes significantly and the switch performance becomes unacceptable.

The port-to-port implementation can guarantee bandwidth requirements for multirate circuit switching while the implementation of the module-to-module approach is directly and simpler.

Chapter 4

Performance Analysis

In this chapter, we will follow a graphical model in [15] to study the provision of deterministic QoS guarantees at the path-switched Benes network for each data session. We will first introduce the arrival curve and the service curve of a data session. We then show the effect of central stage token assignment on the service guarantees provided to each data session. Finally, we establish the upper bound on the delay, input backlog and output backlog at the Benes network for each data session, assuming each input traffic stream is under leaky-bucket rate control and there is no packet loss due to buffer overflow.

4.1 Traffic Constraints and Performance Guarantees

In this section, we first define the notion of arrival curves and service curves, then we present maximum delay and backlog guarantees for a session when its traffic is constrained.

Let $R_i^{in}(r, t)$ be the amount of traffic from session i arrived in time interval $(r, t]$. We assume $R_i^{in}(r, t)$ conforms to a burstiness constraint (σ_i, ρ_i, C_i), $R_i^{in}(r, t) \sim (\sigma_i, \rho_i, C_i)$. The parameter C_i is the peak rate of session i, ρ_i is an upper bound on the long term average rate of flow of the session i traffic, and σ_i, the burstiness parameter, represents the maximum amount of session i traffic that can arrive in a burst. If $R_i^{in}(r, t)$ is fed to a fictitious work conserving server that operates at a fixed rate ρ_i without shared by the other sessions, the number of packets that are stored in the server will never be larger than σ_i. For example, let $\sigma_i = 10$, $\rho_i = 1$, $C_i = 5$, the burst length is 2.5s. After the 1st second, 4 packets are left in the server and 8 packets are left in the next second. In the following 0.5 second, 2.5 packets are arrived and the server can process 0.5 packet, afterwards, only 1 packet arrive per second, so the total number of packets will not be larger than 10. The leaky-bucket shaped

traffic [4] conforms to this type of constraints, where σ_i is the bucket size, ρ_i is the rate of water flowing into the bucket and C_i is the maximum rate of water flowing out of the bucket.

We now suppose that session i traffic is fed to a network element which may shared by other sessions. Let $R_i^{\text{out}}(r, t)$ be the amount of session i traffic output from it during the interval $(r, t]$.

4.1.1 Arrival Curve and Service Curve

To characterize the incoming traffic constraints, we define a non-decreasing function $A_i(\cdot)$ as follow:

Definition 4.1 *Define a non-decreasing function $A_i(\cdot)$ as*

$$A_i(u) = \min\{C_i u, \sigma_i + \rho_i u\}. \tag{4.1}$$

$A_i(\cdot)$ is called an arrival curve, which specifies the maximum amount of traffic received form a session with burstiness constraint during an interval $(r, r + u]$.

$A_i(\cdot)$ is equivalent to rate of water flowing out from a full leaky-bucket. After $t = \dfrac{\sigma_i}{C_i - \rho_i}$ seconds, the bucket will become empty. In the burst period, the rate of flow of water is C_i and will decreases to ρ_i subsequently. Therefore, we have

$$A_i(t-r) \geq R_i^{\text{in}}(r,t) \text{ for all } t \geq r \tag{4.2}$$

Figure 4.1 illustrates an arrival curve for a burstiness constraint (σ_i, ρ_i, C_i) within the context of peak rate, average rate, and maximum burst length.

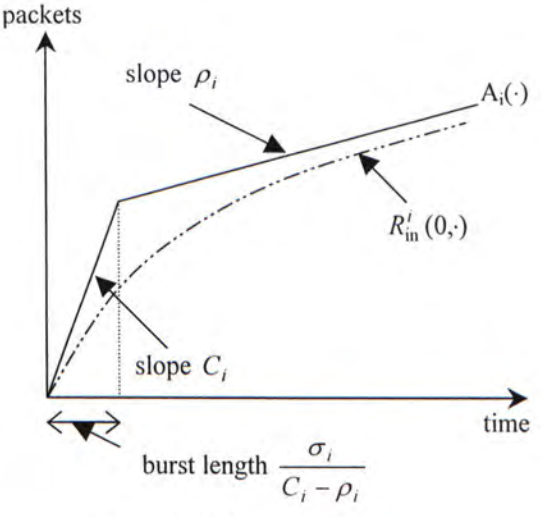


Figure 4.1: An arrival curve.

To characterize the service traffic constraints, we define a non-decreasing function $S_i(\cdot)$ as follow:

Definition 4.2 *Define a non-decreasing function $S_i(\cdot)$ with $S_i(0) = 0$. If for any t, there exists $r \leq t$ such that*

$$R_i^{\text{out}}(r,t) \geq S_i(t-r), \tag{4.3}$$

and there is no session i packet stored in the network element at time r, then the network element guarantees session i a service curve of S_i.

Service curve represents the least amount of service provided by the network element to a data session during its busy period. Figure 4.2

shows a service curve S_i for session i in its busy period. The service curve is a straight line with slope g_i, which is the service rate reserved for session i at the network element. For stability, g_i must be greater than or equal to ρ_i.

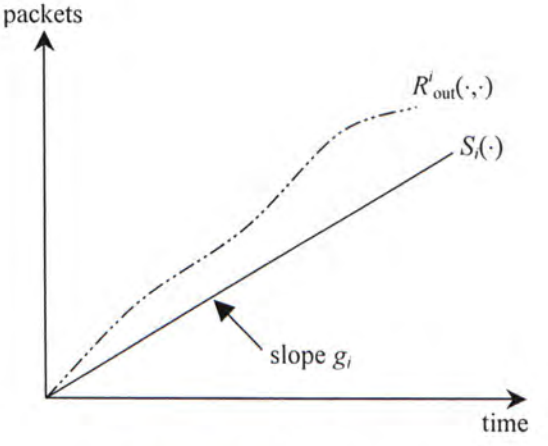


Figure 4.2: A service curve.

We now introduce another parameter θ_i, it is the latency of the service curve. From time 0 to θ_i, session i only receive zero service from the network element.

$$S_i(u) = \begin{cases} g_i(u-\theta_i) & \text{if } u > \theta_i \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

for any $\theta_i \geq 0$ and $g_i \geq 0$. In this case, we say that network element guarantees session i a service of (θ_i, g_i).

There is a general class of servers, called Latency-Rate servers (LR servers) [9], which the behavior is determined by two parameters, the latency θ and the allocated rate g.

The latency of the LR server is the worse case delay seen by the first packet of the busy period of a session. As a result, these kind of servers can guarantee each data session a service of (θ, g). All work-conserving schedulers that provide bandwidth guarantees, such as Weighted Fair Queueing (PGPS) [10], Virtual Clock [17], Self-Clock Fair Queueing [19], Weighted Round Robin [26], and Deficit Round Robin [21], offering this property and therefore belong to this class. We assume the servers in our network belong to the LR servers and guarantee a service of (θ, g) for a session.

4.1.2 Delay Bound and Backlog Bound

Let's consider an input traffic stream R_i^{in}, which conforms to a burstiness constraint (σ_i, ρ_i, C_i), to a network element that guarantees a service curve S_i of (θ_i, g_i). We assume that a packet from session i arrives the network element at time t and departs from it at time $t + d(t)$. $d(t)$ is the delay that the packet experienced in the network element. We also assume that the network element is empty at time zero and session i packets are in a first-in-first-out (FIFO) order. We have

$$R_i^{\text{in}}(0,t) = R_i^{\text{out}}(0,t + d(t)), \tag{4.5}$$

where $R_i^{\text{in}}(0, t)$ is the amount of session i traffic received by the network element during the interval $(0, t]$ and $R_i^{\text{out}}(0, t + d(t))$ is the amount of session i traffic departed from it within the interval $(0, t + d(t)]$.

From the definition 4.2, there exists a maximum $r \le t + d(t)$ such that there is no session *i* packets stored in the network element at time *r*, and

$$R_i^{\text{out}}(r, t + d(t)) \ge S_i(t + d(t) - r). \tag{4.6}$$

Thus,

$$\begin{aligned} R_i^{\text{in}}(0,t) &= R_i^{\text{out}}(0, t + d(t)) \\ &= R_i^{\text{out}}(0,r) + R_i^{\text{out}}(r, t + d(t)) \\ &\ge R_i^{\text{out}}(0,r) + S_i(t + d(t) - r). \end{aligned} \tag{4.7}$$

As there is no session *i* packet stored in the network element at time *r*, the arrival rate is equal to the departure rate,

$$R_i^{\text{in}}(0,r) = R_i^{\text{out}}(0,r). \tag{4.8}$$

Substitute (4.8) into (4.7), we have,

$$\begin{aligned} R_i^{\text{in}}(0,t) &\ge R_i^{\text{in}}(0,r) + S_i(t + d(t) - r) \\ R_i^{\text{in}}(0,t) - R_i^{\text{in}}(0,r) &\ge S_i(t + d(t) - r) \\ R_i^{\text{in}}(r,t) &\ge S_i(t + d(t) - r) \end{aligned} \tag{4.9}$$

Since session *i* traffic conforms to the burstiness constraint such that

$$A_i(t - r) \ge R_i^{\text{in}}(r,t), \tag{4.10}$$

we have

$$A_i(t - r) \ge S_i(t + d(t) - r). \tag{4.11}$$

The delay experienced by a packet is the horizontal distance between the arrival curve and the service curve. There exists a non-negative

number Δ such that $A_i(t-r) \le S_i(t+\Delta-r)$, the minimum value of Δ is the delay encountered by the packet.

In this way, the delay $d(t)$ encountered by a session i packet at the network element is upper bound by

$$\begin{aligned} d(t) &\le \min\{\Delta : \Delta \ge 0, A_i(t-r) \le S_i(t+\Delta-r)\} \\ &\le \max_{u:u\ge 0} \min\{\Delta : \Delta \ge 0, A_i(u) \le S_i(u+\Delta)\} \end{aligned} \tag{4.12}$$

The session i backlog at the network element is the vertical distance between the arrival curve and the service curve. Let $B_i(t)$ be the total number of session i packets stored in the network element at time t. There exists a maximum $r \le t$ such that $B_i(r) = 0$ and $R^i_{\text{out}}(r,t) \ge S_i(t-r)$ as the network element guarantees a service curve S_i to session i.

$$\begin{aligned} B_i(t) &= R_i^{\text{in}}(0,t) - R_i^{\text{out}}(0,t) \\ &= R_i^{\text{in}}(0,t) - R_i^{\text{out}}(0,r) - R_i^{\text{out}}(r,t) \\ &\le R_i^{\text{in}}(0,t) - R_i^{\text{out}}(0,r) - S_i(t-r) \\ &= R_i^{\text{in}}(r,t) - S_i(t-r) \qquad \because R_i^{\text{in}}(0,r) = R_i^{\text{out}}(0,r) \\ &\le A_i(t-r) - S_i(t-r) \\ &\le \max_{u:u\ge 0}\{A_i(u) - S_i(u)\} \end{aligned} \tag{4.13}$$

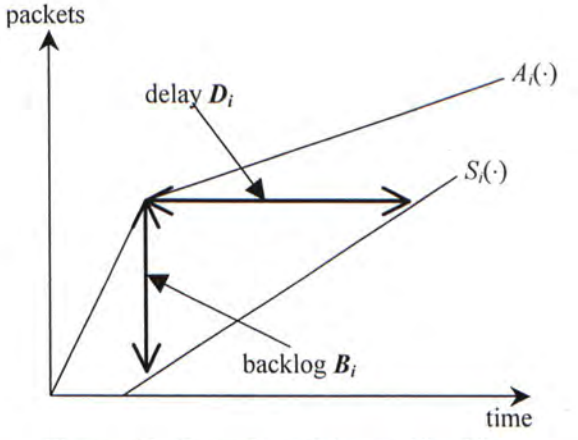


Figure 4.3: Bound on delay and backlog.

Given an arrival curve A_i and a service curve S_i for session i, the upper bound D_i on delay is the maximum horizontal distance between the curves, while the upper bound B_i on backlog is the maximum vertical distance between the curves, as shown in Figure 4.3. This gives us a straightforward method to calculate the deterministic bound for any network element if their incoming traffic is constrained and the service is guaranteed. In the following sections, we will establish the bounds on delay and backlog at the path-switched Benes network.

4.2 Service Guarantees

Suppose that session i traffic passes the Benes network via an input module I_x and output module O_y. The input module guarantees a service curve S_i^{in} of $(\theta_i^{\text{in}}, g_i^{\text{in}})$ and the output module guarantees a service curve S_i^{out} of $(\theta_i^{\text{out}}, g_i^{\text{out}})$.

If we consider the central module connected to a particular output as a token to this output, then the input modules may have tokens to different output in different time slot. A packet wants to depart from the input module must have a token from this input module which matched with the destined output module.

Connection pattern at central module is periodic with frame size *f* so the token assignment for a specific output module repeats for every *f* time slots. Recall the example in chapter 3.3, the number of tokens from input module 0 to output module 0 is 2, 1 and 0 in time slot 0, 1 and 2 respectively. However, due to the different frame alignment, the number of tokens for time slot 0, 1 and 2 may become 1, 0, 2 or 0, 1, 2. The number of packets transmitted under these three alignments is plotted in Figure 4.4.

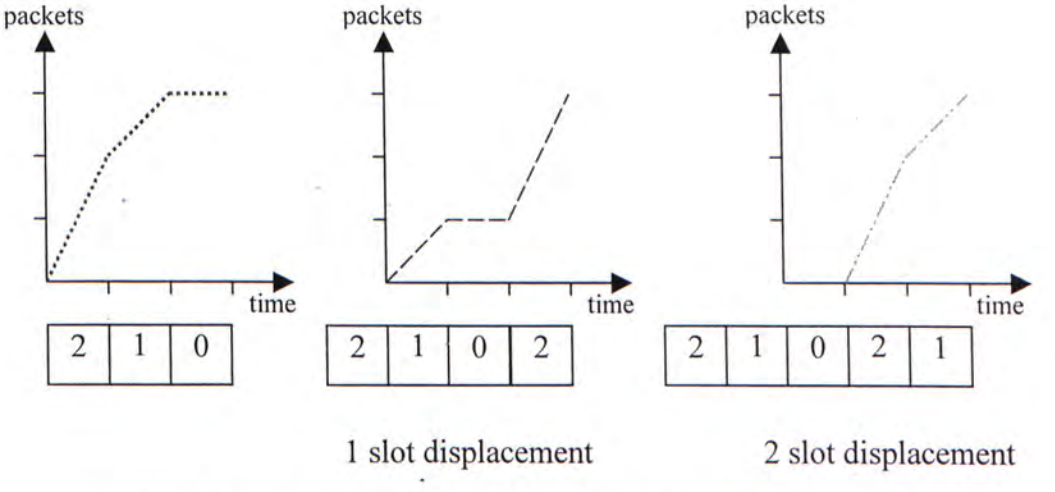


Figure 4.4: Token assignment at the input module with different frame alignment.

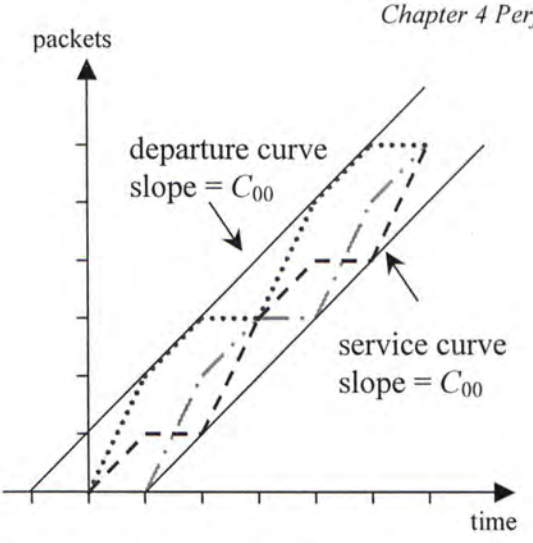


Figure 4.5: Service curve and departure curve for periodic token assignment scheme.

In Figure 4.5, the three curves are plotted on the same graph and a straight line that lower bounds the curves is also drawn. This straight line can be regarded as the service curve for the virtual path connecting input module 0 and output module 0. It is noticed that the slope of this service curve is equal to 1, which is equal to the corresponding value of the capacity assignment matrix (C_{00}), and has a one-slot displacement to the right hand side. The displacement v_{xy} of the service curve represents an extra delay that a packet experiences at the input module I_x in the worst case scenario. This horizontal displacement is due to the uneven token distribution regardless of the service discipline of the input module. Different virtual paths may have different token distribution and therefore may encounter different extent of displacement. Therefore, v_{xy} can be considered as the additional latency to the service guarantees provided the server at input module. The resultant service curve guaranteed by the input module to session i traffic is of $(\theta_i^{\text{in}} + v_{xy}, g_i^{\text{in}})$.

If we consider the upper bound of these three packet transmission curves, another straight line with the same slope can be drawn. This is the departure curve S^{d}_{xy} of the virtual path between I_x and O_y. It upper bounds the amount of packet delivered between input module I_x and output module O_y in any interval starting from time zero. The horizontal distance between the departure curve and the time zero is equal to that of the service curve. For this case, input module 0 and output module 0, the distance is one slot.

4.3 Deterministic Bounds

4.3.1 Delay

To compute the upper bound on the delay for session i traffic, we consider the switch as two elements in tandem. The first one is the input module I_x and the second one is the output module O_y. In the previous section, we showed that session i traffic receives a service guarantee of $(\theta_i^{\text{in}} + v_{xy}, g_i^{\text{in}})$ at input module I_x and a service guarantee of $(\theta_i^{\text{out}}, g_i^{\text{out}})$ at output module O_y. In [15], it has been shown that the service curve S_i^{out} is in the form of

$$S_i^{\text{out}}(t-r) = \max(g_i^{\text{out}}(t-r-\theta_i^{\text{out}} - \theta_i^{\text{in}} - v_{xy}),0), \qquad (4.14)$$

where $r \leq t$.

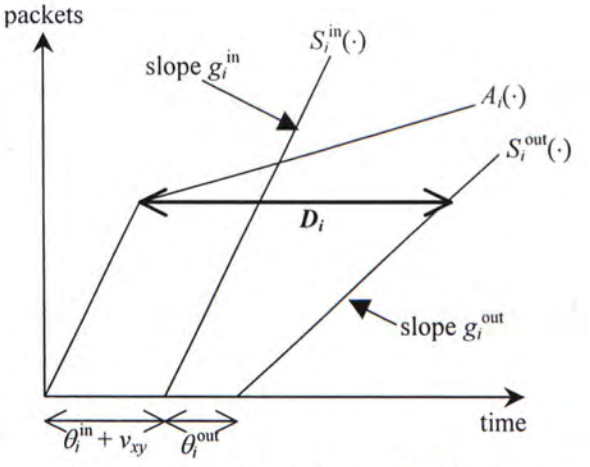


Figure 4.6: Delay bound at path-switched Benes network.

Figure 4.6 shows the arrival curve A_i for session i and the service curve S_i^{in} and S_i^{out}. The maximum delay D_i encountered by session i packets is the maximum horizontal distance between the arrival curve A_i and the service curve S_i^{out}.

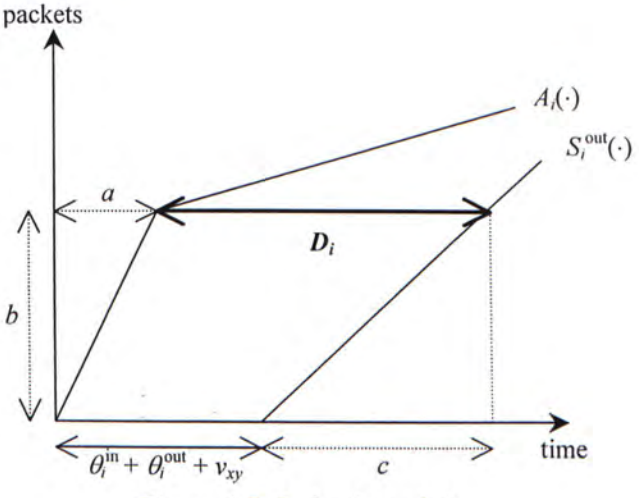


Figure 4.7: Delay bound $\boldsymbol{D_i}$.

From Figure 4.7, we can see that

$$D_i = \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + c - a \tag{4.15}$$

a is the burst length and equals to $\frac{\sigma_i}{C_i - \rho_i}$

By simple geometry,

$$g_i^{\text{out}} = \frac{b}{c} \text{ and } C_i = \frac{b}{a}$$

$$\therefore c = \frac{aC_i}{g_i^{\text{out}}} \tag{4.16}$$

Substitute (4.16) into (4.15),

$$\begin{aligned} D_i &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + \frac{aC_i}{g_i^{\text{out}}} - a \\ &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + a\left(\frac{C_i}{g_i^{\text{out}}} - 1\right) \\ &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + \frac{\sigma_i}{C_i - \rho_i}\left(\frac{C_i}{g_i^{\text{out}}} - 1\right) \end{aligned} \tag{4.17}$$

4.3.2 Backlog at Input Module

The maximum amount of session i traffic B_i^{in} stored at the input module I_x is the maximum vertical distance between the arrival curve A_i and the input module service curve S_i^{in}. However, due to the variation in burst length ($\frac{\sigma_i}{C_i - \rho_i}$), latency ($\theta_i^{\text{in}} + v_{xy}$), burst arrival rate ($C_i$) and service rate ($g_i^{\text{in}}$), the arrival and service curves will appear differently and this

$$D_i = \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + c - a \tag{4.15}$$

a is the burst length and equals to $\frac{\sigma_i}{C_i - \rho_i}$

By simple geometry,

$$g_i^{\text{out}} = \frac{b}{c} \text{ and } C_i = \frac{b}{a}$$

$$\therefore c = \frac{aC_i}{g_i^{\text{out}}} \tag{4.16}$$

Substitute (4.16) into (4.15),

$$\begin{aligned} D_i &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + \frac{aC_i}{g_i^{\text{out}}} - a \\ &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + a\left(\frac{C_i}{g_i^{\text{out}}} - 1\right) \\ &= \theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy} + \frac{\sigma_i}{C_i - \rho_i}\left(\frac{C_i}{g_i^{\text{out}}} - 1\right) \end{aligned} \tag{4.17}$$

4.3.2 Backlog at Input Module

The maximum amount of session i traffic B_i^{in} stored at the input module I_x is the maximum vertical distance between the arrival curve A_i and the input module service curve S_i^{in}. However, due to the variation in burst length ($\frac{\sigma_i}{C_i - \rho_i}$), latency ($\theta_i^{\text{in}} + v_{xy}$), burst arrival rate ($C_i$) and service rate ($g_i^{\text{in}}$), the arrival and service curves will appear differently and this

will affect the maximum vertical distance between them. There are totally 4 cases, which are summarize in the following table.

	Latency < burst length	Latency ≥ burst length
burst rate > service rate	Case I, Figure 4.8 (a)	Case II, Figure 4.8 (b)
burst rate ≤ service rate	Case III, Figure 4.8 (c)	Case IV, Figure 4.8 (d)

Table 4.1: Different cases on the upper bound of input backlog.

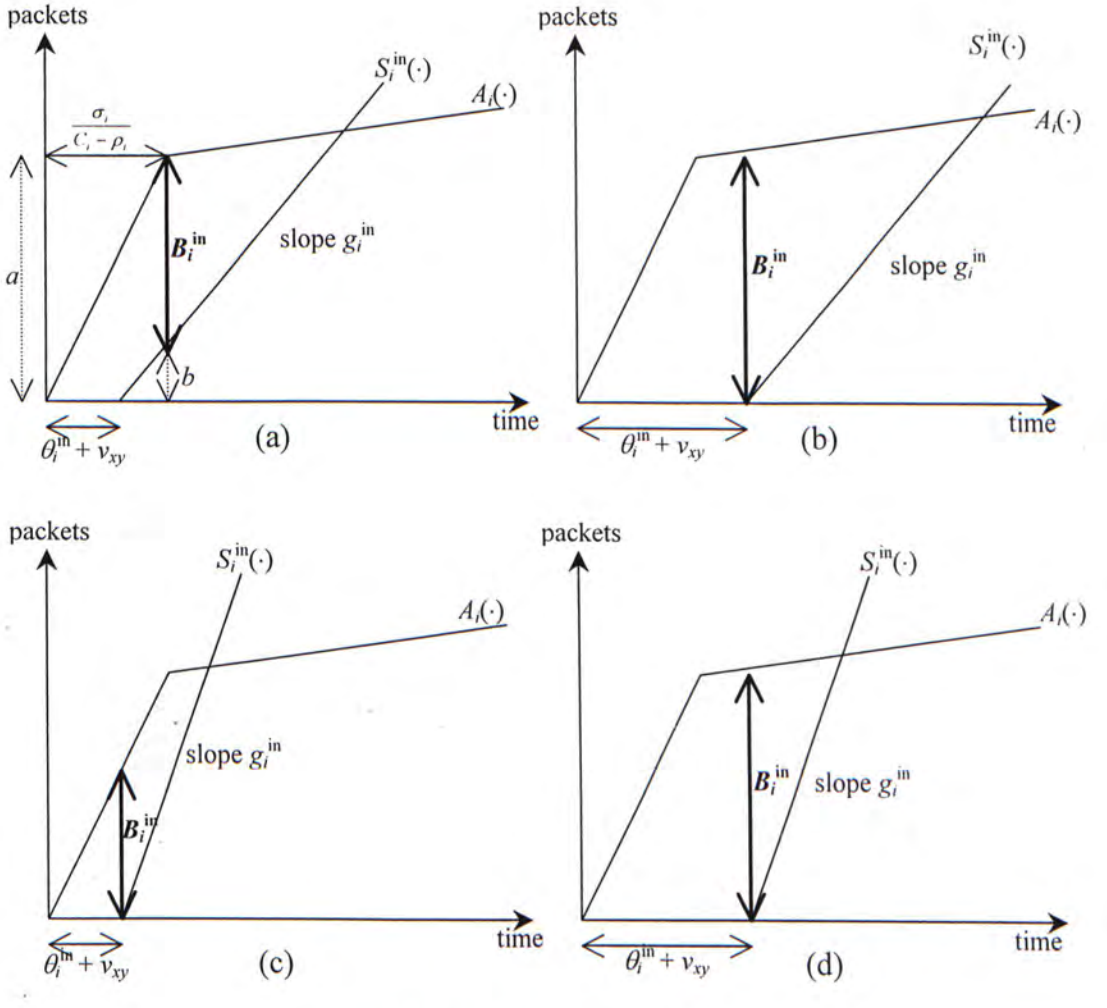


Figure 4.8: Backlog bounds for input module.
(a) Case I. (b) Case II. (c) Case III. (d) Case IV

For case I,

$$\begin{aligned} B_i^{\text{in}} &= a - b \\ &= \frac{\sigma_i}{C_i - \rho_i} C_i - \left(\frac{\sigma_i}{C_i - \rho_i} - \theta_i^{\text{in}} - v_{xy} \right) g_i^{\text{in}} \\ &= \frac{\sigma_i}{C_i - \rho_i} (C_i - g_i^{\text{in}}) + g_i^{\text{in}} (\theta_i^{\text{in}} + v_{xy}) \\ &= (\theta_i^{\text{in}} + v_{xy}) g_i^{\text{in}} + \frac{C_i - g_i^{\text{in}}}{C_i - \rho_i} \sigma_i \end{aligned} \tag{4.18}$$

For case II to IV, B_i^{in} is the height of arrival curve at the time $\theta_i^{\text{in}} + v_{xy}$, we can just substitute $\theta_i^{\text{in}} + v_{xy}$ into the arrival curve formula to get the result.

Case II:

$$B_i^{\text{in}} = (\theta_i^{\text{in}} + v_{xy}) \rho_i + \sigma_i \tag{4.19}$$

Case III:

$$B_i^{\text{in}} = (\theta_i^{\text{in}} + v_{xy}) C_i \tag{4.20}$$

Case IV:

$$B_i^{\text{in}} = (\theta_i^{\text{in}} + v_{xy}) \rho_i + \sigma_i \tag{4.21}$$

Summarize all the cases, we have

$$B_i^{\text{in}} = \begin{cases} (\theta_i^{\text{in}} + v_{xy})g_i^{\text{in}} + \dfrac{C_i - g_i^{\text{in}}}{C_i - \rho_i}\sigma_i & \text{if} \theta_i^{\text{in}} + v_{xy} < \dfrac{\sigma_i}{C_i - \rho_i} \text{and} C_i > g_i^{\text{in}} \\ (\theta_i^{\text{in}} + v_{xy})C_i & \text{if} \theta_i^{\text{in}} + v_{xy} < \dfrac{\sigma_i}{C_i - \rho_i} \text{and} C_i \le g_i^{\text{in}} \\ (\theta_i^{\text{in}} + v_{xy})\rho_i + \sigma_i & \text{otherwise} \end{cases} \tag{4.22}$$

4.3.3 Backlog at Output Module

Let $R_i^{\text{xy}}(r, t)$ be the amount of session i traffic from input module I_x to output module O_y in time interval $(r, t]$. In [15], it shows that the traffic stream R_i^{xy} conforms to the burstiness constraint

$$(\sigma_i + \rho_i(\theta_i^{\text{in}} + v_{xy}), \rho_i, \infty).$$

From the definition of arrival and service constraints, we have

$$A_i^{\text{in}}(t-r) \ge R_i^{\text{xy}}(r,t) \ge S_i^{\text{out}}(t-r) \tag{4.23}$$

where

$$A_i^{\text{in}}(u) = (\sigma_i + \rho_i(\theta_i^{\text{in}} + v_{xy}) + \rho_i u) \tag{4.24}$$

As the departure curve S^d_{xy} introduced in the previous section upper bounds the amount of packets transmitted from input module I_x to output module O_y, $R_i^{\text{xy}}(r, t)$ is also upper bounded by S^d_{xy}. Figure 4.9 shows the arrival curve for R_i^{xy} and the service curve S_i^{out}, noted that the upper bound on backlog at the output module is the maximum vertical distance between the curves $\min\{A_i^{\text{in}}(t), S^d_{xy}(t)\}$ and $S_i^{\text{out}}(t)$.

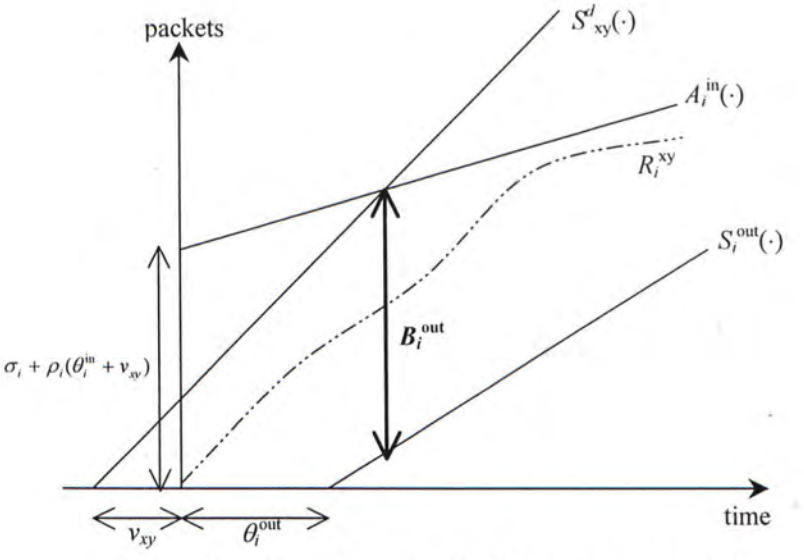


Figure 4.9: Upper bound on backlog at output module.

There are two cases for the curves of the output module, which depends on whether the departure curve intersects with the arrival curve or not. Let's consider the case when they intersect (Figure 4.10).

The y-intercept of the departure curve S^d_{xy} must be smaller than $\sigma_i + \rho_i(\theta_i^{in} + v_{xy})$ in order to cut the arrival curve A_i^{in}. This gives

$$C_{xy}v_{xy} < \sigma_i + \rho_i(\theta_i^{in} + v_{xy})$$

$$v_{xy} < \frac{\sigma_i + \rho_i\theta_i^{in}}{C_{xy} - \rho_i} \tag{4.25}$$

Let w be the horizontal distance of the intersection. The maximum amount of session i traffic B_i^{out} stored at the output module O_y depends on w. If $w \le \theta_i^{out}$ (Figure 4.10(a)), B_i^{out} is the vertical distance between the curve A_i^{in} and S_i^{out} at time θ_i^{out}. Thus,

$$B_i^{\text{out}} = \sigma_i + \rho_i(\theta_i^{\text{in}} + v_{xy}) + \rho_i\theta_i^{\text{out}}$$
$$= \sigma_i + \rho_i(\theta_i^{\text{in}} + \theta_i^{\text{out}} + v_{xy}) \qquad (4.26)$$

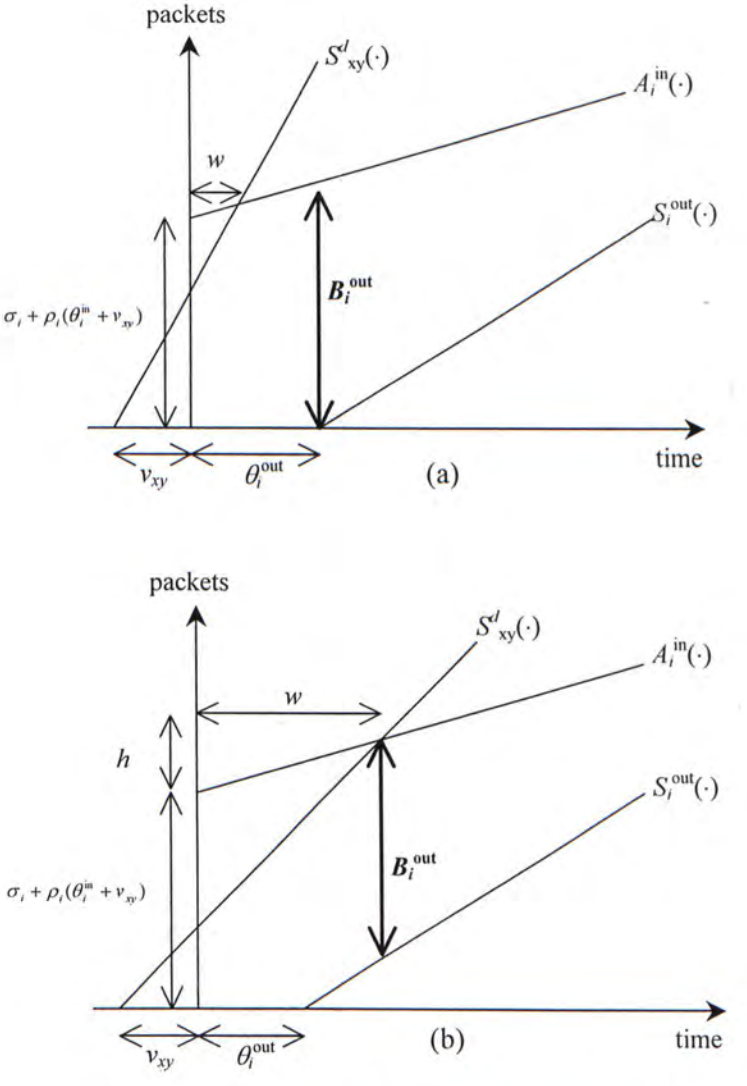


Figure 4.10: Backlog bounds for output module when departure curve intersects with arrival curve.
(a) $w \le \theta_i^{\text{out}}$. (b) $w > \theta_i^{\text{out}}$.

When $w > \theta_i^{\text{out}}$,

$$B_i^{\text{out}} = \sigma_i + \rho_i(\theta_i^{\text{in}} + v_{xy}) + \rho_i w - g_i^{\text{out}}(w - \theta_i^{\text{out}}) \qquad (4.27)$$

From Figure 4.10(b),

$$\frac{h+\sigma_i+\rho_i(\theta_i^{\text{in}}+v_{xy})}{w+v_{xy}}=C_{xy}, \tag{4.28}$$

and

$$h=w\rho_i \tag{4.29}$$

Substitute (4.29) into (4.28),

$$w\rho_i+\sigma_i+\rho_i(\theta_i^{\text{in}}+v_{xy})=C_{xy}(w+v_{xy})$$

$$w(C_{xy}-\rho_i)=\sigma_i+\rho_i(\theta_i^{\text{in}}+v_{xy})-C_{xy}v_{xy}$$

$$w=\frac{\sigma_i+\rho_i\theta_i^{\text{in}}}{C_{xy}-\rho_i}-v_{xy} \tag{4.30}$$

Substitute w into (4.27), we have

$$B_i^{\text{out}}=\sigma_i+\rho_i\left(\theta_i^{\text{in}}+v_{xy}\right)+\rho_i\left(\frac{\sigma_i+\rho_i\theta_i^{\text{in}}}{C_{xy}-\rho_i}-v_{xy}\right)-g_i^{\text{out}}\left(\frac{\sigma_i+\rho_i\theta_i^{\text{in}}}{C_{xy}-\rho_i}-v_{xy}-\theta_i^{\text{out}}\right)$$

$$=\left(v_{xy}+\theta_i^{\text{out}}\right)g_i^{\text{out}}+\left(\sigma_i+\rho_i\theta_i^{\text{in}}\right)\frac{C_{xy}-g_i^{\text{out}}}{C_{xy}-\rho_i} \tag{4.31}$$

For the case when there is no intersection, the curves are shown in Figure 4.11. The backlog bound is equal to the vertical distance between A_i^{in} and S_i^{out} at time θ_i^{out}, therefore we have

$$B_i^{\text{out}}=\sigma_i+\rho_i(\theta_i^{\text{in}}+\theta_i^{\text{out}}+v_{xy}) \tag{4.32}$$

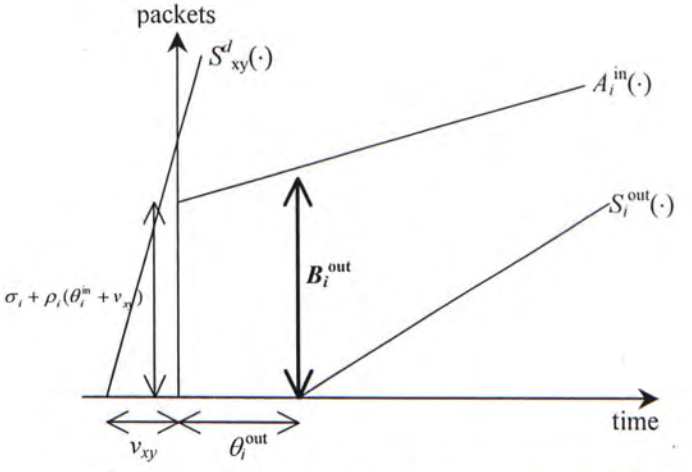


Figure 4.11: Backlog bounds for output module when there is no intersection.

Combine (4.26), (4.31) and (4.32), we have

$$B_i^{out} = \begin{cases} \left(v_{xy} + \theta_i^{out}\right) g_i^{out} + \left(\sigma_i + \rho_i \theta_i^{in}\right) \dfrac{C_{xy} - g_i^{out}}{C_{xy} - \rho_i} & \text{if } v_{xy} < \dfrac{\sigma_i + \rho_i \theta_i^{in}}{C_{xy} - \rho_i} - \theta_i^{out} \\ (\theta_i^{in} + \theta_i^{out} + v_{xy})\rho_i + \sigma_i & \text{otherwise} \end{cases} \quad (4.33)$$

Chapter 5

Simulation Results

In this chapter, we compare the performance of path-switched Benes network with original Benes network by simulation. Our simulation mainly consider the following parameters:

- Throughput;
- Packet loss rate;
- Packet delay;
- Amount of buffering.

In order to be comparable with the results in [23] and [24], we use 16 × 16 and 64 × 64 path-switched Benes network in different part of the simulation.

5.1 Uniform Traffic

The throughput to offered load ratio of path-switched Benes network and original Benes network is shown in Figure 5.1 and 5.2 respectively. The parameter w in Figure 5.1 is the look-ahead window size and the simulation of Figure 5.2 assumed output buffering is used in each switching element. From the graphs, we can see that the throughput of our model increases as the window size increases. Even the window size is zero, the performance is still better than the original Benes network without buffering. The maximum throughput will further approaches to 1 if the window size is increased continuously [4], however, the rate of increase in throughput drops rapidly.

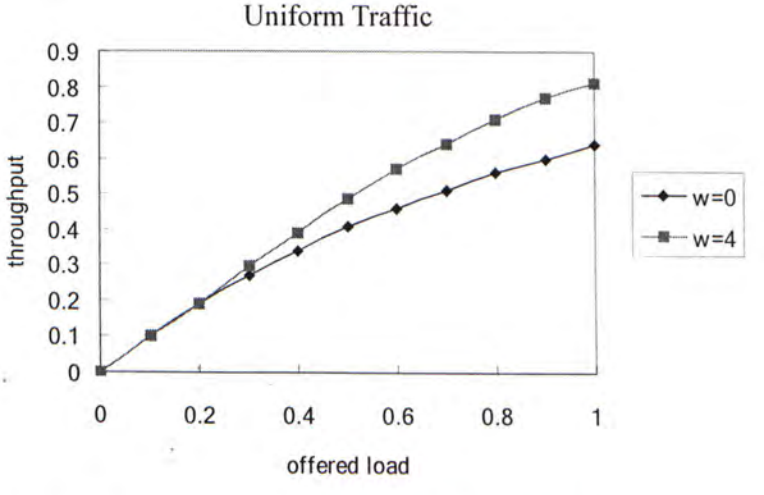


Figure 5.1: Throughput versus offered load for path-switched Benes network under uniform traffic.

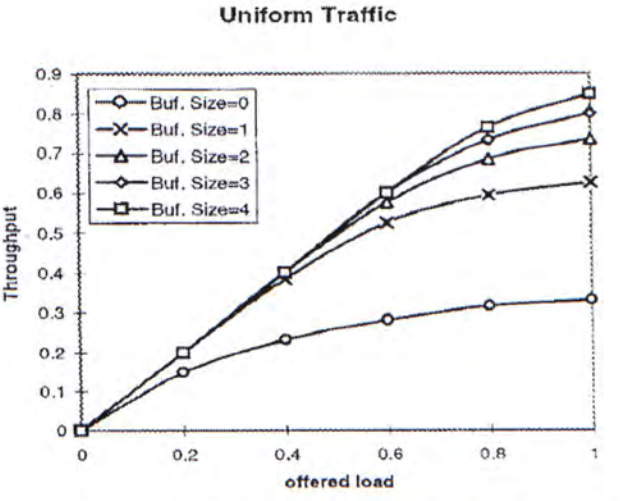


Figure 5.2: Throughput versus offered load of original Benes network under uniform traffic.

Figure 5.3 and 5.4 shows the delay as a function of offered load for path-switched and original Benes network. The delay of our model rises as the offered load increases, this is mainly due to the higher system queueing capacity. When the offered load approaches to 1, the delay of our model is similar to the original one with three buffers. The delay of the original Benes network without buffer is a constant, which is the time needed for a packet to pass $2\log_2 N$-1 stages, and is lower than our model under high offered load. However, no buffer at the switching elements will incur a high loss probability. In our simulation, the number of packets dropped is almost zero, so our model can provide a reasonable delay with extremely small packet loss probability.

Figure 5.3: Delay versus offered load for path-switched Benes network under uniform traffic.

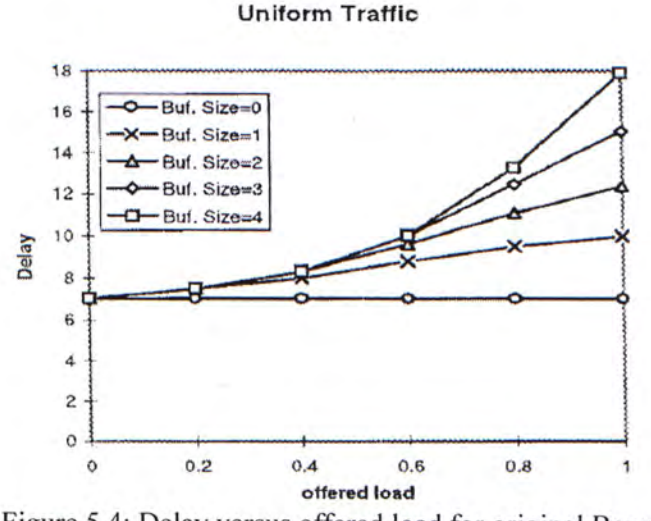


Figure 5.4: Delay versus offered load for original Benes network under uniform traffic.

5.2 Multirate Traffic

We use a special type of connection requests called hot-spot traffic to compare the performance of path-switched and original Benes network. The hot-spot traffic consists of a single output of higher access rate superimposed on a background regular uniform traffic. In other words, a hot-spot traffic implies for a number of simultaneous requests for a

specific output port. The hot-spot coefficient, h, is defined to be the fraction of packets directed to the hot-spot port. The load of the hot-spot port is $h\rho + (1\text{-}h)\rho$, where ρ is the average offered load to the network. Thus, $h\rho$ is the number of packets directed to the hot-spot destination and $(1\text{-}h)\rho$ is the number of packets directed uniformly to all N ports.

Figure 5.5 and 5.6 shows the throughput of the networks with 50% loading under different hot-spot coefficients. We can observe that our model can handle the hot-spot traffic even when the coefficient increases to 0.08.

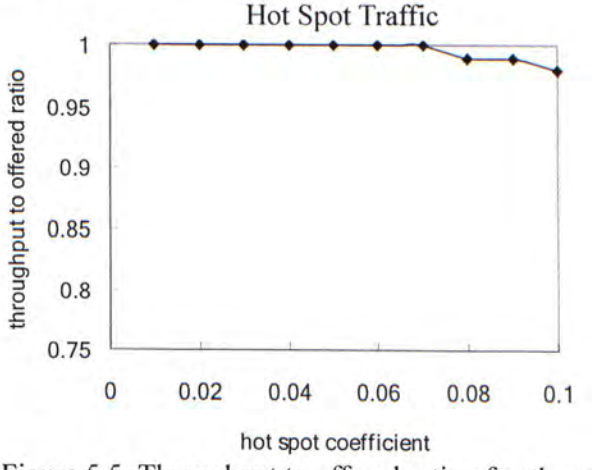


Figure 5.5: Throughput to offered ratio of path-switched Benes network with different hot spot coefficient.

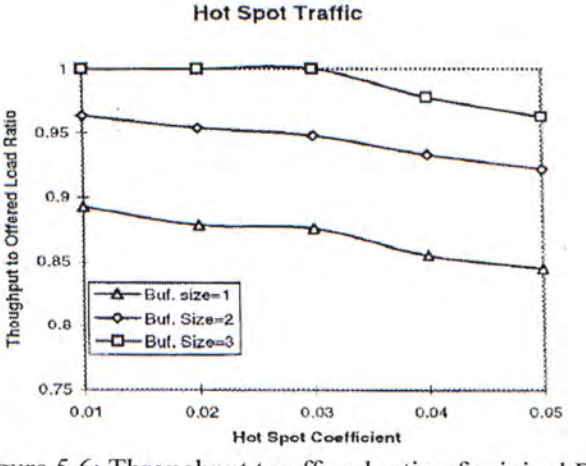


Figure 5.6: Throughput to offered ratio of original Benes network with different hot spot coefficient.

The delay versus offered load of path-switched Benes network with hot-spot coefficient 0.08 is shown in Figure 5.7. We can see that the performance of our model is not degraded under hot-spot traffic.

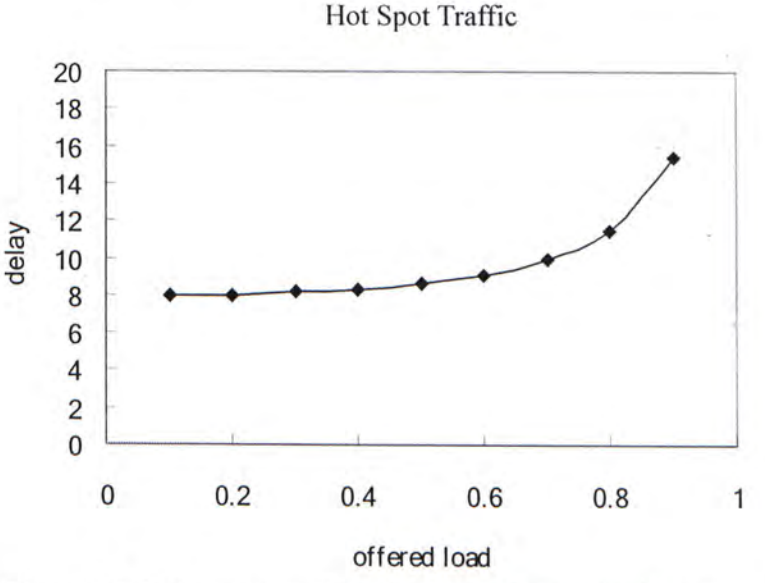


Figure 5.7: Delay versus offered load of path-switched Benes network with hot spot coefficient 0.08

Figure 5.8 shows the delays of different output port of original Benes network under hot-spot traffic. The variable g is the closeness factor of a destination output to the hot-spot port, the smaller the g, the closer the output is to the hot-spot port. The delay with g equal to 4 is the smallest because packets destined to this port do not share any path with the packets destined to hot-spot port in stages $\log_2 N$ to $2\log_2 N$-1. In path-switched Benes network, the delays of different output ports are more or less the same under the same hot-spot coefficient (Figure 5.9)

due to bandwidth guarantee of the capacity assignment and the periodically use of the connection patterns.

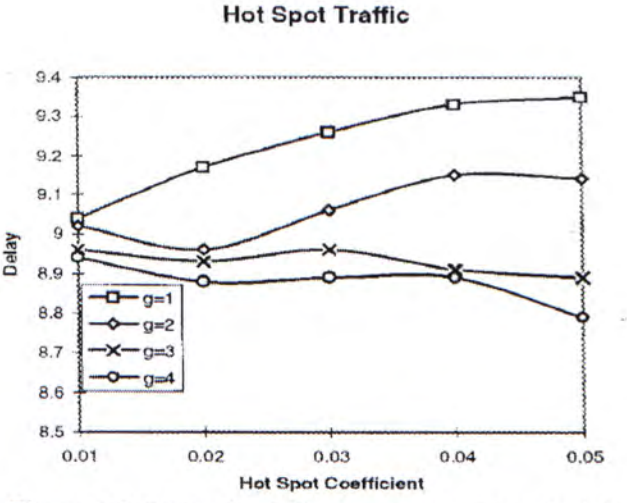


Figure 5.8: Delay for different output ports of original Benes network with various hot spot coefficient.

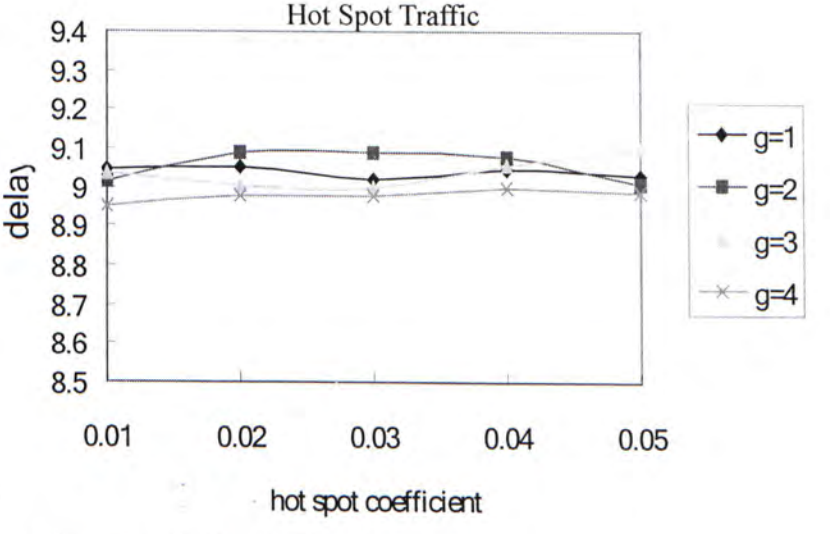


Figure 5.9: Delay for different output ports of path-switched Benes network with various hot spot coefficient.

Chapter 6

Conclusions and Future Research

Assigning nonblocking routes in Benes networks is an old problem. Many algorithms have been developed. The sequential algorithms, such as the looping algorithm are designed mainly for circuit switching, where the switch configuration can be rearranged at relatively low speed. In high-speed packet switching, the fabrics must be able to provide internally conflict-free paths simultaneously, and accommodate packets requesting connections in real time as they arrive at the inputs. That means the switching elements may need to change their states in every time slot. Although the low complexity and high scalability of the Benes network are very attractive in switch designs, it is still not appealing in high-speed packet switching.

In this thesis, we have proposed a new scheme to assign routes in Benes network by using the idea of path switching. In this so called path switched Benes network, a predetermined periodical connection pattern is used in the switching elements. As a result, calculation of path on the fly can be avoided. On the other hand, the capacities allocated to each connection are satisfied with the requests.

We have used a graphical model to study the performance guarantees offered by the path switched Benes network. Unlike other approaches, which obtain the deterministic bounds by complicated calculations, the deterministic bounds in this thesis are derived graphically in terms of arrival curves and service curves. The computation is very simple; we just need to find the maximum horizontal or vertical distance between the curves.

While the problem of assigning nonblocking routes in Benes network cannot be solved absolutely, the scheme we proposed in this thesis can greatly reduce the effect of this problem in high-speed packet switching. Simulation results show that our model performs well under both uniform and multirate traffic.

6.1 Suggestions for future research

Multicast services have been increasingly used by various continuous media applications. In our model, we have assumed the traffic is unicast. Path-switched Clos network has been shown to be capable of supporting multicast traffic with some modifications [25], [27]. We believe that the results can be extended to path-switched Benes network. However, the detail issues are yet to be investigated. It is also interesting to study the performance guarantees for multicast traffic in the path-switched Benes network.

In Chapter 4, we have studied the effect of token distribution at the central stage on the service performance. In order to provide a better delay guarantees, a token assignment algorithm must be applied. Liew's token assignment algorithm [22] can provide near-optimal performance guarantees to each data session all the time in path-switched Clos network and perhaps can be used in our model.

Bibliography

[1] T. T. Lee and C. H. Lam, "Path Switching – A Quasi-Static Routing Scheme for Large-Scale ATM Packet Switches," *IEEE J. Select. Areas. Commun.*, vol. 15, pp. 914-924, June 1997.

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

[3] R. Melen and J. S. Turner, "Nonblocking Multirate Networks," *SIAM J. Comput.*, vol. 18, no.2, pp. 301-313, April 1989.

[4] T. T. Lee and S. C. Liew, "Principles of Broadband Switching and Networking," 1995.

[5] G. Niestegge, "Nonblocking Multirate Networks," *Proc. 5th ITC Seminar*, Como Lake, Italy, May 1987.

[6] J. S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE Trans. Commun.*, vol. 36, no.6, pp. 734-743, June 1988.

[7] E. Valdimarsson, "Blocking in Multirate Networks," *in Proceedings of IEEE INFOCOM'91*, vol.2, pp. 579-588, April 1991.

[8] M. Narasihma, "The Batcher-banyan self-routing network: University and simplications," *IEEE Trans. Commun.*, vol. 36, pp.1175-1179, October 1988.

[9] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 611-624, October, 1998.

[10] A. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344-357, June 1993.

[11] R. Melen and J. S. Turner, "Nonblocking Networks for Fast Packet Switching," *in Proceedings of IEEE INFOCOM'89*, vol.2, pp. 548-557, April 1989.

[12] M. G. Hluchyj and M. J Karol, "Queueing in high-performance packet switching," *IEEE J. Select Areas on Commun.*, vol. 6, no.9, pp. 1587-1597, December 1988.

[13] A. Hung and S. Knauer, "Starlite: a wideband digital switch," *in Proceedings of GLOBECOM'84*, pp. 121-125, November 1984.

[14] L. Favalli, "Rearrangeability Conditions for Multirate Bene Networks," *in Proceedings of IEEE GLOBECOM'93*, vol.2, pp. 734-738, November-December 1993.

[15] M. C. Chan, "Providing Quality of Service Guarantees in Cross-Path Packet Switch," PhD dissertation, The Chinese University of Hong Kong, June 2000.

[16] D. C. Opferman and N. T. Tsao-Wu, "On a class of rearrangeable switching networks Part I: Control algorithm," *Bell Syst. Tech. J.*, pp.1579-1600, May-June 1971.

[17] L. Zhang, "VirtualClock: a new traffic control algorithm for packet switching networks," *ACM Trans. Computer Systems*, vol. 9, no. 2, pages 101-124, May 1991.

[18] T. T. Lee and S. Y. Liew, "Parallel Routing Algorithms in Benes-Clos Networks," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1841-1847, November 2002.

[19] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," *in Proceedings of IEEE INFOCOM'94*, pp. 636-646, June 1994.

[20] T. T. Lee, "Non-blocking copy networks for multicast packet switching," *IEEE J. Select Areas on Commun.*, vol. 6, no. 9, pp. 1455-1467, December 1988.

[21] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 378-385, June 1996.

[22] S. Y. Liew, T. T. Lee, C. W. Chan, "Bandwidth assignment with QoS guarantee in scalable ATM switches," *in Proceedings of IEEE ICC'99*, vol. 3, pp.1802-1806, June 1999.

[23] N. Mirfakhraei and Yongqun Tang, "Performance Analysis of Benes Networks under Nonuniform Traffic," *in Proceedings of IEEE ICC'96*, vol. 3, pp. 1669-1673, June 1996.

[24] E. Valdimarsson, "Blocking in Multirate Interconnection Networks," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 2028-2035, Februray-April 1994.

[25] M. Jin, Tony. T. Lee, Soung C. Liew, Soung Y. Liew, F. Tong, "Achieving nonblocking properties in a class of scalable ATM switches," submitted to *Trans. On Communication.*

[26] M. Katevenis, S. Sidiropoulos and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE J. Select. Areas Commun,*. vol. 9, pp. 1265-1297, October 1991.

[27] R. H. Lin, C. H. Lam and T. T. Lee, "Performance and Complexity of Multicast Cross-Path ATM Switches," *in Proceedings of IEEE INFOCOM'97*, vol. 3, pp. 947-954, April 1997.