ABSTRACT

DESIGN AND ANALYSIS OF SCALABLE SCHEDULING SCHEMES FOR HIGH-SPEED INPUT-QUEUED PACKET SWITCHES

by
Chuan-bi Lin

Single-stage input-queued (IQ) switches are attractive for implementation of high performance routers because they require no speedup in the used memory. It has been shown that IQ switches can provide 100% throughput under admissible traffic when using either maximum-weight matching schemes or iterative maximal-weight matching schemes with a significant speedup. These different approaches require either high computation complexity or high memory costs that can make them infeasible. Therefore, there is a need for low-complexity and fast matching schemes that provide high throughput under several admissible traffic patterns, without recurring to speedup nor multiple iterations. In this thesis, the concept of captured frame is proposed, and the application of this concept to matching schemes is demonstrated. Two weightless matching schemes, one is based on round-robin selection, called uFORM, and the other is based on random selection, called uFPIM, are presented. Furthermore, the high throughput of these schemes using a single iteration and no speedup, under a variety of admissible traffic patterns, is shown.

As switch scalability is required in high-capacity switches, a Clos-network architecture is considered. Clos-network switches are implemented with small switch modules to reduced the hardware complexity of large-capacity switches. However, the complexity of configuration schemes for these switches is high because of a) the distributed modules, and b) the high port count. This complexity can be reduced by adding memory to the first and third stages in a three-stage configuration. This switch is then called Memory-Space-Memory (MSM) switch. An effective dispatching scheme for MSM Clos-network switches must provide high throughput under any admissible traffic pattern, without expanding internal bandwidth, and while being simple to implement. To satisfy those requirements, two dispatching schemes are proposed for an MSM Clos-network switch, the framed random dispatching (FRD) and the framed concurrent round-robin dispatching (FCRRD) schemes. It is shown that these schemes, using a single matching iteration, achieve high throughput under traffic with uniform and nonuniform distributions.

Although FRD and FCRRD are simple dispatching schemes, the memory used in the MSM Clos-network switch requires speedup. Therefore, an input-queued three-stage Clos-network (IQC) switch is considered. IQC switches use no memory switch modules and are free out-of-sequence forwarding that may occur in buffered Clos-network switches, however, they have greater scheduling complexity. The configuration of IQC switches involve port matching and path routing assignment, in that order. The implementation of a scheduler capable of matching thousands of ports in large size switches may have prohibitively large complexity. To decrease the scheduler complexity for large switches, a matching scheme, called the Module-First Matching (MoM), for IQC switches that hierarchizes the matching process is proposed. In a practical scenario, this scheme performs routing first and port matching thereafter. The high switching performance of the proposed approach under uniform and nonuniform traffic is presented. A practical two-stage Clos-network switch that uses module-first matching (MoM) scheme to improve the scalability and to reduce the configuration complexity for a very large scale switch, is also presented.

A new Clos-network switch that uses the crosspoint buffers in the third-stage modules and two matching schemes to configure the new Clos-network switch are proposed to reduce resolution time and provide high performance. This switch is called Space-Space-Memory (SSM) Clos-network switch. This switch needs no memory speedup in the third-stage modules. The two configuration schemes for SSM Clos-network switches are called the weighted module-first and none-port matching (WMF-NP), and the weighted central modules' link matching (WCMM) schemes. These two approaches provide high performances for SSM Clos-network switches under uniform and nonuniform traffic, and WCMM can reduce the number of the exchange information between different modules.

DESIGN AND ANALYSIS OF SCALABLE SCHEDULING SCHEMES FOR HIGH-SPEED INPUT-QUEUED PACKET SWITCHES

by
Chuan-bi Lin

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2008

APPROVAL PAGE

DESIGN AND ANALYSIS OF SCALABLE SCHEDULING SCHEMES FOR HIGH-SPEED INPUT-QUEUED PACKET SWITCHES

Chuan-bi Lin

Dr. Roberto Rojas-Cessa, Dissertation Advisor Date
Associate Professor, Department of Electrical and Computer Engineering, New Jersey Institute of Technology

Dr. Nirwan Ansari, Committee Member Date
Professor, Department of Electrical and Computer Engineering, New Jersey Institute of Technology

Dr. Mengchu Zhou, Committee Member Date
Professor, Department of Electrical and Computer Engineering, New Jersey Institute of Technology

Dr. Aleksandar Kolarov, Committee Member Date
Senior Scientist, Applied Research, Telcordia Technologies

Dr. Jie Hu, Committee Member Date
Assistant Professor, Department of Electrical and Computer Engineering, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Chuan-bi Lin

Degree: Doctor of Philosophy

Date: January 2008

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2008

- Master of Science in Electrical Engineering,
  University of Bridgeport, Bridgeport, CT, 2001

- Bachelor of Science in Chemical Engineering,
  Tatung Institute of Technology, Taipei, Taiwan, 1994

Major: Electrical Engineering

Publications:

Chuan-Bi Lin and Roberto Rojas-Cessa, "Module Matching Schemes for Input-Queued Clos-Network Switches," *IEEE Communications Letters*, vol. 11, no. 2, pp. 194-196, February 2007.

Roberto Rojas-Cessa and Chuan-Bi Lin, "Captured-Frame Matching Schemes for Scalable Input-Queued Packet Switches," *Computer Communications (Elsevier)*, vol. 30, issue 10, pp. 2149-2161, July 2007.

Roberto Rojas-Cessa and Chuan-Bi Lin, "Captured-Frame Eligibility and Round-robin Matching for Input-Queue Packet Switches," *IEEE Communications Letters*, vol. 8, issue 9, pp. 585-587, September 2004.

Roberto Rojas-Cessa and Chuan-Bi Lin, "Scalable Two-stage Clos-network Packet Switch and Module Matching," *Proc. IEEE Workshop on High Performance Switching and Routing*, 2006.

Chuan-Bi Lin and Roberto Rojas-Cessa, "Framed Occupancy-Based Dispatching Schemes for Buffered Clos-Network Packet Switches," *Proc. IEEE International Conference on Networks*, Kuala Lumpur, Malaysia, November 2005.

Roberto Rojas-Cessa and Chuan-Bi Lin, "Frame Occupancy-based Round-Robin Matching (FORM) for Input-Queued Packet Switches," *Proc. IEEE Globecom*, vol. 3, pp. 1845-1849, November 2004.

Roberto Rojas-Cessa and Chuan-Bi Lin, "Matching Schemes with Frame Occupancy-based Eligibility for Input-Queued Packet Switches," *Proc. IEEE International Conference on Communications (ICC)*, vol. 2, pp. 972-976, Seoul, Korea, May 16-20 2005.

Chuan-Bi Lin and Roberto Rojas-Cessa, "Simulation of Maximal Scheduling Schemes for Input-Queued Packet Switches," *Proc. Conference on Information Systems and Sciences*, Princeton University, Princeton, NJ, April 2003.

To my parents, my wife and my friends

ACKNOWLEDGMENT

First, and definitely most, I would like to express my sincere gratitude to my advisor, Dr. Roberto Rojas-Cessa, for his guidance and encouragement throughout the course of my thesis. Moreover, Dr. Roberto Rojas-Cessa always shares his brilliant ideas to all his students without reservation. I greatly appreciate his patient instructions, help, and the opportunity he had given me to complete my doctoral research.

I am deeply grateful to other members of my thesis Committee: Professor Nirwan Ansari for providing a lot of help and guidance all along my stay in NJIT, Dr. Aleksandar Kolarov (Senior Scientist, Telcordia Technologies), Professor Mengchu Zhou, and Professor Jie Hu for their comments and encouragement that improved my work.

I extend my special thanks to Dr. Ronald Kane, Dean of Graduate Studies, for his support and Professor Rong-yaw Chen, Professor of Mechanical Engineering, for his help and encouragement.

I would also like to thank Dr. Ziqian Dong for providing a lot of knowledge of hardware and software of computer simulations.

Thanks to Dr. Zhen Guo, Miss Lin Cai, and Mr. Zhen Qin for their help and encouragement.

I am also grateful to the donors of the National Science Foundation and the PhoneTel Fellowship which provided financial support for my study.

I also thank my parents, brother, and friends for their support, understanding, and encouragement during these years.

Last, and most importantly, I would like to thank my wife for her love, support, confidence, and encouragement. Without her, I could not be able to complete this work. I dedicate this thesis to her.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Packet Switches

The explosive growth of the Internet has brought the demand for higher capacity at the core network and higher link rates. The high-speed connection links and high-performance switches/routers are critical elements of the new network infrastructure. Multi-terabit Internet protocol (IP) routers, asynchronous transfer mode (ATM) switches [1–3], and optical switches are examples of the technology that address those demands. Packet switching plays an important role in these switches/routers. In a packet switch, incoming variable-length packets are segmented into fixed-length packets, or cells, at the ingress side of the switch to perform internal switching, and packets are re-assembled at the egress side before they leave the switch.

The way packet switches work can be classified into two groups: time-division switching (TDS) and space-division switching (SDS) [4, 5]. TDS provides a single common internal communication path from input to output ports. Cells coming in on different input ports pass through the switch to different output ports, but do so at different times. The internal communication structure can be a ring, a bus, or memory. TDS can achieve optimal throughput and delay performance. However, TDS needs speedup to handle N incoming and N outgoing cells in a time slot, and therefore the switch size is limited and the control is more complex. Therefore, SDS is more desirable than TDS for high speed and scalable switches. SDS provides a multiplicity of paths from input ports to output ports. Cells coming in on different input ports and going to different output ports can proceed through the switch simultaneously on these separate paths, without interfering with each other. SDS can be classified based on the number of available paths between any

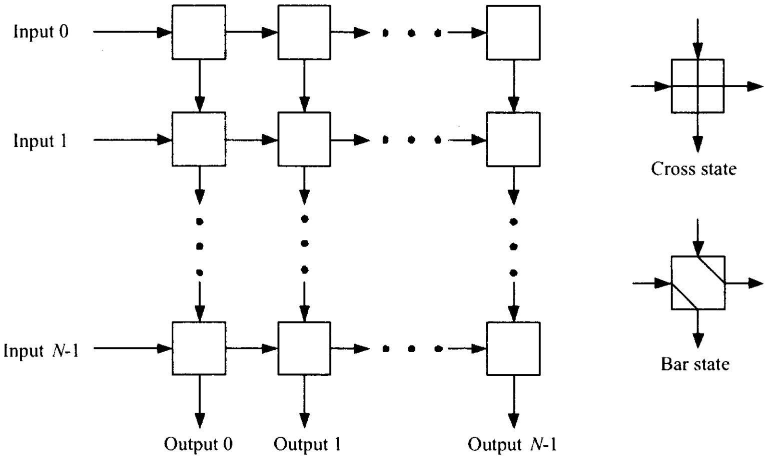

Figure 1.1 An $N \times N$ crossbar switch.

input/output pairs as single-stage switches, which have one single path, and multiple-stage switches, which have multiple possible paths.

Crossbar-based switches are very attractive switches, because they have: simple in architecture, internally nonblocking, and modularity. An $N \times N$ crossbar switch is shown in Figure 1.1, where horizontal lines represent the inputs, and vertical lines represent the outputs. The switch consists of N^2 individually operated crosspoints, one corresponding to each input-output pair. Each crosspoint has two possible states: cross and bar. The bar state is used to connect an input to an output; otherwise, the cross state is used. Therefore, a crossbar switch is a simple and nonblocking switch fabric. However, a crossbar switch can suffer from output port contention as packets from different input ports can be destined to the same output port simultaneously. To solve this contention, some packets must be stored at the buffers while one packet is transmitted to the output port.

Although output buffers, placed at the output ports, can solve the output port contention and achieve 100% throughput, they needs memory speedup to deal with N packets to be stored in an output buffer in each time slot in an $N \times N$ switch. This makes it complex to build output-queued (buffered) crossbar switches. Other ways to resolve the output port contention are to place the buffers in other locations of a crossbar switch: a) at the input ports, b) at the crosspoints, or c) at the input ports and output ports. An input-queued (IQ) switch is the most attractive architecture among the three buffered switches because an IQ switch doesn't need memory speedup [6, 7]. However, an IQ switch with first-in-first-out (FIFO) input buffers may suffer from head-of-line (HOL) blocking problem. HOL blocking causes idle outputs to remain so, even in the existence of traffic for them at an idle input, thus impeding the delivery of high throughput. The introduction of virtual output queues (VOQs), where one queue per output port is placed in an input port of an IQ packet switch, is used to remove the head-of-line (HOL) blocking problem [8].

To scale up a switch, multiple stages may be an alternative. Three-stage Clos-network switches are desirable because they employ simple and nonblocking crossbars as switch modules, resulting in low hardware complexity for building large-scale switches. Furthermore, Clos-network switches can also provide reliability since multiple paths can be used to connect from any input port to any output port. The structure of three-stage Clos-network switches, as shown in Figure 1.2, consists of three stages of switch modules [9]. The first- and third-stage have k input modules (IM) and output modules (OM), respectively, and the second-stage has m central modules (CM). At the first stage, $N = n \times k$ input lines are divided into k groups of n lines, so that an IM has n input ports. There are m output lines in an IM, each connects to all m CM. Similarly, each CM has k output lines so that it connects to all k OM. At the third stage, an OM has n output ports.

Three-stage Clos-network switches suffer from second-stage internal contention among cells from different first-stage modules excluding output contention. To remove this contention, the use of buffers in the second-stage modules without expanding internal bandwidth was considered [10, 11]. However, switches with buffers in the second-stage modules may suffer from serving packets in out-of-sequence order, which is undesirable as re-sorting packets might increase the switch complexity and cost. The other options are to use a switch with bufferless second-stage modules, where buffers are only placed in the first and third stages, or else to use a switch with bufferless three-stage modules where buffers are placed in the input ports, both options can be used without expanding inter-

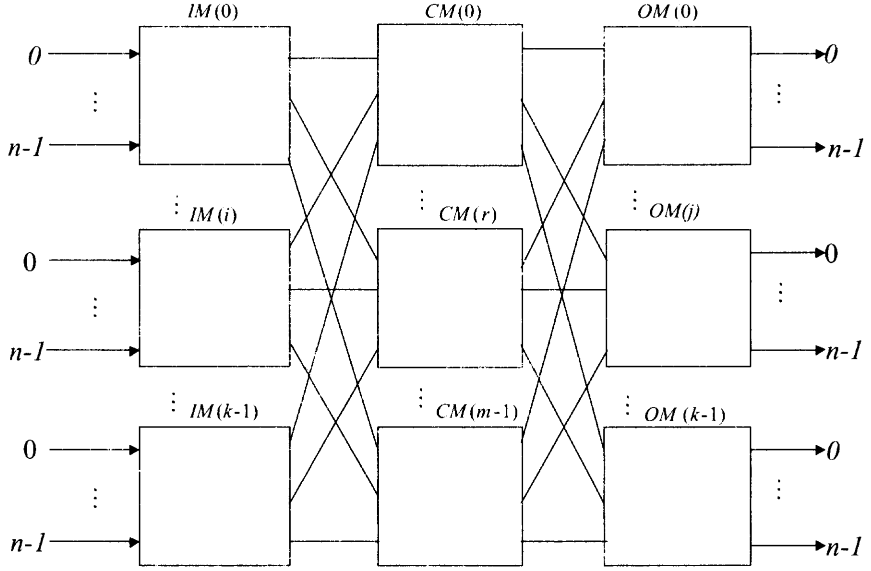

Figure 1.2 A Clos-network switch architecture.

nal bandwidth. The former is also called a memory-space-memory (MSM) Clos-network switch [12]. Since MSM Clos-network switches cannot resolve the internal contention, dispatching cells from first to second stage becomes an important issue. The latter is an input-queued Clos-network switch, called IQC switch. The IQC switch needs to configure ports matching and path routing before packets are transmitted. The switch configuration can be a complex process.

1.2 Scheduling Schemes for Packet Switches

A high-performance packet switch should provide maximum throughput, low switching delay, fair sharing, and high port speed. To achieve these goals, the scheduling scheme used in a packet switch architecture plays an important role. An efficient scheme requires feasible and fast arbiters that have: a) low complexity, b) fast contention resolution, c) fairness, and d) high matching efficiency.

A maximum-weight matching (MWM) scheme finds the matching whose weight is the highest [13]. A maximum size matching scheme [14–16] finds the maximum bipartite matching of inputs with packets queued for N outputs. An MWM scheme can provide

100% throughput under any admissible traffic patterns, where weights are assigned to input queues proportionally to their occupancy or HOL cell age. The cell arrival rate at input i for output j, received by $VOQ_{i,j}$, is denoted as $\lambda_{i,j}$. Admissible traffic is considered as:

$$\sum_{i} \lambda_{i,j} \leq 1, \sum_{j} \lambda_{i,j} \leq 1, \tag{1.1}$$

However, MWM schemes are not simple to implement as they require a computation complexity of $O(N^3 \log N)$. On the other hand, the most efficient maximum size matching scheme is known to converge in with a complexity of $O(N^{2.5})$ [17]. Because of this, maximum-size matching schemes are too slow for practical use in high speed switches.

A maximal matching is an alternative to maximum matching [18]; PIM [15], iSLIP [19], DRRM [20, 21], and iLPF [22] are some examples. These maximal-matching schemes need a number of iterations, where one iteration is the number of times that an scheduling scheme is performed to obtain a cumulative matching result, to achieve satisfactory matching results. PIM and iSLIP use simple and efficient matching arbiters, which perform the request-grant-accept approach. PIM and iSLIP are simple to implement in hardware. Other existing schemes that are neither maximum nor maximal have been proposed, and those schemes are based on load-balancing stages [23–25] or on frame-based matching [26, 27].

A class of matching algorithms based on randomized selection has been proposed to achieve high throughput with low computation complexity [28, 29]. These randomized schemes keep the matches that are likely to continue sending cells. These schemes are ALGO3, APSARA, LAURA, and SERENA. ALGO3 uses a Hamiltonian walk to achieve stability. The other three use the randomized weight augmentation, a merging procedure, and randomness, and the information provided by recent arrivals to achieve 100% throughput under admissible traffic. These randomized matching algorithms require computations and comparisons of the sum of the weight for each matching VOQ-output pair in every time slot. The resulting schemes are then a combination of weight- and weightless-based matching schemes.

1.3 Motivation and Contribution

MWM schemes have been used to show that IQ switches with VOQs can provide 100% throughput under any admissible traffic while using no speedup. Two maximum weight matching schemes were presented in [30]: one is the longest queue first (LQF) matching scheme where weights are assigned to input queues with longer occupancy; the other scheme is the oldest cell first (OCF) matching, where weights are assigned to HOL cells according to their waiting time. Moreover, the LQF algorithm can cause the starvation of one or more inputs because the matched input that has longest queue occupancy keeps the service. On the other hand, PIM with multiple iterations can provide high throughput and iSLIP with a single iteration can deliver 100% throughput under uniform traffic. However, these two weightless-based matching schemes cannot provide 100% throughput under nonuniform traffic patterns without speedup.

It is of interest to know if a weightless-based scheme can achieve a throughput of 100% under admissible traffic with nonuniform distributions, such as unbalanced traffic, with neither speedup nor multiple iterations.

A solution of weightless-based matching schemes for IQ switches is proposed. This new approach is called unlimited frame-size occupancy-based round-robin matching (uFORM) scheme, which uses the concept of capture-frame size. A frame comprises one or more cells that can be considered eligible for matching. The captured-frame concept is also used on PIM scheme giving place to a new scheme, called uFPIM. The two schemes improve the performance of the arbitration schemes without captured-frame concept. The frame size can be adjusted or limited for different switch sizes of IQ switches. A round-robin-based matching scheme, which adjusts or limits the captured-frame size, is called FORM. It can provide high throughput under admissible traffic patterns for different switch sizes with optimal captured-frame sizes.

As for IQ switches, a number of matching dispatching schemes have been proposed for MSM Clos-network switches. It has been shown that weight-based matching dispatching schemes in MSM Clos-network switches can provide high throughput under independent (uniform or nonuniform) admissible traffic. Two weight-based matching dispatching schemes are presented in [31]: one is the maximum weight matching dispatching (MWMD) scheme, based on MWM for single-stage IQ switches [13], the other dispatching scheme for this switch is the maximal oldest cell first matching dispatching (MOMD) scheme, based on the OCF algorithm for single-stage IQ switches [13]. The MOMD scheme can achieve 100% throughout with multiple iterations under uniform and unbalanced traffic [32]. However, weight-based dispatching schemes have intrinsically high computation complexity, as MWM schemes for single-stage IQ switches. On the other hand, a weightless-based dispatching scheme, called concurrent round-robin dispatching (CRRD) scheme, was proposed [33]. The CRRD scheme can provide 100% throughput under uniform traffic and is simple to implement in hardware because no comparisons need to be performed among those contending queues. However, it has been shown that weightless-based dispatching schemes can not provide high throughput with multiple iterations in input module (IM) (also referred as IM iterations) under several nonuniform traffic patterns.

An implementable solution of dispatching packets schemes for MSM Clos-network switches is also provided. This novel approach is called frame occupancy-based dispatching scheme. The approach extends the study of the captured-frame concept into dispatching schemes for MSM Clos-network switches. These dispatching schemes, the framed concurrent round-robin dispatching (FCRRD) scheme and the framed random dispatching (FRD) scheme, both using a single iteration in the matching between first- and second-stage modules, can achieve high throughput with a small number of IM iterations under several nonuniform traffic patterns. Furthermore, the throughput of FCRRD is also 100% as in CRRD.

A variety of matching schemes to configure IQC switches have been proposed [34, 35]. These schemes solve the configuration process in two phases: port matching in the first phase and routing afterwards, as the routing process uses the results of port matching. The matching processes in those schemes show high complexity and long time consumption of the configuration process. For example, in a 1024×1024 switch, these schemes would consider a scheduler that can match 1024 input ports to 1024 output ports, at once. However, the scheduler for such match can be complex to implement [36]. Those schemes show high performance under uniform traffic. However, it has not been shown these schemes provide high throughput under several nonuniform traffic patterns for IQC switches.

This results raise the question: can a scheme reduce the complexity of the configuration process and provide high throughput under several nonuniform traffic patterns for IQC switches?

As another contribution, a novel configuration scheme for IQC switches is proposed. This approach is called Module-first Matching (MoM) schemes. MoM schemes perform matching between modules in the first and third stages in the first phase, and matching between input and output ports of those matched modules, afterwards. Therefore, MoM can reduce the size of schedulers for IQC switches. As in the example above, a switch with 1024 ports, and $n = m = k = 32$, the largest matching size performed by MoM is 32 instead of 1024, and a 32×32 scheduler is feasible to implement. The MoM scheme can provide high performance under uniform and nonuniform traffic patterns. On the other hand, a practical two-stage Clos-network switch is proposed. While employing the two-stage Clos-network switch and MoM scheme, a very large scale switch is built and the configuration complexity of this scale switch is reduced.

Configuration schemes for IQC switches that perform module matching first and port matching thereafter can reduce the complexity of matching process and scheduler size [37–39]. However, these schemes require more than one iteration in the module matching process to achieve an acceptable performance. As in each module matching iteration the matching information (e.g., requests and grants) travels from IMs to CMs and back to IM (for port matching) in an scheduler implementation that follows a distributed approach, this number of iterations can accumulate a long processing delay, which can limit the switch scalability.

One question raises: Is it possible to reduce the number of iterations between different modules (referred as IM-CM iterations), the complexity of the matching process, and provider high throughput under uniform and nonuniform traffic patterns for the three-stage Clos-network switches?

A three-stage Clos-network switch with buffers in the last stage and two configuration schemes for the novel three-stage Clos-network switch are proposed. This new switch is a Space-Space-Memory (SSM) Clos-network switch that uses buffers in the crossbars of the third-stage modules. The SSM Clos-network switch can reduce a complex matching process as output contention is resolved by using crosspoint buffers in the output modules. The two weighted configuration schemes are called the weighted weighted module-first and none-port matching scheme (WMF-NP) scheme, and the weighted central modules' link matching (WCMM) scheme. The two approaches can reduce the configuration complexity and the number of IM-CM iterations as more IM-CM iterations produce long resolution delay. The SSM Clos-network switch using the proposed configuration schemes provides high throughput performance under uniform and uniform traffic models, while using no speedup at the crosspoint buffers of the third-stage modules. In addition, WCMM provides these advantages while executing single iteration between IMs and CMs, therefore, reducing the time for configuring the switch.

CHAPTER 2

CAPTURE-FRAME SELECTION SCHEMES FOR INPUT-QUEUED PACKET SWITCHES

2.1 Introduction

Input-queued (IQ) switches are attractive because their memories work without the speedup requirement of an output-queued (OQ) switch. As a result, IQ switch architectures have been adopted by several manufacturers of switches/routers. The introduction of virtual output queues (VOQs), where one queue per output port is placed in an input port of an IQ packet switch, is used to remove the head-of-line (HOL) blocking problem [8]. HOL blocking causes idle outputs to remain so, even in the existence of traffic for them at an idle input, thus impeding the delivery of high throughput.

It is common to find the following practices in packet-switch design: 1) segmentation of incoming variable-size packets at the ingress side of a switch to perform internal switching with fixed-size packets, or cells, and re-assembling the packets at the egress side before they depart from the switch; 2) use of VOQs, to avoid HOL blocking; and 3) use of crossbar fabrics for implementation of packet switches because of their non-blocking capability, simplicity, and market availability. These practices is followed in this chapter.

One major requirement for an IQ switch is the delivery of high throughput under different traffic conditions. In this chapter, admissible traffic [13] with Bernoulli and bursty arrivals that have destinations with uniform and nonuniform distributions are considered.

The matching scheme used in IQ switches determines in large measure the achievable throughput. Maximum weight matching (MWM) schemes have been used to show that IQ switches with VOQs can provide 100% throughput under admissible traffic [13] while using no speedup. However, MWM schemes have intrinsically high computation complexity that is translated into long resolution time and high hardware complexity. This makes these schemes prohibitively expensive for a practical implementation of high-speed switches with currently available technologies. An alternative is to use maximal-weight matching schemes. These schemes can provide high throughput performance under uniform traffic by using multiple iteration and under nonuniform traffic patterns by using a speedup of two or more. The hardware and time complexity of these schemes can be considered high for the ever increasing data rates because of the large number of iterations and speedup. Furthermore, some weight-based schemes may starve queues with little traffic to provide more service to the congested ones, therefore, presenting unfairness [30].

Maximal-size matching schemes can be used to resolve contention in IQ switches in a fast manner. An example of these size matching schemes is PIM [15], which is based on random selection. However, PIM cannot achieve 100% throughput under admissible uniform traffic because contentions cannot be totally avoided by this scheme. Schemes based on round-robin selection can provide higher throughput than PIM [19] under uniform traffic. Some examples of round-robin schemes are iSLIP [19], iDRRM [20, 21], and SRR [40], which can deliver 100% throughput under uniform traffic with a single iteration. iSLIP showed that the desynchronization effect, where arbiters reach the point where each of them prefers to match with different input/outputs, is beneficial for switching under this traffic pattern. However, schemes based on round-robin selections have not been shown to provide nearly 100% throughput under nonuniform traffic patterns without speedup or without pre-calculated switch configurations for traffic with pre-known distributions [23]. The exhaustive dual round-robin matching (EDRRM) scheme [41] has shown a throughput higher than iSLIP under nonuniform traffic patterns at the cost of reduced performance under uniform traffic.

This results raise the question: can weightless matching schemes provide 100% throughput under admissible traffic?

To answer this question, the schemes that perform matching for train of cells instead of for a single cell are presented. Matching in train-of-cells basis have been shown to improve throughput using an optimal train size for a given traffic distribution [26, 42]. This approach seems to be beneficial for nonuniform distributions as outputs receiving large amount of traffic may utilize efficiently an achieved match. However, it is difficult to define an optimal train size for all traffic distributions.

As a answer to that question, the captured-frame concept and its application on maximal-size matching schemes for IQ switches are introduced. The resulting schemes are maximal-size based that can provide service to VOQs in proportion to their input loads to achieve high throughput in a similar way weight-based schemes do, with however, also providing high throughput under uniform traffic. This is achieved by using the frame service, where the frame length depends on the accumulation of cells in a given period of (service) time, and therefore, the frame length is adjusted dynamically. The resulting schemes are called as the unlimited frame-size occupancy-based round-robin matching (uFORM), and the unlimited frame-size occupancy-based PIM (uFPIM). In this chapter, it is demonstrated that the captured-frame concept, used for cell matching eligibility, improves the performance of the arbitration schemes, which are run in a cell basis. In this chapter, the analysis of the achievable throughput of uFPIM in single-stage IQ switches under uniform traffic is proposed. The switching performance of uFORM and uFPIM under uniform and nonuniform admissible traffic is also showed in this chapter. It is shown that uFORM retains the high performance of round-robin schemes under uniform traffic and provides high throughput under nonuniform traffic.

This chapter is organized as follows. Section 2.2 describes the single-stage IQ switch and introduces preliminary definitions. Section 2.3 introduces the uFPIM and uFORM matching schemes. Section 2.4 analyzes the throughput of uFPIM in single-stage IQ switches. Section 2.5 presents a simulation study of the throughput and delay performance of uFORM and uFPIM under uniform and nonuniform traffic patterns. Section 2.6 presents the conclusions.

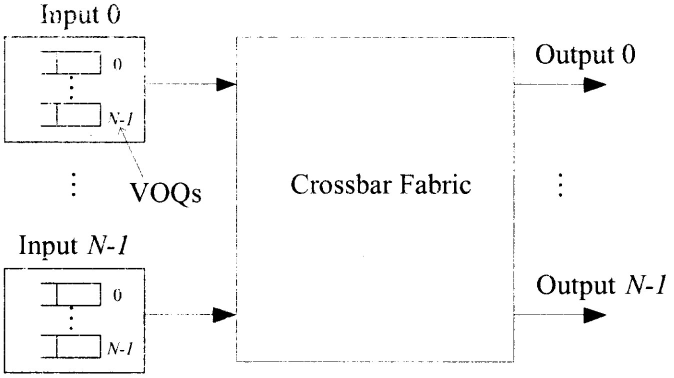

Figure 2.1 Input-queued switch with VOQs.

2.2 Single-Stage Input-Queued Switch Model and Definitions

A single-stage $N \times N$ switch, with VOQs at the inputs is considered. A VOQ that stores cells from input i to output j is denoted as $VOQ(i, j)$. Unless otherwise stated, it is considered that a VOQ can store a large number of cells. In this switch, each input can dispatch one cell each time slot and each output can receive up to one cell per time slot (i.e., no speedup is used). Figure 2.1 shows the switch model.

The following definitions in the description of the proposed matching schemes are presented.

Frame. A frame is related to a VOQ. A frame is the set of one or more cells in a VOQ that are eligible for dispatching. Only the HOL cell of the VOQ is eligible for matching each time slot.

On-service status. A VOQ is said to be in on-service status if the VOQ has a frame size of two or more cells and the first cell of the frame has been matched. An input is said to be on-service status if the status of a VOQ becomes on.

Off-service status. A VOQ is said to be in off-service status if the last cell of the VOQ's frame has been matched or no cell of the frame has been matched. Note that for frame sizes of one cell, the associated VOQ is off-service during the matching of its one-cell frame.

Captured frame size. At the time t_c of matching the last cell of the frame associated to $VOQ(i, j)$, the next frame is assigned a size equal to the cell occupancy at $VOQ(i, j)$.

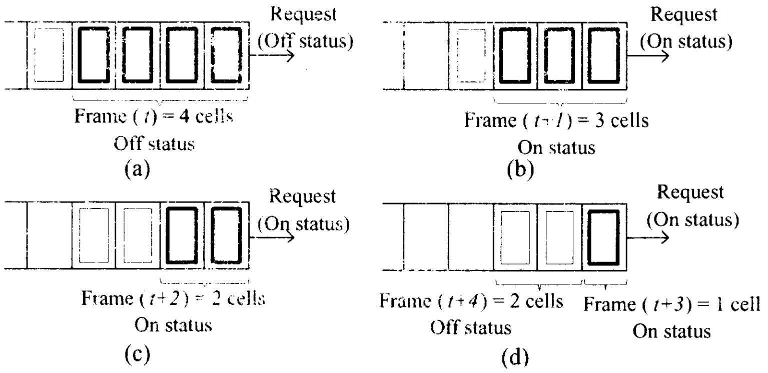

Figure 2.2 Example of a frame and the service status of a VOQ.

Cells arriving in $VOQ(i,j)$ at time t_d, where $t_d > t_c$, are not considered for matching until the current frame is totally served and a new frame is captured. This is called a captured frame as it is the equivalent of having a snapshot of the VOQ occupancy at time t_c, where the occupancy determines the frame size.

Figure 2.2 shows an example of the frame capture and the service status of a VOQ. At time slot t, the frame is off service, and the request for a match of the HoL cell is off service as well. Assuming that the size of the frame is four cells and that the VOQ is first matched during time slot t, the VOQ becomes on service at time slot $t + 1$. The status of the VOQ remains on service for the rest of the frame duration, or until time slot $t + 3$. After the last cell of the frame is matched, a new frame is captured with a size of two cells, as these cells are the only ones in the queue at this time. Then, the status of the VOQ changes to off service in the following time slot.

For each VOQ there is a captured frame-size counter, $CF_{i,j}(t)$. The value of $CF_{i,j}(t)$ indicates the frame size; that is, the maximum number of cells that a $VOQ(i,j)$ can have as candidates in the current and future time slots. $CF_{i,j}(t)$ takes a new value when the last cell of the current frame of $VOQ(i,j)$ is matched. $CF_{i,j}(t)$ decreases its count each time a cell is matched, other than the last. Each VOQ has a status flag $F_{i,j}$ to indicate the on/off service status. If VOQ is in on-service status, $F_{i,j} = 1$. Otherwise, $F_{i,j} = 0$.

2.3 Matching Schemes with Captured Frame for Single-Stage IQ Switches

2.3.1 uFPIM Matching Scheme for Single-Stage IQ Switches

The uFPIM scheme uses CF counters and F flags as indicated above. uFPIM follows three steps as in the PIM scheme:

Step 1: Request. Non-empty on-service VOQs send a request to their destined outputs. Non-empty off-service VOQs send a request to their destined outputs if input i is off-service.

Step 2: Grant. If an output arbiter a_j receives any requests, it chooses a request from the on-service VOQ (also called an on-service request) in a random fashion. If none on-service request exists, the output arbiter chooses an off-service request in a random fashion.

Step 3: Accept. If the input arbiter a_i receives any grants, it accepts one on-service grant in a random fashion. If none on-service grant exists, the arbiter chooses an off-service grant in a random fashion. The CF counter updates the value according to the following: If the input arbiter a_i accepts a grant from a_j, and:

i) If $CF_{i,j}(t) > 1$: $CF_{i,j}(t+1) = CF_{i,j}(t) - 1$, and this VOQ is set as on-service, $F_{i,j} = 1$.

ii) Otherwise $(CF_{i,j}(t) = 1)$: $CF_{i,j}(t+1)$ is assigned the occupancy of $VOQ(i,j)$, and $VOQ(i,j)$ is set as off-service, $F_{i,j} = 0$.

Figure 2.3 shows an example of a matching in the uFPIM scheme. The CF values are shown as input contents. This example only shows the captured-frame sizes and the service status at each VOQ. In the request phase, Inputs 0, 1, and 2 send off-service requests to all outputs they have at least a cell for. Input 3 sends a single on-service request to Output 0, as the off-service VOQ is inhibited as described in the scheme. The output and input arbiters select a request by service status and in a random fashion among all requests of the same service status, as shown by the grant and accept phases. Output 0 selects the on-service request from Input 3 over the off-service request from Input 1. After the match is completed, the CF values are updated as shown in the figure. Note that at time slot $t + 1$, three VOQs become on service.

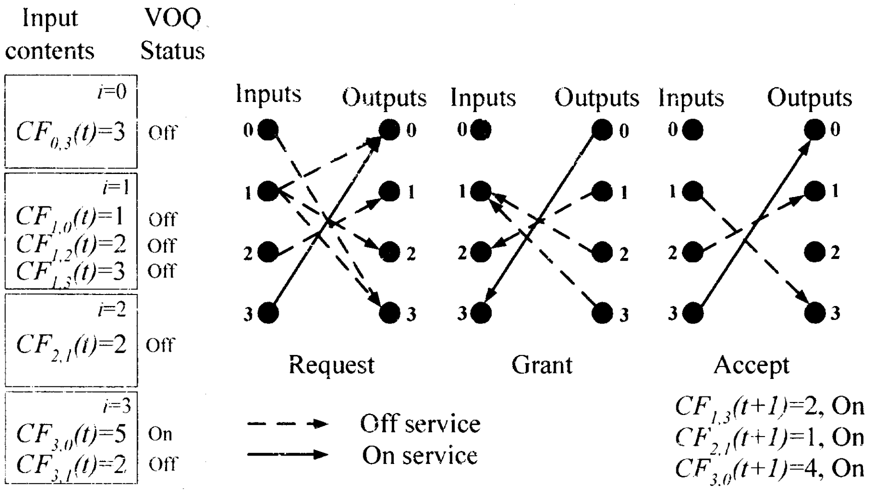

Figure 2.3 Example of uFPIM in a 4 × 4 switch.

2.3.2 *u*FORM Matching Scheme for Single-Stage IQ Switches

uFORM follows request-grant-accept steps as in uFPIM, and uses round-robin selection instead of random-based selection. The matching process is as follows:

Step 1: Request. Non-empty on-service VOQs send a request to their destined outputs. Non-empty off-service VOQs send a request to their destined outputs if input i is off-service.

Step 2: Grant. If an output arbiter a_j receives any requests, it chooses a request from the on-service VOQ (also called an on-service request) that appears next in a round-robin schedule, starting from the pointer position. If none on-service request exists, the output arbiter chooses an off-service request that appears next in a round-robin schedule, starting from its pointer position.

Step 3: Accept. If the input arbiter a_i receives any grants, it accepts an on-service grant in a round-robin schedule, starting from the pointer position. If none on-service grant exists, the arbiter chooses an off-service grant that appears next in round-robin schedule starting from its pointer position. The input and output pointers are updated to one position beyond the matched one. In addition to the pointer update, the CF counter updates the value according to the following: If the input arbiter a_i accepts a grant from a_j, and:

i) If $CF_{i,j}(t) > 1$: $CF_{i,j}(t+1) = CF_{i,j}(t) - 1$, and this VOQ is set as on-service, $F_{i,j} = 1$.

ii) Otherwise ($CF_{i,j}(t) = 1$): $CF_{i,j}(t+1)$ is assigned the occupancy of $VOQ(i,j)$, and $VOQ(i,j)$ is set as off-service, $F_{i,j} = 0$.

The prefix unlimited to the names of these two matching schemes is used because the captured-frame size is not limited to a maximum value at the capture time.

Figure 2.4 shows an example of uFORM in a 4×4 switch. In this example, the contents of the VOQs are the same as that of the uFPIM example. The pointers of the input and output arbiters are initially positioned as shown in the request phase. The off inputs send requests to all outputs they have a cell for. In the grant phase, the output arbiters select the request according to the request status and the pointer position. Output 0 selects the on-service request over the off-service request. Output 3 receives two off-service requests, and selects Input 1 because that input has higher priority according to the pointer position. Outputs 1 and 2 receive a single off-service request, therefore, the requests are granted. In the accept phase, Input 1 selects Output 3 by using the pointer position. Input 2 accepts the single grant issued by Output 1. Input 3 accepts the single grant, issued by Output 0. Since the results are the same as in the uFPIM example, the CF values and service status are updated as in that example. Note that the input and output arbiters for the on-service ports (Input 3 and Output 0) are updated, but since the service status takes higher precedence, the pointer position in this case becomes secondary in the selection process.

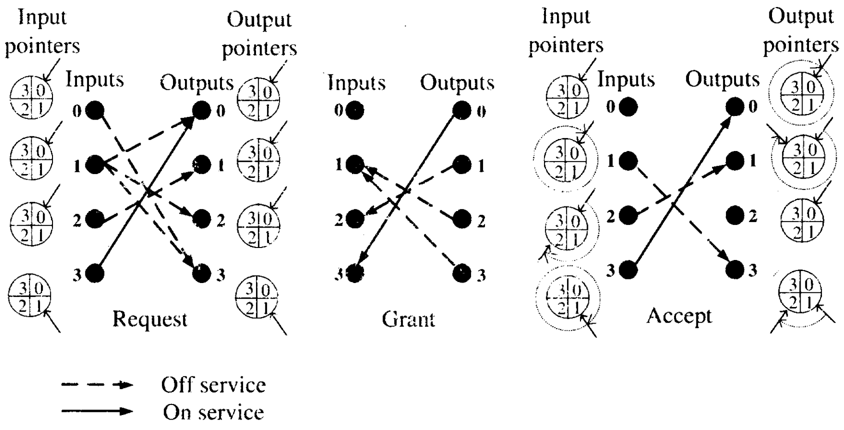

Figure 2.4 Example of the uFORM scheme in a 4 × 4 switch.

2.4 Throughput Analysis of Randomized Selection using the Captured-Frame Concept

This section analyzes the throughput of uFPIM and shows the improvement of over the throughput of PIM. It has been shown that the throughput of an IQ switch using PIM under uniform traffic [43, 44] for a large N and a single iteration, where PIM's throughput (T_{PIM}) is defined by:

$$T_{PIM} = 1 - (1 - \frac{\rho}{N})^N. \tag{2.1}$$

where, N is the number of input/output ports and ρ is the probability of a cell arrival in a time slot. As presented in [43], the probability of a request that is being granted by output j is ρ/N. The probability that output j does not receive a cell from any inputs is $(1 - \rho/N)$. When N is large and $\rho = 1.0$, T_{PIM} is known to be 63.2% under uniform traffic with Bernoulli arrivals.

The uFPIM scheme uses the captured-frame concept. In this scheme, a frame is defined at the end of the VOQ service and those cells that arrived during the (frame) servicing time are considered part of the next frame. Therefore, cell arrivals (after a frame is defined) do not affect arbitrarily the matching process. Furthermore, once a match is achieved, the match is kept during the frame duration and the input is set on-service, thus, reducing the number of contending ports that participate in random selection. In subsequent time slots, the number of matches is increased because the use of frames makes a match last during the time that a frame is served. Therefore, the probability of a request of being granted by an output i is

$$\rho/(N - E(m))$$

and the throughput of uFPIM, T_{uFPIM}, is defined by

$$T_{uFPIM} = \frac{E(m)}{N} +$$

$$\frac{N - E(m)}{N}(1 - (1 - \frac{\rho}{N - E(m)})^{N-E(m)}), \tag{2.2}$$

where $E(m)$ is the average number of on-service inputs.

In the worst case, $E(m)$ is then defined by the number of cells in a frame. Because of the two states of an input (on-service or off-service), it is considered that the average of the duration of a frame, P_m, follows a binomial distribution:

$$P_m = (\genfrac{}{}{0pt}{}{N}{m})p^m(1 - p)^{N-m}, m = 0, 1, 2, \ldots, N, \tag{2.3}$$

where p is the probability that an input becomes on service. Therefore,

$$E(m) = \sum_{m=0}^{N} m \cdot P_m \tag{2.4}$$

or

$$\begin{aligned} E(m) &= 0 \cdot P_0 + 1 \cdot P_1 + \ldots + N \cdot P_N \\ &= 0 + 1 \cdot (\begin{matrix} N \\ 1 \end{matrix}) p^1 (1-p)^{N-1} + \cdots + N \cdot (\begin{matrix} N \\ N \end{matrix}) p^N \\ &\geq N \cdot p \end{aligned} \tag{2.5}$$

When the switch size N is large,

$$E(m) = N \cdot p. \tag{2.6}$$

Recalling that after the first cell of the frame with two or more cells is matched, the status of the VOQ becomes on-service. That is, a VOQ must not be matched at the least first two time slots (as the number of cells require to form a frame with the minimum size to become on-service is two), and the VOQ can get matched in the third time slot, therefore, capturing a new frame-size. Let's have p_{1st-um} and p_{2nd-um} denote the probability that an output does not get matched in the first and second time slots, respectively, and p_{3rd-m} is the probability that the output gets matched in the third time time slot, p becomes:

$$\begin{aligned} p &= 1 - p_{1st-um} \cdot p_{2nd-um} \cdot p_{3rd-m} && (2.7) \\ &= 1 - (1 - \frac{1}{N-A_1})^{N-A_1} \cdot (1 - \frac{1}{N-A_2})^{N-A_2} \\ &\cdot \ (1 - \frac{1}{N-A_3})^{N-A_3} \\ &\geq 1 - ((1 - \frac{1}{N})^N)^3, && (2.8) \end{aligned}$$

where A_1, A_2, and A_3 represent the number of on-service inputs at the first, second, and third time slots, respectively. Here, because $E(m)$ is difficult to calculate, the following approximation

$$1 - ((1 - \frac{1}{N})^N)^3$$

is used to calculate p. From Eq. (2.5) and Eq. (2.8),$E(m)$ it is obtained. Using $E(m)$ in Eq. (2.2) the maximum throughput of uFPIM is estimated when $\rho = 1.0$. For example, if $N = 32, p \geq 0.952$, and $E(m) \simeq N \cdot p \simeq 30$, and then $T_{uFPIM} = 0.986$.

When N is large,

$$p = 1 - ((1 - \frac{1}{N})^N)^3 \simeq 1 - (\frac{1}{e})^3 = 0.95 \tag{2.9}$$

then

$$E(m) = 0.95N.$$

Now, using this value in Eq. (2.2) and

$$\begin{aligned} T_{uFPIM} &= \frac{0.95N}{N} + \\ &\quad \frac{N - 0.95N}{N} \cdot \\ &\quad (1 - (1 - \frac{1}{N - 0.95N})^{N-0.95N}) \\ &= 0.95 + 0.05 \cdot (1 - \frac{1}{e}) \\ &= 0.982 \end{aligned} \tag{2.10}$$

In this way, for a large N, the actual T_{uFPIM} is higher than 0.982.

Figure 2.5 shows the throughput of uFPIM produced by analysis and simulation under Bernoulli uniform traffic. The analysis result is close to the simulation result under different switch sizes. The actual throughput is expected higher than the one obtained through analysis because of the Eqs. (2.6) and (2.8).

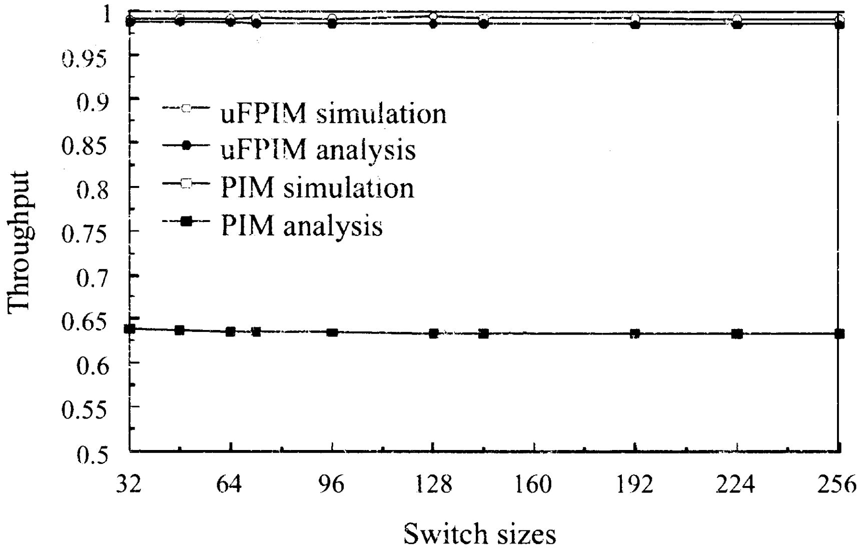

Figure 2.5 Throughput comparison of analysis and simulation of uFPIM with different switch sizes under Bernoulli uniform traffic.

2.5 Performance Evaluation of uFPIM and uFORM

*i*SLIP (with one iteration, or 1SLIP) and PIM on this study for comparison purposes are considered. The performance evaluations are produced by computer simulation, where results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay. The traffic models considered have destination with uniform and nonuniform distributions. The simulation does not consider the segmentation and re-assembly delays for variable size packets.

2.5.1 Uniform Traffic

Figure 2.6 shows the simulation results of 32 × 32 IQ switches with 1SLIP, PIM, uFORM, and uFPIM under uniform traffic with Bernoulli arrivals. This figure shows that uFORM, as *i*SLIP, delivers 100% throughput under uniform traffic. Under this traffic, PIM delivers about 63% throughput. However, when using the captured frame-size concept in uFPIM, the throughput improves to nearly 100%. The reason for the improvement by uFPIM is that, once a match is achieved, the match is kept during the frame duration. Therefore, contention among the others ports is reduced with each time slot.

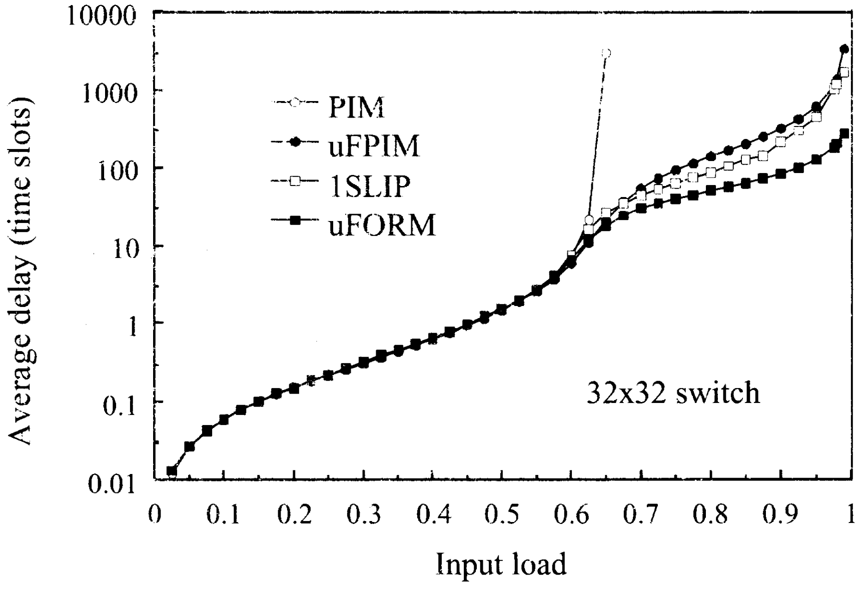

Figure 2.6 Average delay of uFORM and uFPIM schemes under Bernoulli uniform traffic.

Figures 2.7 and 2.8 show the average cell delay produced by uFORM and uFPIM schemes as a function of the offered load for switches with 8, 16, 32, 64, 128, and 256 ports. It can be seen that as the switch size increases, the average cell delay increases. However, in a load close to 1.0, small switches, $N = \{8\}$ of uFORM and $N = \{8, 16\}$ of uFPIM, produce a long average delay.

2.5.2 Nonuniform Traffic

These four schemes under five nonuniform traffic models is simulated: unbalanced [32], Chang's [23], asymmetric [45], diagonal [46, 47], and power-of-two (PO2) [26].

The unbalanced traffic model uses a probability, w, as the fraction of input load directed to a single predetermined output, while the rest of the input load is directed to all outputs with uniform distribution. Let us consider input port s, output port d, and the offered input load for each input port ρ. The traffic load from input port s to output port d, $\rho_{s,d}$ is given by,

$$\rho_{s,d} = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \tag{2.11}$$

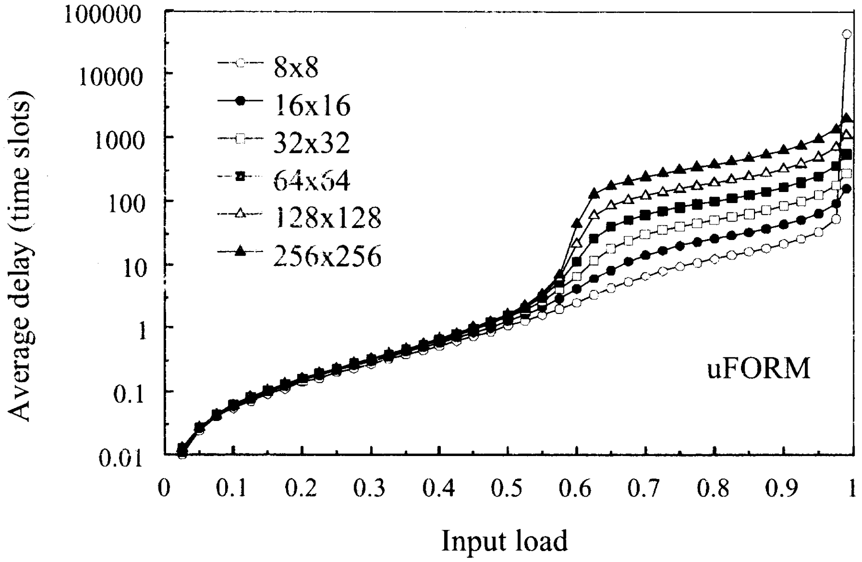

Figure 2.7 Average delay of uFORM in function of switch size, under Bernoulli uniform traffic.

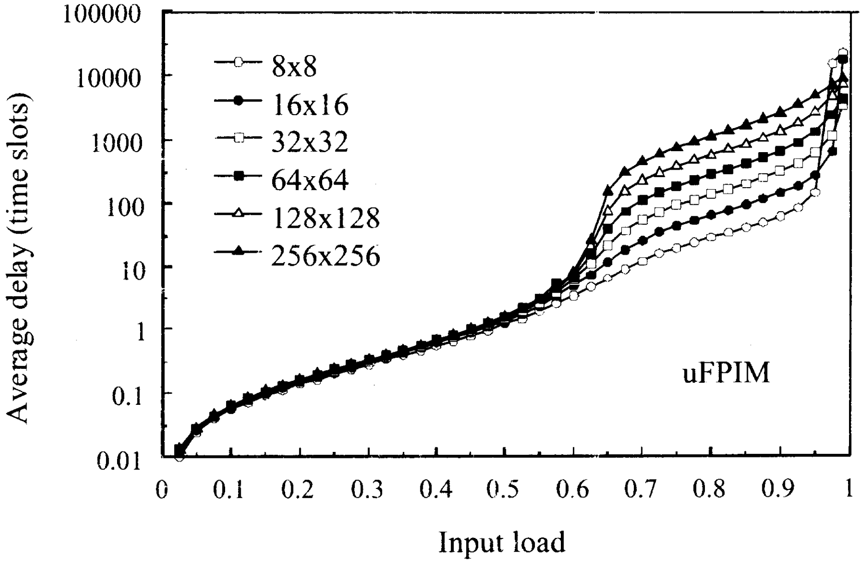

Figure 2.8 Average delay of uFPIM in function of switch size, under Bernoulli uniform traffic.

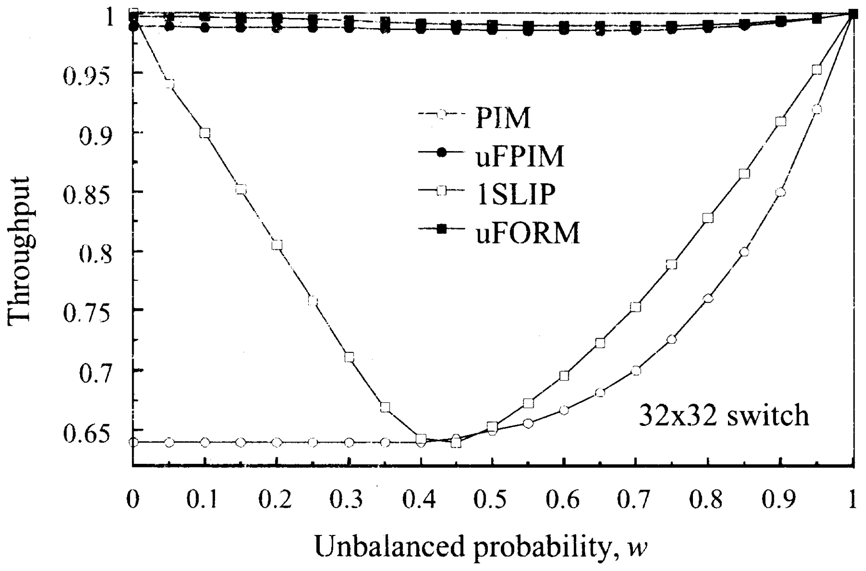

Figure 2.9 Throughput performance of uFORM and uFPIM under unbalanced traffic.

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, it is completely directional, from input i to output j, where $i = j$. This means that all traffic of input port s is destined for only output port d, where $s = d$. Figure 2.9 shows the throughput performance of 1SLIP, PIM, uFPIM, and uFORM under unbalanced traffic. This figure shows that uFORM provides over 99% throughput under the complete range of w and that uFPIM reaches up to 99% throughput, while both PIM and 1SLIP reach 64% throughput. The high throughput of uFORM and uFPIM under this traffic model is the product of considering the VOQ occupancy. uFORM ensures service to queues with high load by capturing a large frame size for each, and to the queues with low load by using round-robin selection.

Figure 2.10 shows the throughput performance of uFPIM and uFORM with different switch sizes under Bernoulli unbalanced traffic with 1.0 input load. This figure shows that uFORM provides over 99% throughput and that uFPIM reaches just 99% throughput for large switches. However, small switches, $N = \{8, 16\}$, have the lower throughput because they are more sensitive to the value of the captured frame sizes.

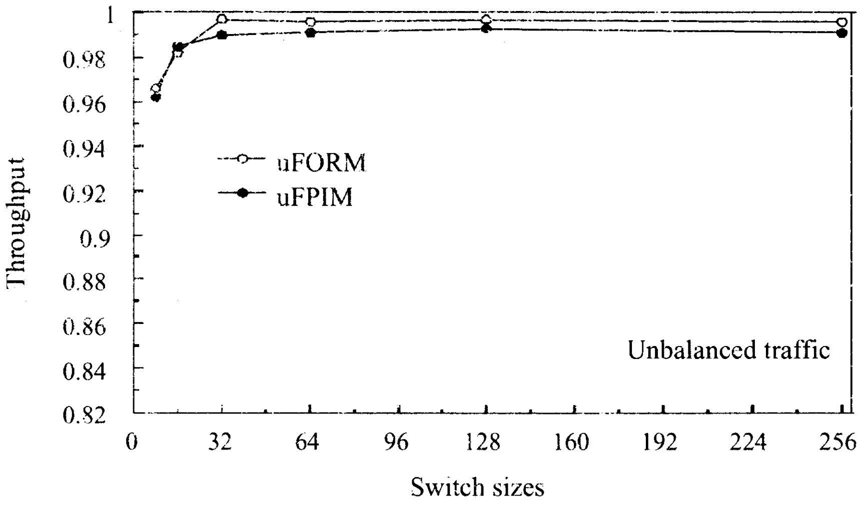

Figure 2.10 Throughput performance of uFORM and uFPIM in function of switch size, under unbalanced traffic.

Chang's traffic model can be defined as $\rho = 0$ for $i = j$, and $\rho_{i,j} = \frac{1}{N-1}$ otherwise. Figure 2.11 shows the average cell delay achieved by the four matching schemes under this traffic model. The results show that the obtained throughput is 64% by PIM, 97% by 1SLIP, and 99% by both uFORM and uFPIM.

The asymmetric traffic model is defined as the following. Consider the asymmetry coefficients $a_0 = 0, a_1 = (f-1)/(f^{N-1}-1), a_j = a_1 \cdot f^{j-1} \; \forall j \neq 0$. Then $\lambda_{(i,(i+j) \bmod N)} = a_j$ and $\lambda_{i,j}/\lambda_{[(i+1) \bmod N],j} = f \; \forall i \neq j$ and $[(i+1) \bmod N] \neq j$, and $f = (100:1)^{-1/(N-2)}$, where $\lambda_{i,j}$ is the input load from input i to output j. The coefficients define that each neighboring port receives a load difference of f. Figure 2.12 shows the average cell delay of the matching schemes under the asymmetric traffic model. The results show that the obtained throughput is 70% by PIM, 72% by 1SLIP, and above 99% by uFORM and uFPIM.

The diagonal traffic model distributes all the load of an input between two different outputs, making the distribution heavily distributed among a small number of output. This traffic model is defined as $\rho_{i,j} = 0.5$ for $j = i$ and $j = (i+1) \bmod N$, and $\rho_{i,j} = 0$ otherwise. Figure 2.13 shows the average cell delay of thses matching schemes under this

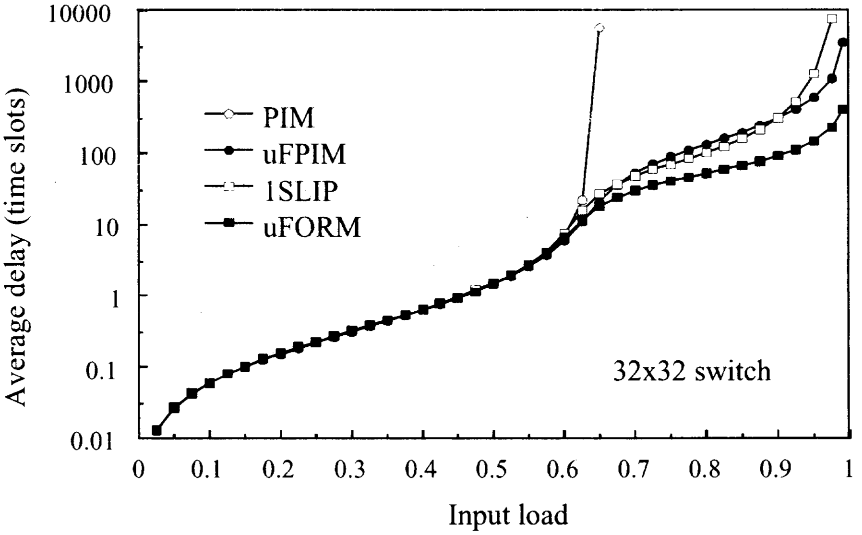

Figure 2.11 Throughput performance of uFORM and uFPIM under Chang's traffic.

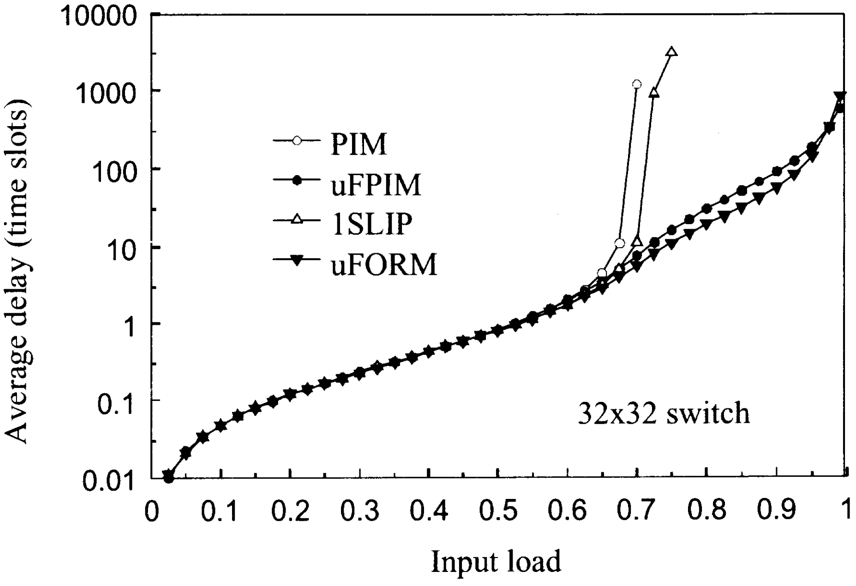

Figure 2.12 Throughput performance of uFORM and uFPIM under asymmetric traffic.

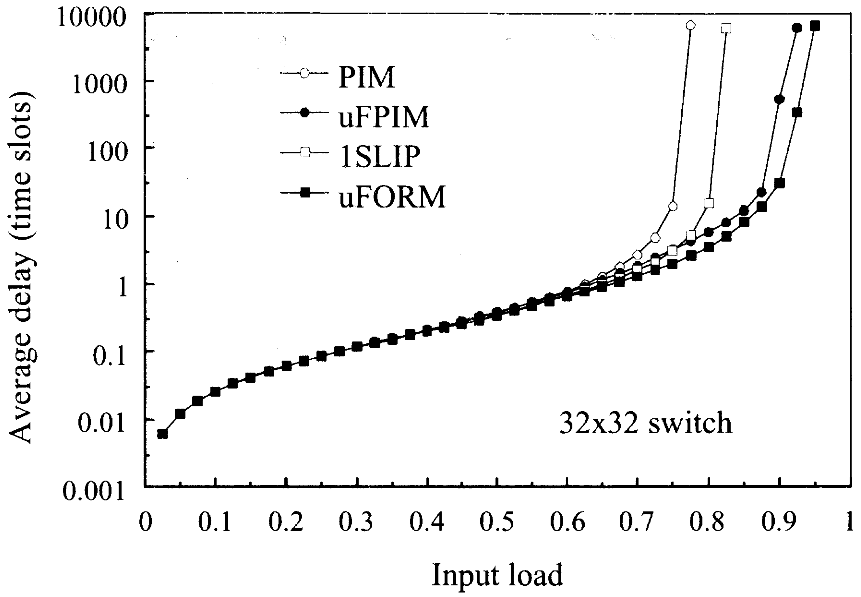

Figure 2.13 Throughput performance of uFORM and uFPIM under diagonal traffic.

traffic model. These results show that the obtained throughput is 75% by PIM, 85% by 1SLIP, 90% by uFPIM, and 95% by uFORM.

The PO2 traffic model can be represented by a matrix load as: where λ_i is the load at input i (e.g., $\lambda_{0,0} = \frac{\lambda_0}{2^1}, \cdots, \lambda_{N-1,N-1} = \frac{\lambda_{N-1}}{2^{N-1}}$). This traffic model presents a large nonuniform distribution among all inputs and outputs. The distribution difference in an input changes along all N possible destinations. Figure 2.14 shows the performance of the four matching schemes under the PO2 traffic model. Because of the complexity of describing the PO2 traffic model in the simulation program, a 30×30 switch is considered for simulation. Under this traffic model, the obtained throughput is 72% by PIM, 75% by 1SLIP, and 95% by both uFPIM and uFORM. Although uFPIM and uFORM provide below 99% throughput under PO2, these schemes show, nevertheless, performance improvement over the other schemes.

In summary, Figures 2.6-2.14 show that the throughput is improved by using the captured-frame concept to define the set of eligible cells for the matching process.

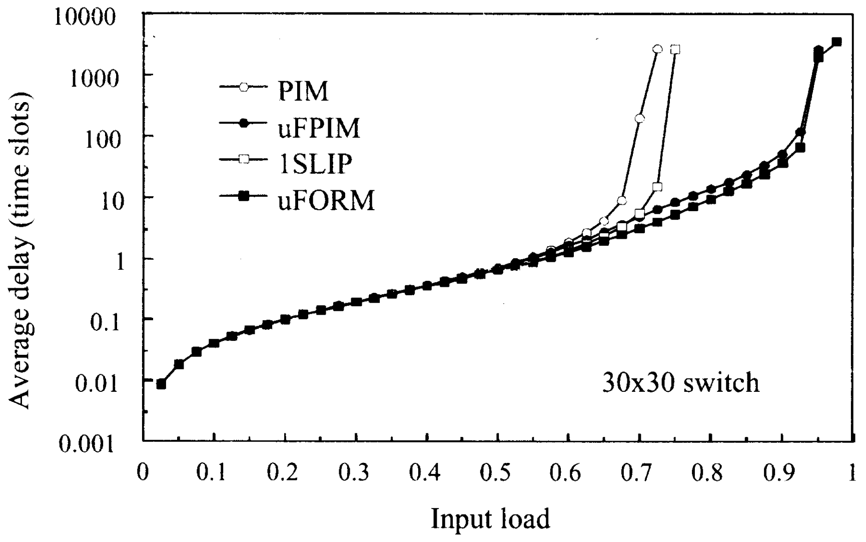

Figure 2.14 Throughput performance of uFORM and uFPIM under PO2 traffic.

2.5.3 Discussion of Performance Results

The use of a captured frame size and the service concepts used here make uFORM and uFPIM deliver high performance under uniform and unbalanced traffic patterns. Note that in the case where a VOQ has no cells at the capturing time, VOQ can still participate in a matching when a cell arrives after that, as long as the input is off-service.

When a VOQ changes its status to on-service, that VOQ has higher priority than the others to continue sending its request in subsequent time slots. When an input is off-service, all nonempty VOQs (independently of the CF value) send a request to their respective outputs.

Under uniform traffic, the captured frame sizes are not expected to reach large values because of the cell distribution among all queues. Therefore, most queues may remain in off-service status while completing service for one-cell frames. The performance is then determined by the selection policy. Furthermore, as the captured frame includes old cells, the delay may be smaller than pure round-robin or random based matching. Under unbalanced traffic, some queues are expected to have heavier loads than others. The queues with large occupancies have a higher service than the queues with lower occupancy. The difference on frame sizes results in more service for queues with a larger number of arrivals than those for queues with a small number of arrivals. Moreover, the selection policy ensures that all queues receive service.

2.6 Conclusions

In this chapter, the captured-frame size concept to determine cell eligibility in the matching process for input queued packet switches, and two matching schemes, uFORM and uFPIM, that use the captured frame concept, a single iteration, and no speedup, both for single-stage IQ switches were introduced. In this chapter, it was analyzed that the throughput of uFPIM is performed under uniform traffic with independent and identical distributions, and it was demonstrated that uFPIM can achieve higher throughput than PIM. Furthermore, the proposed schemes are tested under several nonuniform traffic patterns. The presented schemes show above 99% throughput under the unbalanced traffic model, using a single iteration and no speedup. uFORM and uFPIM were also studied under Chang's, asymmetric, diagonal and PO2 traffic models and these schemes showed higher switching performance than those schemes without the captured-frame concept. The new schemes give similar performance to that of weight-based matching schemes under nonuniform traffic patterns without recurring to queue comparisons and keep the high throughput of weightless schemes under uniform traffic. Furthermore, the proposed concept is scalable as the throughput performance increases as the switch size increases.

CHAPTER 3

FRAME OCCUPANCY-BASED ROUND-ROBIN MATCHING SCHEME FOR INPUT-QUEUED PACKET SWITCHES

3.1 Introduction

A single-stage IQ switch, based on a crossbar switch fabric and VOQs, has the throughput performance dependable mainly on the used matching scheme. In general, matching schemes are required to provide: a) low complexity, b) fast contention resolution, c) fairness, and d) high matching efficiency. One major requirement for an IQ switch is the delivery of high throughput under admissible traffic. The admissible traffic [13] with Bernoulli and bursty arrivals, and uniform and nonuniform distributions are considered.

As discussed in Chapter 2, the captured-frame concept and its application on maximal-size matching schemes for IQ switches have been introduced. These two weightless-based matching schemes, the unlimited frame-size occupancy-based round-robin matching (uFORM) and the unlimited frame-size occupancy-based PIM (uFPIM), can achieve throughput of nearly 100% under admissible traffic with nonuniform distributions, using a single iteration and no speedup, for large IQ switches. However, smaller switches are less sensitive to the unlimited frame-size occupancy values.

As an application to different switch sizes, a variation of uFORM is developed, this is called frame occupancy-based round-robin matching (FORM). This scheme, which captures the limited frame-size occupancy values, also provides high throughput under uniform and unbalanced traffic patterns for small switch sizes with different limited frame-size values.

This chapter is organized as follows. Section 3.2 presents the switch model under study and several definitions. Section 3.3 introduces the proposed arbitration scheme. Section 3.4 presents a simulation study of the throughput and delay performance of the resulting switch under uniform and nonuniform traffic patterns. Section 3.5 presents the conclusions.

3.2 Switch Model and Preliminary Definitions

The following definitions are used in the description of the proposed matching scheme.

Frame. A frame is related to a VOQ. A frame is the set of one or more cells in a VOQ that are eligible for matching. Only the HOL cell of the frame is eligible for matching at each time slot.

On-service status. A VOQ is said to be on-service status if the VOQ has a frame size of two or more cells and the first cell of the frame has been matched. An input is said to be on-service status if there is at least one on-service VOQ.

Off-service status. A VOQ is said to be off-service if the last cell of the VOQ's frame has been matched (i.e., finished service) or no cell of the frame has been matched (i.e., not started service yet). Note that for a frame size of one cell, the associated VOQ is off-service during the matching of its one-cell frame. An input is said to be off-service if all VOQs are in off-service status.

Captured frame size. At the time t_c of matching the last cell of the frame associated to $VOQ_{i,j}$, the next frame is assigned a size equal to the minimum of the cell occupancy, denoted as $L_{i,j}(t_c)$, at $VOQ_{i,j}$ and a minimum limiting value f_m, where $1 \leq f_m \leq L_{i,j}(t_c)$. Cells arriving at $VOQ_{i,j}$ at time t_d, where $t_d > t_c$, are not considered for matching until the current frame is totally served and a new frame is captured.

3.3 Frame Occupancy-based Round-Robin Matching (FORM) Scheme

The proposed matching scheme is based on round-robin selection. For each output, there is an output arbiter a_j that selects a request among all received according to the policies described in the matching algorithm. For each input, there is an input arbiter a_i that accepts a grant among all received according to the policies described in the matching scheme. Each arbiter has a pointer that indicates the counter-part port with the highest priority position in a round-robin schedule.

For each VOQ, there is a captured frame-size counter, $CF_{i,j}(t)$. This captured frame size is counted as it is the equivalent of having a snapshot of the occupancy of a VOQ at a given time t, thus, the frame size is then equivalent to the occupancy at time t. The value of $CF_{i,j}(t)$, $|CF_{i,j}(t)|$, indicates the frame size; that is, the maximum number of cells that a $VOQ_{i,j}$ can have as candidates in the following and future time slots. $|CF_{i,j}(t)|$ takes a new frame-size value when the last cell of the current frame of $VOQ_{i,j}$ is matched. $|CF_{i,j}(t)|$ decreases its count by one each time a cell is matched other than the last. VOQs are considered either on-service or off-service. All VOQs are initially considered with a frame size of one cell and in off-service status.

This scheme follows request-grant-accept steps, as in the iSLIP algorithm [19]. The arbitration process is as follows:

Step 1: Request. Non-empty on-service VOQs send a request to their destined outputs. Non-empty off-service VOQs send a request to their destined outputs only if the input is off-service.

Step 2: Grant. If an output arbiter a_j receives two or more requests, it chooses a request of an on-service VOQ (also called an on-service request) that appears next in a round-robin schedule, starting from the pointer position. If no on-service request exists, the output arbiter chooses an off-service request that appears next in a round-robin schedule, starting from the pointer position.

Step 3: Accept. If the input arbiter a_i receives two or more grants, it accepts one on-service grant that appears next in a round-robin schedule, starting from the pointer position. If no on-service grant exists, the arbiter chooses an off-service grant that appears next in a round-robin schedule, starting from its pointer position. The input pointers are updated to one position beyond the accepted ports. The output pointers are updated to one position beyond the accepting port. In addition to the pointer update, the CF counter updates its value according to the following: If the input arbiter a_i accepts a grant from output arbiter a_j:

i) If $|CF_{i,j}(t)| > 1$: $|CF_{i,j}(t+1)| = |CF_{i,j}(t)| - 1$, and $VOQ_{i,j}$ is set as on-service.

ii) Otherwise (i.e., $|CF_{i,j}(t)| = 1$): $|CF_{i,j}(t+1)|$ is assigned the minimum of the occupancy of $VOQ_{i,j}$ and f_m, and $VOQ_{i,j}$ is set as off-service.

The variable f_m is a value to limit the captured frame size. Note that f_m may be equal to a constant or a variable value. In this paper, f_m is used as a constant. The frame size is used to determine the service status of a VOQ. Although the frame size is used to determine eligibility of a VOQ to participate in the matching process, matching is performed on a time-slot basis. The value of f_m affects the performance of FORM in different traffic scenarios. The effects of using different f_m values are shown in Section 3.4. Note that when $f_m = 1$, FORM becomes 1SLIP (iSLIP, with $i = 1$). The description above presents the matching procedure for a single iteration. FORM can consider multiple iterations. However, that is beyond the scope of this paper.

3.4 Performance Evaluation

The iSLIP with one and four iterations (1SLIP and 4SLIP, respectively)on this study for comparison purposes is considered. Since the study is the performance with a single iteration, the comparison between FORM and 1SLIP is performed. The performance evaluations are produced through computer simulation. The traffic models considered have destinations with uniform and nonuniform distributions, the latter called unbalanced [32]. Both models use Bernoulli arrivals. The simulation does not consider the segmentation and re-assembly delays for variable size packets. Simulation results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay. The VOQs are assumed to have infinite capacity.

3.4.1 Uniform Traffic

Figure 3.1 shows simulation results of three 32×32 IQ switches, using the scheduling schemes: 1SLIP, 4SLIP, and FORM, and an OQ switch, all under uniform traffic with Bernoulli arrivals. This figure shows that FORM with $f_m = 2N$, as iSLIP, delivers 100% throughput under uniform traffic. FORM, with $f_m = 1$, is the equivalent of 1SLIP. Therefore, the average delay of FORM, with $F_m = 1$, is depicted by the 1SLIP curve. The desynchronization effect is also present in FORM under uniform traffic. This effect and the frame service policy allow FORM to deliver high throughput and low average cell delay under uniform traffic. The average cell delay of FORM is low as the frame consideration has an effect similar to having $f_m = 1$ and several iterations. After a frame starts being served, the VOQ in service will keep the match in a number of subsequent time slots equal to the frame size. This increases the number of matches by reducing the number of unmatched ports, resulting in a lower average delay than 1SLIP. Note that FORM shows a slightly longer average cell delay than 4SLIP. However, then the input load is 0.99, FORM has the same average cell delay. This makes FORM efficient for high input loads, with a single iteration.

Figure 3.2 shows the average cell delay of switches of different sizes, all using FORM. This case shows the results for $f_m = 2N$. It can be seen that as the switch size increases, the average cell delay increases. However, in a load close to 1.0, small switches develop a long delay. Simulation experiments showed that small switches, $N = \{4, 8\}$, have higher performance when $f_m \leq N$, and larger switches are less sensitive to the f_m value. This figure shows that the performance of FORM with an intermediate f_m value, $f_m = 2N$, for both small and large switch sizes, is high in all cases.

Figure 3.3 shows FORM with $f_m = 2N$ and an OQ switch under bursty traffic, modeled as an on-off modulated Markov process, with average burst length l. The traffic has bursts with average lengths of 16 and 32 cells ($l = 16$ and $l = 32$), and Bernoulli traffic, $l = 1$. The simulation shows that the FORM scheme provides 100% throughput

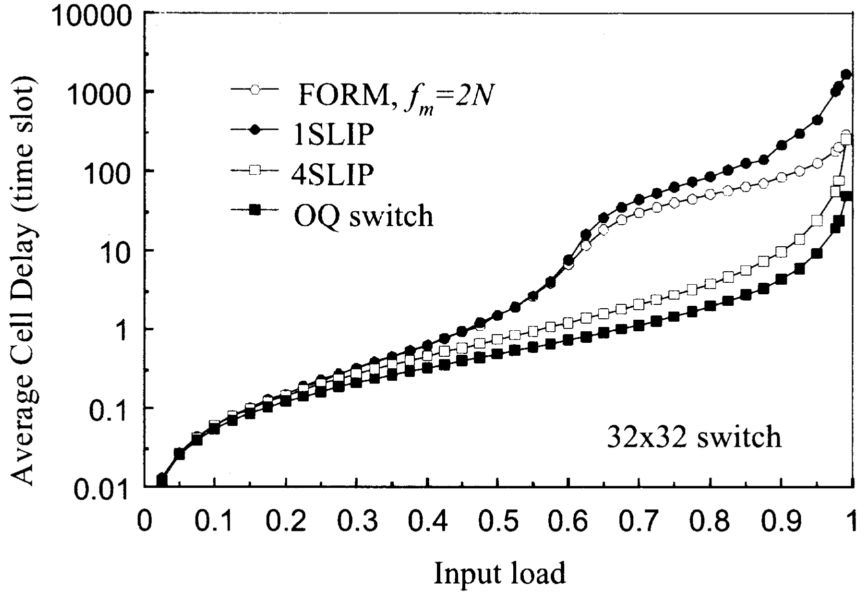

Figure 3.1 Average cell delay of FORM scheme under Bernoulli uniform traffic.

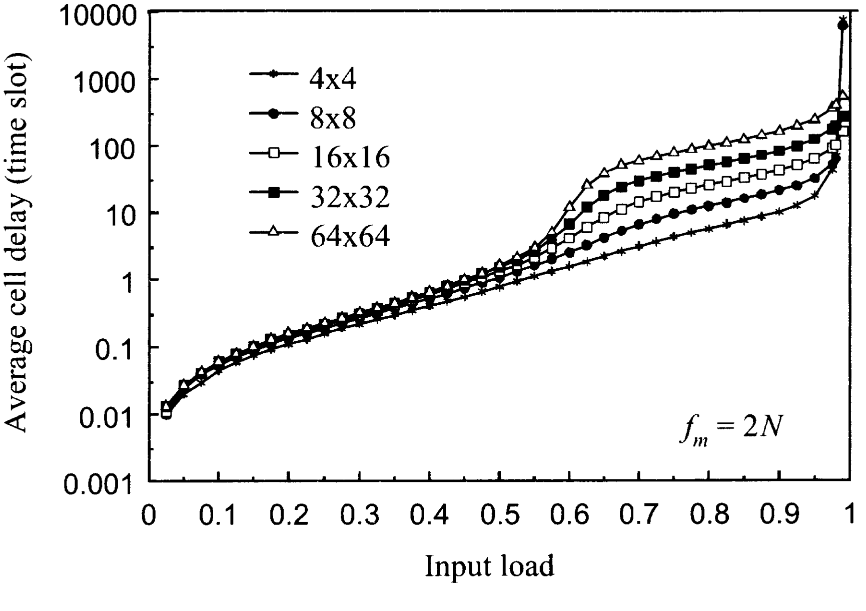

Figure 3.2 Average delay of FORM in function of switch size, under Bernoulli uniform traffic.

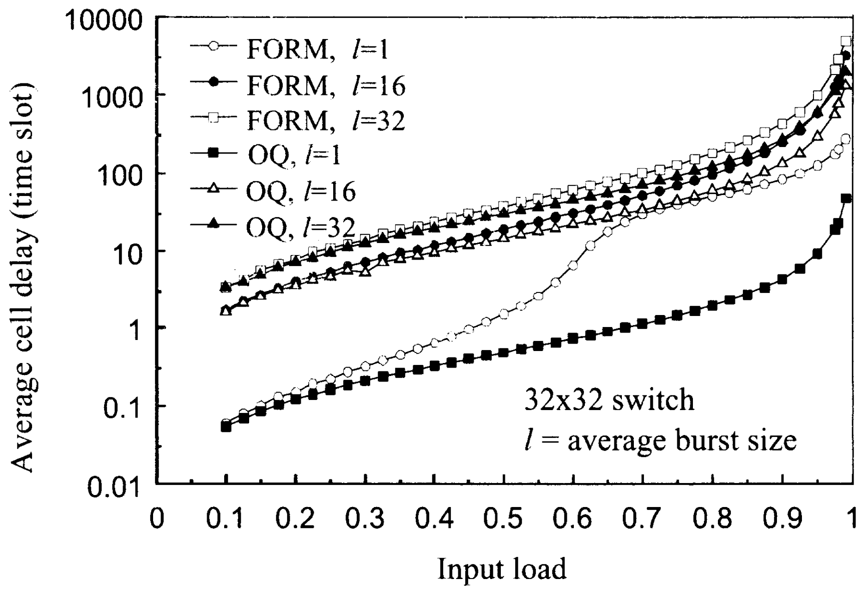

Figure 3.3 Average delay of FORM, with $f_m = 2N$, under bursty uniform traffic.

under uniform traffic under Bernoulli and bursty arrivals. The curves for $l = 16$ and $l = 32$ show a constant delay of FORM over the OQ average cell delay. This constant delay is proportional to the burst length. Therefore, switching performance is not affected by the frame concept used in FORM.

Under uniform traffic, the average frame size is small, as the uniform distribution of traffic among VOQs results in small average queue occupancies. Note that FORM does not suffer from VOQ starvation, even in the case when VOQ occupancy and f_m have large values, as the captured frame has a finite size, and the arrival of new cells does not affect the CF value arbitrarily.

3.4.2 Nonuniform Traffic

The study presented in this section uses a nonuniform traffic model, the unbalanced traffic model [32]. Three switches, of size 32, are considered under this traffic model. Each switch uses the schemes: 1SLIP, 4SLIP, and FORM. Figure 3.4 shows that FORM with $f_m = 3N$ provides over 99% throughput under the complete range of w. FORM provides an improved matching efficiency using a single iteration, compared to the other schemes.

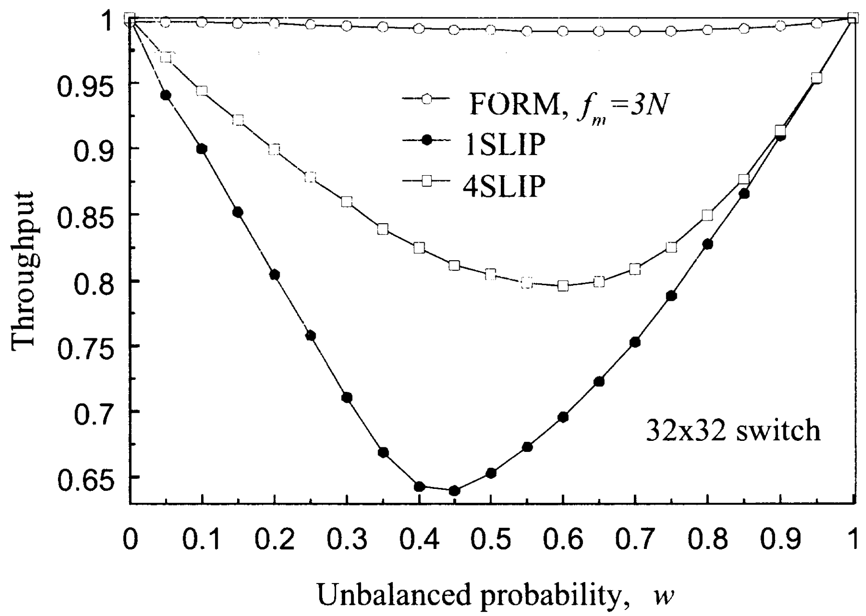

Figure 3.4 Throughput performance of FORM under unbalanced traffic.

The high throughput of FORM under this traffic model is the product of considering the VOQ occupancy. The occupancy of that queue can be expected to have a length in proportion to its received service and to the arrival rate. FORM ensures service to queues with high load by capturing a large frame size for each, and to the queues with low load by using round-robin selection.

Figure 3.5 shows a 32×32 switch with FORM under unbalanced traffic. This graph shows that for $f_m > 2N$, the throughput under unbalanced traffic is higher than 99%. Note that the lowest throughput value along the w range is the one considered.

To illustrate the dependency of N, Figure 3.6 shows the throughput of FORM for different switch sizes, $N = \{4, 8, 16, 32, 64\}$, where $f_m = 2N$ for switches of sizes $N = \{4, 8, 16\}$ and $f_m = 4N$ for switches of sizes $N = \{32, 64\}$.

The figure shows that the smaller switches offer high performance (nearly 99% throughput) when $f_m = 2N$, while larger size switches offer higher performance with rather larger values of f_m. In this case, a 32×32 switch offers a throughput above 99% under this traffic model with $f_m > 2N$. As the switch size increases, FORM is less sensitive to the f_m value

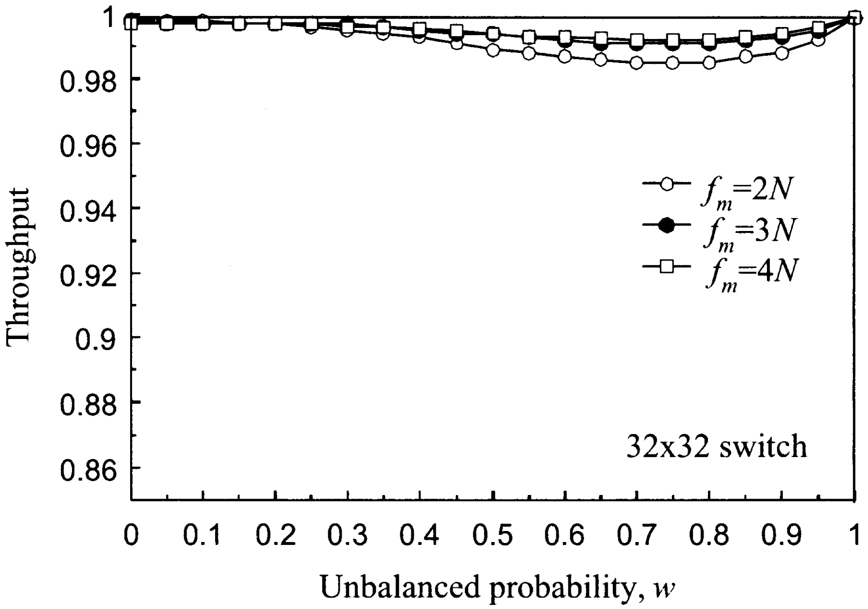

Figure 3.5 Throughput performance of a 32x32 switch for different f_m values.

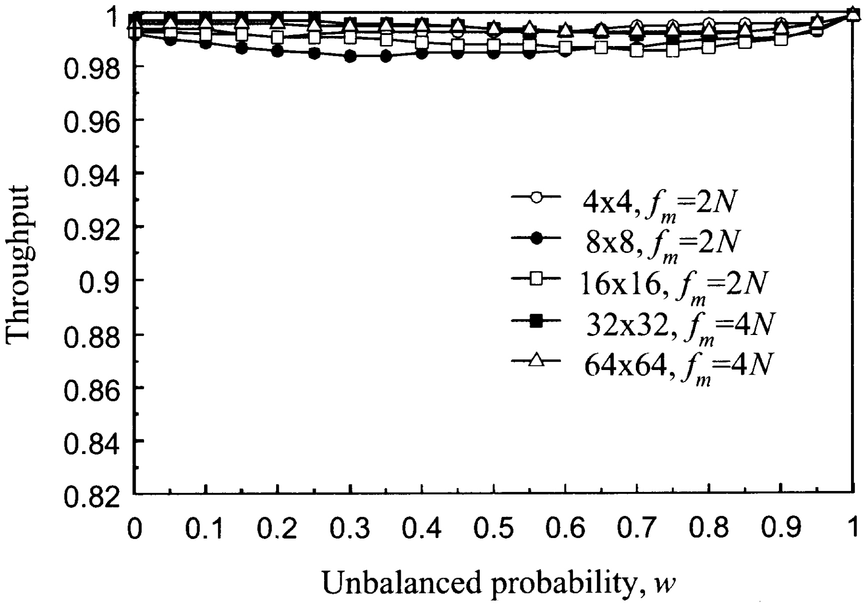

Figure 3.6 Throughput performance of FORM for different switch sizes under unbalanced traffic.

for delivering high throughput. The decreased dependency on f_m with the increase of the switch size was observed not only under unbalanced traffic, but also under uniform traffic.

3.5 Conclusions

A novel matching scheme, FORM, was introduced for IQ packet switches. This scheme is based on round-robin selection and uses the concept of captured frame size, where the frame size depends on VOQ occupancy at complete-service time. The chapter presented a study when the maximum frame size is limited to several constant values. As the switch size increases, FORM shows above 99% throughput under unbalanced traffic models, while retaining the high performance of round-robin based schemes under uniform traffic. This matching scheme does not need to compare the status of different VOQs as it is based on simple round-robin. The hardware and timing complexity of FORM is low. This makes FORM an efficient and implementable scheme.

CHAPTER 4

FRAME OCCUPANCY-BASED DISPATCHING SCHEMES FOR BUFFERED THREE-STAGE CLOS-NETWORK SWITCHES

4.1 Introduction

There are two broad approaches to implement a high-performance switch: single and multiple stages. Single-stage switches are mainly based on crossbar switch fabrics. Several single-stage high-speed switch are described in [13, 20, 48]. However, the single-stage approach makes it difficult to implement a large-scale switch, in terms of the number of ports, because a larger number of switch chips are needed to form a bi-dimensional array of chips.

A multiple-stage switch, such as a three-stage Clos-network switch [9], needs fewer switch chips for implementing a switch with large number of ports. This makes the Clos-network switch very attractive for scalable switches.[1] Two broad types of Clos-network switches are considered: bufferless and buffered. A bufferless Clos-network switch has no memory in any of the three stages. Although the design of the switch modules is rather simple, this switch may require a complex matching process and a long resolution time. A variety of matching schemes for bufferless Clos-network switches have been proposed [34, 46, 49, 50].

Within the buffered Clos-network switches, they can be categorized into two types: one has the buffers in the second-stage modules and the other has no buffers in the second-stage modules. Implementing buffers in the second-stage modules helps to resolve contention among cells from different first-stage modules [10, 51, 52]. However, switches with buffers in the second-stage modules may suffer from serving packets in out-of-sequence order, which is undesirable as re-sorting packets might increase the switch complexity and

[1] Clos-network switches also use a smaller number of crosspoint elements.

cost. The other option is to use a switch with bufferless second-stage modules, where buffers are only placed in the first and third stages. This architecture, which avoids the out-of-sequence problem, is called a memory-space-memory (MSM) Clos-network switch [12]. This definition in the remainder of this chapter is presented. By adding buffers to the first stage of the switch, a dispatching scheme needs to be used to avoid contention within the input module. This matching is implemented as a matching process.

There are several studies on matching schemes for dispatching packets from the first stage of MSM Clos-network switches. As in single-stage switches, a maximum-weight matching dispatching (MWMD) scheme has been used in MSM Clos-network switches to provide high throughput under admissible traffic [31], but the MWMD scheme has high computation complexity that could slow down high-speed switches. An alternative is to use maximal-weight matching dispatching schemes. However, the hardware and time complexity of these schemes can be considered high for the ever increasing data rates. Schemes based in round-robin dispatching matching, which are maximal-size matching schemes, such as CRRD [33], have been proposed to deliver 100% throughput under uniform traffic and with a low implementation complexity. CRRD showed that the desynchronization effect, where arbiters reach the point where each of them prefers to match with different input/outputs, improves switching performance under uniform traffic. However, CRRD has a limited throughput under some nonuniform traffic patterns.

Frame-based scheduling with fixed-size frames has been shown to improve switching performance [26]. However, how to choose a suitable frame size is complex. In this chapter, the framing is applied, based on queue occupancy [53], to improve throughput under several nonuniform traffic patterns, without allocating any buffers in the second stage to avoid the out-of-sequence problem, and to offer a low implementation complexity. Here, the frame occupancy-based random dispatching (FRD), which is based on random dispatching [12], and the frame-occupancy concurrent round-robin dispatching (FCRRD) scheme, which is based on the CRRD scheme and on the captured-frame concept [53] are presented. In this chapter, it is shown that the captured-frame concept, used for matching eligibility, improves the performance of dispatching schemes. The RD scheme and FRD for this purpose are provided. In addition, the results show that FCRRD can achieve higher throughput than that of CRRD under nonuniform traffic patterns, while retaining the high performance under uniform traffic and the low implementation complexity of round-robin schemes.

This chapter is organized as follows. Section 4.2 presents the MSM Clos-network switch model and some preliminary definitions. Section 4.3 proposes the captured frame eligibility and the frame occupancy-based round-robin dispatching scheme. Section 4.4 presents the performance study of the proposed scheme under uniform and nonuniform traffic patterns. Section 4.5 presents the conclusions.

4.2 MSM Clos-Network Switch Model and Preliminary Definition

A MSM Clos-network switch is a three-stage switch architecture [9], as Figure 4.1 shows. The chapter uses he same terminology in [33], as follows:

$IM(i)$: $(i+1)$th input module, where $0 \leq i \leq k-1$.

$CM(r)$: $(r+1)$th central module, where $0 \leq r \leq m-1$.

$OM(j)$: $(j+1)$th output module, where $0 \leq j \leq k-1$.

n: number of input/output ports in each IM/OM, respectively.

k: number of IMs/OMs.

m: number of CMs.

$IP(i,h)$: $(h+1)$th input port (IP) at $IM(i)$, where $0 \leq h \leq n-1$.

$OP(j,l)$: $(l+1)$th output port (OP) at $OM(j)$, where $0 \leq l \leq n-1$.

$VOQ(i,j,l)$: Virtual output queue at $IM(i)$ that stores cells destined for $OP(j,l)$.

$L_I(i,r)$: output link of $IM(i)$ that is connected to $CM(r)$.

$L_C(r,j)$: output link at $CM(r)$ that is connected to $OM(j)$.

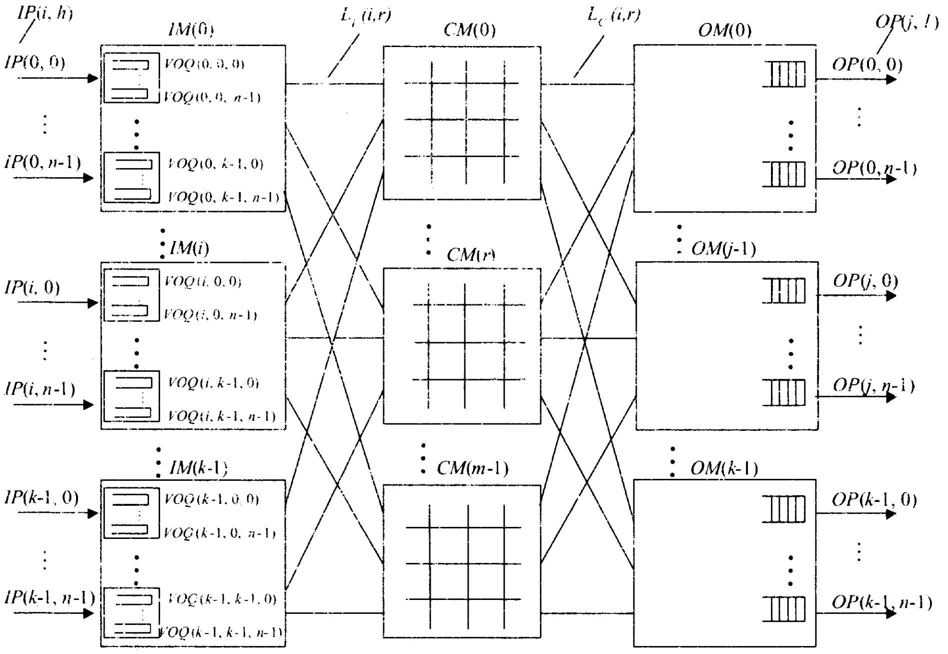

Figure 4.1 Clos-network switch with VOQs in the IMs.

The switch has k input modules (IM), m central modules (CM), and k output modules (OM). An $IM(i)$ has n input ports, each of which is denoted as $IP(i, h)$. Each $IM(i)$ has $n \times k$ VOQs to eliminate head-of-line (HOL) blocking. A $VOQ(i, j, l)$ stores cells going from $IM(i)$ to $OP(l)$ at $OM(j)$. In an IM, there are m output links. An output-link $L_I(i, r)$ is connected from $IM(i)$ to $CM(r)$. A $CM(r)$ has k output links, each of which is $L_C(r, j)$, which are connected to k OMs. An $OM(j)$ has n output ports, each of which is denoted as $OP(j, l)$, and has an output buffer.

The following definitions, adapted from [53], are used in the description of the proposed dispatching scheme.

Frame. A frame is related to a VOQ. A frame is the set of one or more cells in a VOQ that are eligible for dispatching. Only the HOL cell of the VOQ is eligible per time slot.

On-service status. A VOQ is said to be in on-service status if the VOQ has a frame size of two or more cells and the first cell of the frame has been matched.

Off-service status. A VOQ is said to be in off-service status if the last cell of the VOQ's frame has been matched or no cell of the frame has been matched. Note that for frame sizes of one cell, the associated VOQ is off-service during the matching of its one-cell frame.

Captured frame size. At the time t_c of matching the last cell of the frame associated to $VOQ(i, j, l)$, the next frame is assigned a size equal to the cell occupancy at $VOQ(i, j, l)$. Cells arriving to $VOQ(i, j, l)$ at time t_d, where $t_d > t_c$, are not considered for matching until the current frame is totally served and a new frame is captured.

4.3 Frame Occupancy-Based Concurrent Round-Robin Dispatching Scheme

In $IM(i)$, there are m output-link round-robin arbiters and nk VOQ round-robin arbiters. An output-link arbiter, which is associated with $L_I(i, r)$, has its own pointer $P_L(i, r)$. A VOQ has an arbiter associated with it. For the sake of simplicity, VOQs are re-denoted as $VOQ(i, v)$, where $v = hk + j$ and $0 \leq v \leq nk - 1$) and each VOQ has a pointer $P_V(i, v)$. In CM(r), there are k round-robin arbiters, which have their own pointer $P_C(r, j)$

For each VOQ there is a captured frame-size counter, $CF_{i,j,l}(t)$. The value of $CF_{i,j,l}(t)$, indicates the frame size at time slot t; that is, the maximum number of cells that a $VOQ(i, j, l)$ can have as matching candidates in the current and future time slots. $CF_{i,j,l}(t)$ takes a new value when the last cell of the current frame of $VOQ(i, j, l)$ is matched. $CF_{i,j,l}(t)$ decreases its count each time a cell is matched, other than the last.

The arbitration process includes two phases. This scheme follows request-grant-accept approach, as in the CRRD algorithm [33]:

Phase 1: Matching within IM

- First iteration

- Step 1: Non-empty VOQs send a request to the output-link arbiter L_I, where each request indicates the on-service or off-service status of the VOQ.

- Step 2: If an output-link arbiter receives any request, it chooses an on-service request in a round-robin fashion starting from the position of $P_L(i, r)$. If none on-service request exists, L_I chooses an off-service request in a round-robin fashion starting from the position of $P_L(i, r)$. L_I then sends a grant to the selected VOQ.

- Step 3: If the VOQ arbiter receives any grant, it accepts an on-service grant in a round-robin fashion, starting from the position of $P_V(i, v)$. If none on-service grant exists, the VOQ arbiter accepts an off-service grant that appears next in round-robin schedule, starting from the position of $P_V(i, v)$.

- ith iteration

- Step 1: Each unmatched VOQ sends another request to all unmatched output-link arbiters.

- Step 2 and 3: The same procedure is performed as in the first iteration for matching between unmatched nonempty VOQs and unmatched output links.

Phase 2: Matching between IM and CM

- Step 1: After Phase 1 is complete, $L_I(i, r)$ sends the request to $CM(r)$. Each round-robin arbiter associated with $OM(j)$ then chooses a request from the on-service $L_I(i, r)$ that appears next in a round-robin schedule, starting from the position $P_C(r, j)$ and sends the grant to $L_I(i, r)$ of $IM(i)$. $P_C(r, j)$ is updated to one position beyond the granted one. If none on-service request exists, the $OM(j)$ chooses an off-service request that appears next in a round-robin schedule, starting from its position $P_C(r, j)$ and sends the grant to $L_I(i, r)$.

- Step 2: If the $IM(i)$ receives the grant from the $CM(r)$, P_V and P_L are updated to one position beyond the granted link and VOQ, respectively. $IM(i)$ sends the corresponding cell from that VOQ at the next time slot. Otherwise, the $IM(i)$ cannot send the cell at the next time slot. The request from the $CM(r)$ that is not granted will be attempted again at the next time slot because the pointers that are related to the ungranted requests are not moved. In addition to the pointer update, the $CF_{i,j,h}$ counter updates the value according to the following:

If an $IM(i)$ received a grant from a $CM(r)$, the counters are updated as follows. If:

i) $CF_{i,j,l}(t) > 1$: $CF_{i,j,h}(t+1) = CF_{i,j,l}(t) - 1$ and this $VOQ(i,j,l)$ is set as on-service.

ii) If $CF_{i,j,l}(t) = 1$: $CF_{i,j,l}(t+1)$ is assigned the occupancy of $VOQ(i,j,l)$, and $VOQ(i,j,l)$ is set as off-service.

Note that the matching within IM can have several iterations as the arbiters can be placed in the IM modules. The matching between IM and CM is considered with one iteration only as, depending on the implementation, the IM and CM modules may be located far from each other.

4.4 Simulation Evaluation

The performance evaluations are produced through computer simulation. The simulation showed that the comparison between the performance of the RD scheme [12], and the framed version of it, FRD. Here, CRRD and FCRRD with multiple iterations in IM, which are denoted as I_{IM}, and only a single iteration between IMs and CMs are considered. FRD, as RD, assumes that up to r non-empty VOQs are matched, disregarding of the number of iterations, and therefore, the results don't indicate the number of iterations performance. The traffic models considered have destinations with uniform and nonuniform distributions and Bernoulli and bursty arrivals. The bursty traffic follows an on-off Markov modulated process and has an average burst length, l, of 10 cells. The simulation does not consider the segmentation and re-assembly delays for variable size packets. Simulation results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay.

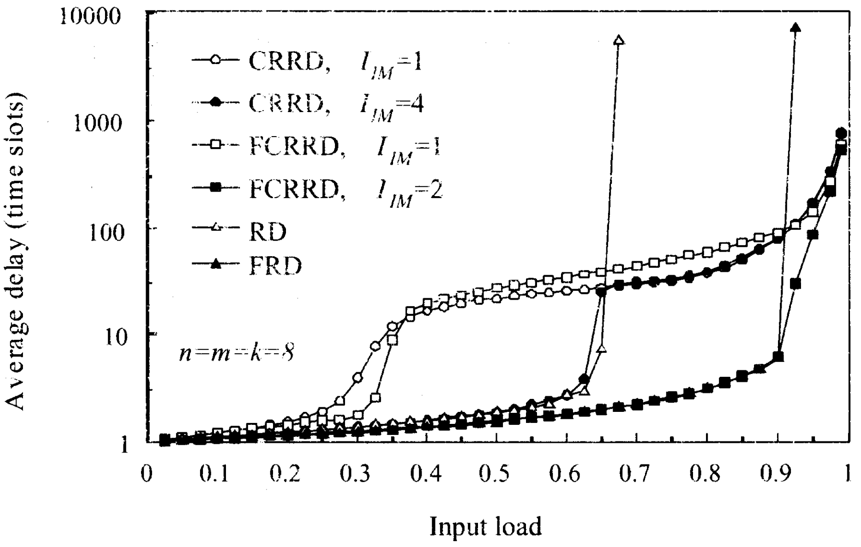

Figure 4.2 Average delay of FCRRD and CRRD schemes (n=m=k=8) under Bernoulli uniform traffic with multiple iterations in IM.

4.4.1 Uniform Traffic

Figure 4.2 shows the simulation results of RD, FRD, CRRD and FCRRD, all under uniform traffic with Bernoulli arrivals in an $n = m = k = 8$ switch (i.e., 64×64 switch without internal expansion). This figure shows that FRD delivers higher throughput than RD, and therefore, showing the improving effect of the captured-frame concept. FRD achieves 90% throughput and RD achieves 65% throughput. This figure also shows that FCRRD, as CRRD, delivers 100% throughput with any number of iterations in IM (also referred as IM iterations) under this traffic type. The average delay of FCRRD with two IM iterations is lower than that of CRRD with four IM iterations. Therefore, FCRRD converges when the number of iterations in the IM reaches two. The reason for this improvement is that once a match is achieved, the match is kept during the frame duration. Therefore, contention in IM is reduced as the number of participant VOQs is reduced with each match. Figure 4.3 shows that FCRRD provides 100% throughput under Bernoulli uniform traffic while using different switch sizes $n = m = k = \{2, 4, 8, 16\}$. The average delay increases when the switch size increases.

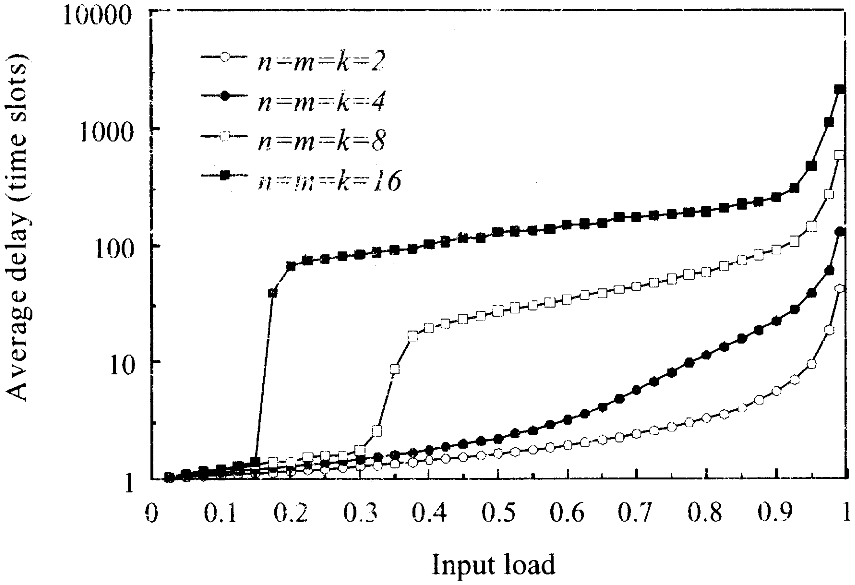

Figure 4.3 Average delay of FCRRD scheme with different switch sizes under Bernoulli uniform traffic.

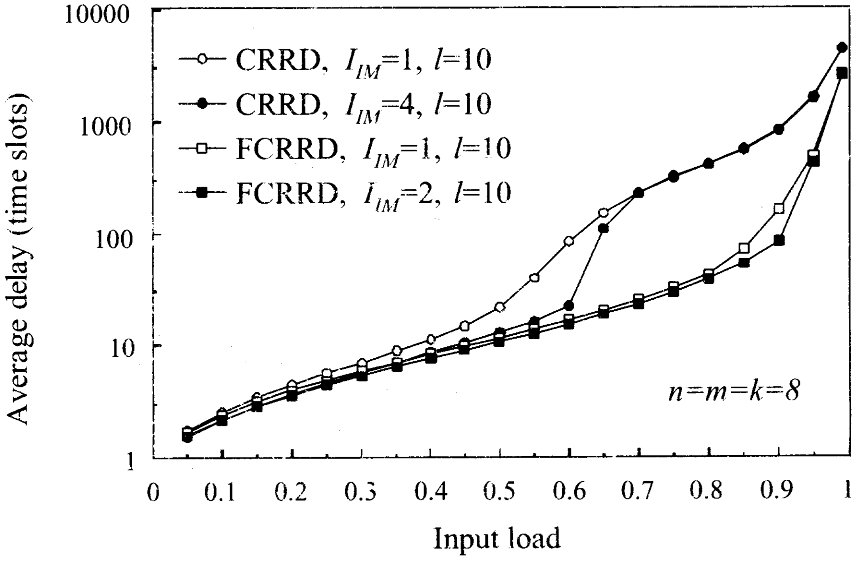

Figure 4.4 Average delay of FCRRD and CRRD schemes (n=m=k=8) under bursty traffic with multiple iterations.

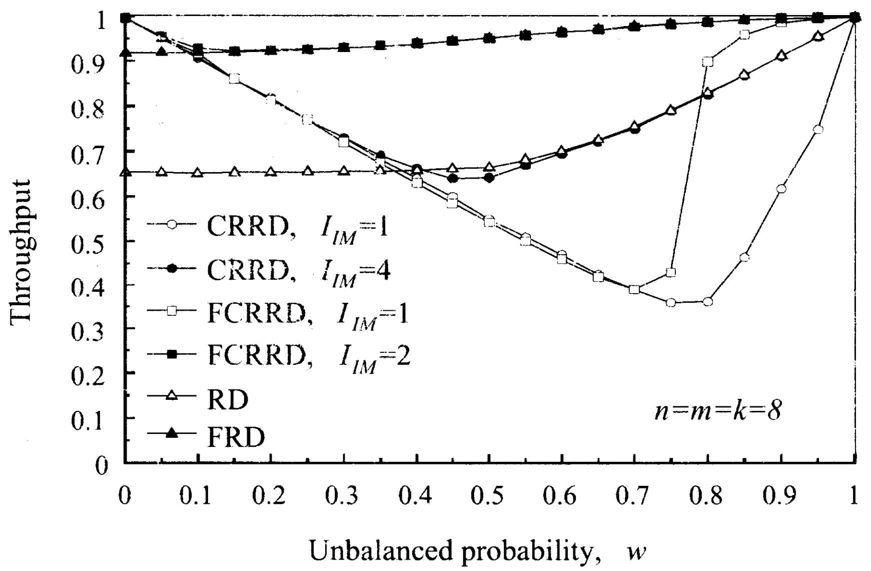

Figure 4.5 Throughput performance of FCRRD and CRRD (n=m=k=8) with multiple iterations under unbalanced traffic.

Figure 4.4 shows that FCRRD provides 100% throughput even when the input traffic is bursty, with $l = 10$. As the figure shows, the average delay of FCRRD with one and two IM iterations is smaller than that of CRRD with any number of IM iterations.

4.4.2 Nonuniform Traffic

This section shows the simulated results for RD, FRD, CRRD, and FCRRD with multiple IM iterations under different nonuniform traffic patterns: unbalanced [32], Chang's [23], asymmetric [45], and diagonal [35].

Figure 4.5 shows the throughput performance of RD, FRD, CRRD, and FCRRD under unbalanced traffic. When $w = 0$, the throughput of RD is about 65% and it increases as w increases. However, the throughput of FRD is above 90% when $w = 0$ and increases to 100% as w increases, because of the effect of the captured-frame concept. This figure also shows that the throughput of FCRRD with two IM iterations is higher than that of CRRD with four IM iterations under the complete range of w. The high throughput of FCRRD under this traffic model is the product of considering the VOQ occupancy and traffic isolation. FCRRD, as uFORM, ensures service to queues with high load by capturing

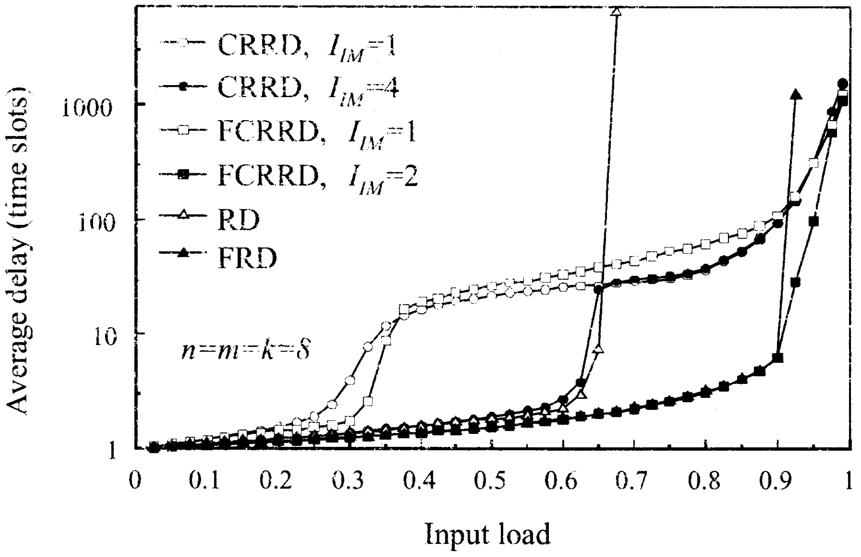

Figure 4.6 The performance of FCRRD and CRRD (n=m=k=8) under Chang's traffic.

a large frame size for each, and to the queues with low load by using round-robin selection. The captured-frame size allows the scheduler to isolate the stored cells from incoming cells that could make the queuing delay longer, as observed in single-stage switches.

Figure 4.6 shows the performance of these dispatching schemes under Chang's traffic model. This figure shows that RD delivers about 63% throughput while FRD can deliver about 91% throughput under this traffic pattern. This figure also shows that the average cell delay of FCRRD with two IM iterations is higher than that of CRRD with four IM iterations under Chang's traffic, as seen in the other traffic models.

Figure 4.7 shows the simulation results under asymmetric traffic. These results show that RD delivers about 75% throughput as CRRD with four IM iterations, while CRRD and FCRRD, both with one IM iteration, deliver below 45% throughput. FRD and FCRRD with two IM iterations provide close to 100% throughput. This figure shows that random selection is as effective as round-robin selection under this traffic model.

Figure 4.8 shows the simulation results under diagonal traffic. These results show that FCRRD, with two IM iterations, delivers higher throughput than FCRRD with one IM iteration and than CRRD with any number of iterations. Also, RD and FRD show higher

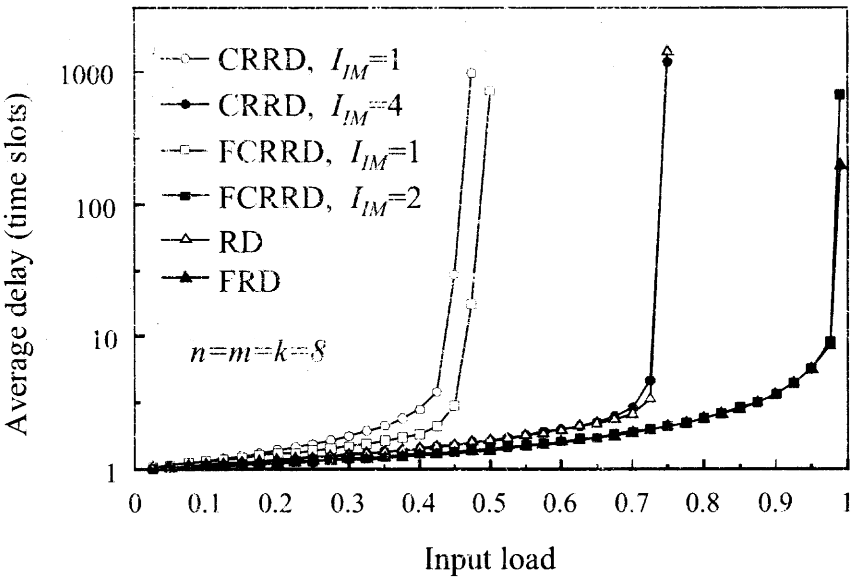

Figure 4.7 The performance of FCRRD and CRRD (n=m=k=8) under asymmetric traffic.

throughput than round-robin based schemes. The performance of FRD is comparable to that of the FCRRD with two IM iterations, of about 95% throughput. One of the reasons for this improvement is that the pointer update in round-robin schemes might not provide effective desynchronization of pointers as traffic is directed to only two different outputs per input.

The use of the captured frame and the service concepts make FRD and FCRRD deliver high switching performance under uniform and nonuniform traffic patterns. This is because when a VOQ is on-service status, service remains for the next time slots until the current frame is depleted.

4.5 Conclusions

In this chapter, two dispatching schemes for MSM Clos-network switches FRD and FCRRD are introduced. These schemes use random and round-robin selection, respectively, and the concept of unlimited captured frame-size, where the frame size depends on VOQ occupancy at completed-service time. As compared to RD and CRRD, FRD and FCRRD show higher performance under several nonuniform traffic patterns. Furthermore, the throughput

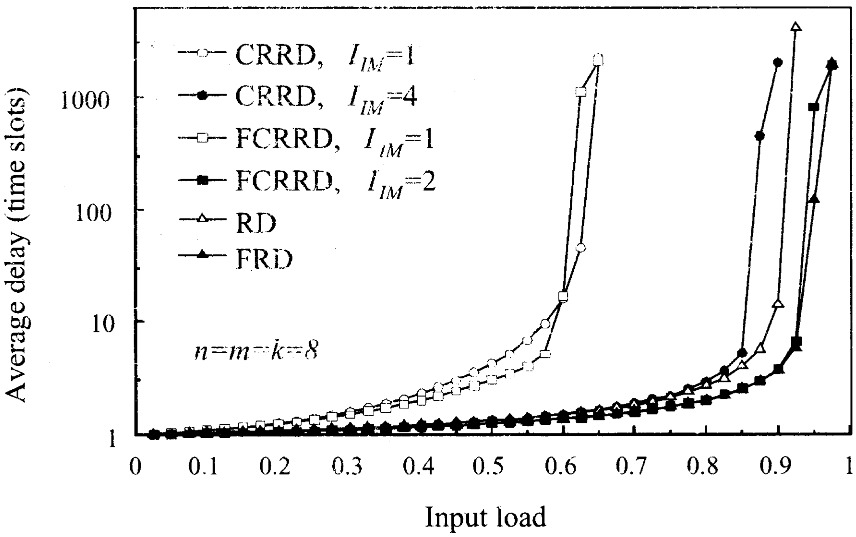

Figure 4.8 The performance of FCRRD and CRRD ($n=m=k=8$) under diagonal traffic.

of FCRRD keeps 100% under uniform traffic as that of CRRD. The result also showed that FCRRD with 2 iterations is sufficient to achieve a high switching performance. The reduction of the number of iterations is important in MSM Clos-network switches as the input modules are located in different physical locations from the central modules in a large switch.

The FRD and FCRRD schemes do not need to compare the status of different VOQs as they are based in random and round-robin selection, respectively. The hardware and timing complexity of FCRRD is comparable to that of CRRD because only the frame counters and the on/off service flags are added.

CHAPTER 5

MODULE MATCHING SCHEMES FOR INPUT-QUEUED CLOS-NETWORK PACKET SWITCHES

5.1 Introduction

The three-stage Clos-network switches use small switches as modules in each stage to build a switch with a large number of ports and less hardware than that of a single-stage switch of the same size [9]. Each of these modules can be a crossbar switch. Input-queued Clos-network (IQC) switches have queues in the input ports to store cells (variable-length packets are segmented into fixed-length packets, called cells, for internal switching) in case of input or output contention. The configuration of these switches is complex as output contention and path routing need to be resolved for every time slot before the transmission of packets occurs.

Although Clos-network switches reduce the hardware amount, in terms of the number of crosspoints, the module size, and the number of modules required to implement high-capacity packet switches, there are other issues that can limit their scalability: a) The time for configuring all modules before a packet is sent through the switch. This requires a fast packet scheduler and an efficient exchange of scheduling information among the arbiters that select the packets that are to be switched at a given time slot. b) The number of matching units(e.g., ports), as a large number of matching ports would require large arbiters implemented in hardware. For a switch with large number of ports N, a matching process involves all N ports. The implications of this are high-hardware complexity that may not allow a centralized implementation of schedulers, and long resolution times. For example, a switch with $N = 1024$, using a scheduler, with an implementation complexity of $O(N^2)$ [36] and a time complexity of $O(\log N)$, would be difficult to build. Implementation of schedulers of up to $N \leq 64$ have been reported as practical for implementation [36]. c)

Time relaxation for the configuration of all modules, or an asynchronous configuration, as each input port may be located far from each other. The time to configure the switch not only implies fast arbiters as in a). but also that the exchange of control information must travel the shortest inter-chip distances to decrease the delay overhead. For example, for a distributed implementation of a scheduler, which may be segmented into parts placed in different modules, the scheduling process may need to exchange requests and grants between different modules, placed in distinct boards.

One strategy that simplifies the configuration complexity of Clos-network switches is the use of queues in the first- and third-stage modules [33], called the MSM Clos-network switch. In this way, the scheduling of packets becomes a dispatching scheme issue [12] [46] [31] [33]. However, the queues in the first-stage modules are required to work with a speedup of $n + 1$ and those in the third-stage modules are required to work with a speedup of $m + 1$, where n is the number of input ports of the first-stage modules, and m is the number of second-stage modules. This makes it complex to build MSM Clos-networks switches.

Various matching schemes to configure IQC switches have been proposed [34] [35] [54]. Many of these schemes solve the configuration process in two phases: port matching first and routing thereafter, as routing uses the results of the port matching phase. For a 1024×1024 switch these schemes require a scheduler able to simultaneously match 1024 input ports to 1024 output ports. However, a scheduler of that size may be complex to implement [36].

The scheduler complexity is simplified for IQC switches by applying a similar concept of Clos networks used to reduce the complexity of the data path of large switches to the configuration process. It is proposed to perform matching between first- and third-stage modules first, and matching between the input and output ports of matched modules afterwards. This hierarchical approach is called as module-first matching (MoM). The longest input queue-occupancy first selection as a weight-based MoM (WMoM) selection is used to show the switching performance when using this simple configuration approach. The comparison between the switching performance of WMoM and the weightless MoM schemes based on round-robin and random selections are performed. It is shown that MoM simplifies the configuration of IQC switches. For switches with a large number of ports, say 1024, and n=m=k=32, where k is the number of first- and third-stage modules, MoM can use a scheduler size of 32 instead of 1024, and a fast 32×32 scheduler is feasible to implement. MoM can also provide high throughput under several traffic models despite its simplicity.

As a practical example, the module-first matching is used to determine the configuration of the second-stage modules and port matching for the configuration of the first-stage modules. With the configuration of the first- and second-stage modules, the third-stage modules become needless and the architecture becomes a two-stage switch. Therefore, a practical two-stage Clos-network switch is proposed. The two-stage Clos-network switch and module-first matching (MoM) are used to reduce the complexity of very large scale switches, of up to Exabit capacity, using currently feasible scheduler size.

The remainder of this chapter is organized as follows. Section 5.2 presents the three-stage IQ Clos-network switch used in the description of module-first matching scheme. Section 5.3 describes WMoM as an example of the proposed configuration scheme for IQC switches. Section 5.4 discusses the implementation of the proposed approach. Section 5.5 presents the performance evaluation. Section 5.6 presents the scalability of very large scale switches by using two-stage Clos-network switch and MoM. Section 5.7 presents the conclusions.

5.2 IQC Switch Architecture

The three-stage IQ Clos-network switch is uses virtual output queues (VOQs) in the input ports, as Figure 5.1 shows. A terminology is similar to that in [33], which is as follows:

- $IM(i)$: $(i+1)$th input module, where $0 \leq i \leq k-1$.

- $CM(r)$: $(r+1)$th central module, where $0 \leq r \leq m-1$.
- $OM(j)$: $(j+1)$th output module, where $0 \leq j \leq k-1$.
- n: number of input/output ports in each IM/OM, respectively.
- k: number of IMs/OMs.
- m: number of CMs.
- $IP(i,g)$: $(g+1)$th input port (IP) at $IM(i)$, where $0 \leq g \leq n-1$.
- $OP(j,h)$: $(h+1)$th output port (OP) at $OM(j)$, where $0 \leq h \leq n-1$.
- $VOQ(i,g,j,h)$: Virtual output queue at $IP(i,g)$ that destined for $OP(j,h)$.

There are k input modules (IM), m central modules (CM), and k output modules (OM) in the switch. IMs have a dimension of $n \times m$, OMs have a dimension of $m \times n$, and CMs have a dimension of $k \times k$. The input ports at $IM(i)$ are denoted as $IP(i,g)$. The output ports of $OM(j)$ are denoted as $OP(j,h)$. Each $IP(i,g)$ has $N = n \times k$ VOQs to avoid head-of-line (HOL) blocking. A $VOQ(i,g,j,h)$ stores cells going from $IP(i,g)$ to $OP(j,h)$.

Figure 5.2 shows the proposed two-stage Clos-network switch, which uses the same notation of the three-stage switch. However, since the third stage is removed, the OMs are not used. This architecture is called two-stage Clos-network switch as it can be derived from the original three-stage Clos-network.

5.3 Weight-Based Module-First Matching (WMoM) Scheme

This section describes MoM with a weight-based selection scheme as an example. Other selection schemes can be used by following the described process. The MoM scheme uses two classes of schedulers for matching: the module matching scheduler, S_M, which determines the matched *IM(i)-OM(j)* pairs, and the port matching scheduler, S_P, which

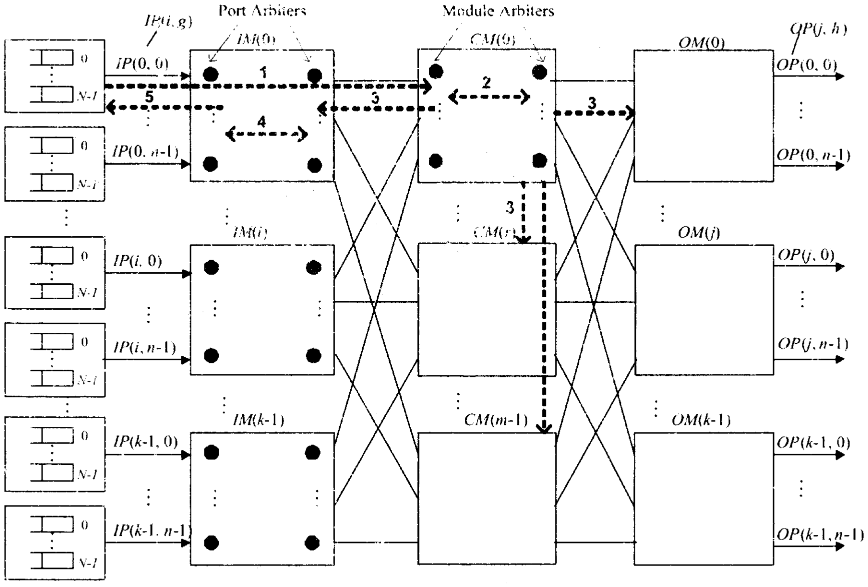

Figure 5.1 The three-stage input-queued Clos-network switch architecture.

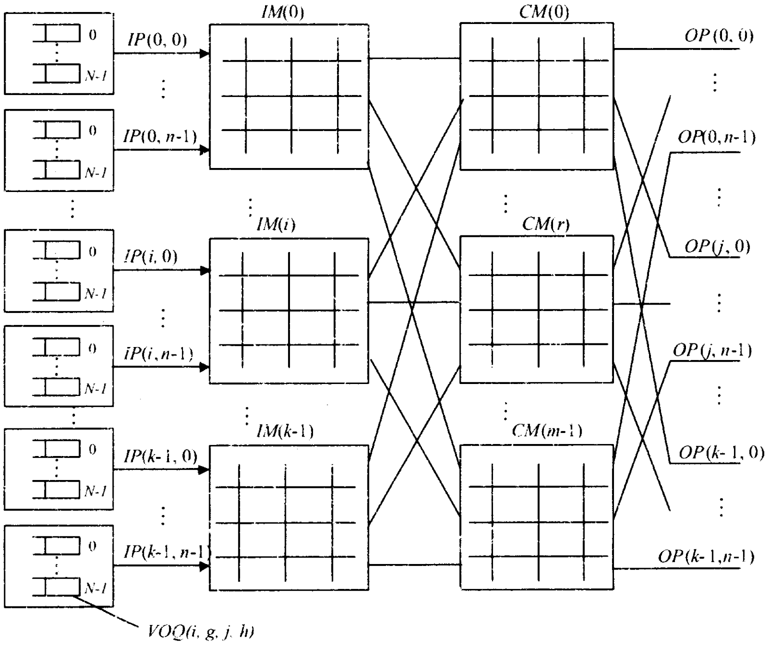

Figure 5.2 The two-stage input-queued Clos-network switch architecture.

determines the matched *VOQ(i, g, j, h)-OP(j, h)* pairs after the *IM-OM* pairs are defined. Weight-based MoM (WMoM) uses longest queue-occupancy first as the selection policy, which is similar to the *i*LQF algorithm [13] for single-stage switches. However, WMoM considers the occupancy of all ports in an IM for module matching.

To determine the weights for *IM(i)-OM(j)* matching, the IQC switch uses a VOQ module counter (VMC) to store the number of cells in $IM(i)$ going to $OM(j)$. A VMC is denoted as $VMC(i, j)$. The $VOQ(i, g, j, h) - OP(j, h)$ matching is performed after module matching. Each of the matching processes follows a request-grant-accept approach. In the general description WMoM performs I_{IM-CM} iterations of the complete scheme (e.g., module matching is executed I_{IM-CM} times, where $I_{IM-CM} \geq 1$), and q iterations for module and port matching (e.g., module matching executes q iterations, where $1 \leq q \leq k$). The following is the description of WMoM:

First iteration of WMoM (I_{IM-CM})

Part 1: Module matching: first iteration

Step 1 (request). Each VMC whose count is larger than zero sends a request to the destined output module arbiter at the S_M. Requests include the number of cells for an output-module.

Step 2 (grant). If an unmatched output-module arbiter at the S_M receives any requests, it chooses the one with the largest occupancy. Ties are broken arbitrarily.

Step 3 (accept). If an unmatched input-module arbiter at the S_M receives one or more grants, it accepts the one with the largest occupancy. Ties are broken arbitrarily.

qth iteration of module matching

Step 1: Each unmatched VMC sends a request to all unmatched output-module arbiters at the S_M, as in the first iteration.

Steps 2 and 3: The same procedure is performed as in the first iteration among unmatched VMCs and unmatched output-module arbiters.

Part 2: Port matching

After Part 1 is complete. port matching is performed between those ports of the matched IMs and OMs.

First iteration of port matching

Step 1 (Request): Each nonempty VOQ of the matched $IM(i)$ sends a request to each output arbiter in S_P for the matched $OM(j)$ for which it has a queued cell, indicating the number of cells in that VOQ.

Steps 2 (grant) and 3 (accept): The same procedure as in the module matching is performed for matching nonempty VOQs of a matched $IM(i)$ and OPs of a matched $OM(j)$. This matching is performed by input port arbiters and output port arbiters in S_Ps. These output and input arbiters select requests and grants, respectively, with the largest occupancy selection policy. Ties are broken arbitrarily.

qth iteration of port matching

Step 1: All unmatched VOQs in $IM(i)$ at the previous iterations send another request to corresponding unmatched OPs in the matched $OM(j)$ as in Step 1 of the first iteration.

Steps 2 and 3: The same procedure is performed as in the first iteration for matching between unmatched nonempty VOQs and unmatched output ports in the matched *IM(i)-OM(j)* pairs. The cumulative number of matched ports per IM and OM at this time slot are counted. The number of matched ports is smaller than or equal to n.

For $I_{IM-CM} > 1$, the number of matched ports determines the number of central modules that are used to transfer cells from $IM(i)$ to $CM(r)$ and from $CM(r)$ to $OM(j)$. The selection of modules is performed by selecting those available CMs with the smaller index. For $I_{IM-CM} = 1$. all CM paths are configured by using the module match result, which makes all CMs hold the same configuration.

I_{IM-CM}th iteration of WMoM

Part 1 is performed with those modules that have fewer than n matched ports and whose unmatched ports are non-empty, and Part 2 is performed with the non-empty unmatched ports of the modules matched at the current iteration.

5.4 Implementation of MoM

The first objective of MoM is to provide a feasible solution for performing the matching processes used to configure an IQC switch. A module scheduler with k input and output arbiters is used based on the observation of IMs and OMs matching. Since $k = \frac{N}{n}$, the size of the scheduler can be small. The same is the case for the scheduler that performs matching for the input ports of the matched IM to the output ports of the matched OM, called port scheduler. This scheduler performs a $n \times n$ matching, and therefore, it has n input arbiters and n output arbiters. There is one port scheduler in each IM and there is only one module scheduler that can be placed in one of the CMs, where IMs' requests would converge, in a distributed implementation of MoM. Figure 5.1 shows the port and module arbiters as small circles in IMs and in a CM, respectively. A centralized implementation can also be considered because of the small size of the schedulers.

A second property of MoM is the reduced number of information exchange between input ports and the module scheduler, for any number of iterations that the matching process performs. The way the information flows through the switch to perform MoM with $I_{IM-CM} = 1$ (or in an iteration) is as follows: 1) the inputs send a request to the module scheduler, 2) the module scheduler performs module matching, and if several iterations are needed, the module scheduler can perform that without using another requests from the input ports, 3) the module scheduler sends the grants to port schedulers at IMs, and to CMs (and OMs) for their configuration, 4) the port schedulers at IMs perform matching with any number of iterations, and 5) the port schedulers send a grant to the input ports, one per port. Figure 5.1 shows these steps with dashed arrows as seen by an input port. The processes are indicated with numbers over the arrows, and the arrows indicate the direction that information flows. A bidirectional arrow represents an iterative matching process.

5.5 Performance Evaluation

Three MOM schemes are modeled for simulation: WMoM, MoM with round-robin selection, and MoM with random selection to show the performance of weight-based and weightless-based schemes. It is considered $I_{IM-CM} = 1$ to show the lowest performance of these MoM schemes, and $q = \{1, 8\}$ for a fair comparison of WMoM and the other two schemes. A 256×256 Clos-network switch with *n=m=k=16* is considered. The procedures for the weightless schemes follow the steps described in Section 5.3, except for the selection policy of ports and modules. The traffic models considered have destinations with uniform and nonuniform distributions and Bernoulli arrivals. The simulation does not consider segmentation and re-assembly delays for variable size packets. Simulation results are obtained with a 95% confidence interval and a standard error not greater than 5%.

Figure 5.3 shows the average cell delay of WMoM under uniform traffic with Bernoulli arrivals. This figure shows that WMoM, as the other schemes, has low throughput with *q=1*. Round-robin delivers the highest throughput with *q=1*, however, of up to 80%. When *q=8* WMoM delivers close to 100% throughput under this traffic model, as the other schemes.

In this chapter, WMoM is simulated under four different nonuniform traffic patterns: unbalanced, Chang's, asymmetric, and diagonal. They were described in other sections.

Figure 5.4 shows the throughput performance of WMoM under unbalanced traffic. This figure shows that WMoM delivers 40% throughput, while the other schemes deliver close to 20% throughput (*w=0.9*) with *q=1*. When *q=8*, the throughput of WMoM is close to 100%, while the others remain low. The use of a large q makes WMoM match a larger number of VOQs with high occupancy. The throughput of the other schemes decrease as w increases because they do not consider the VOQ occupancy in their selection policy,

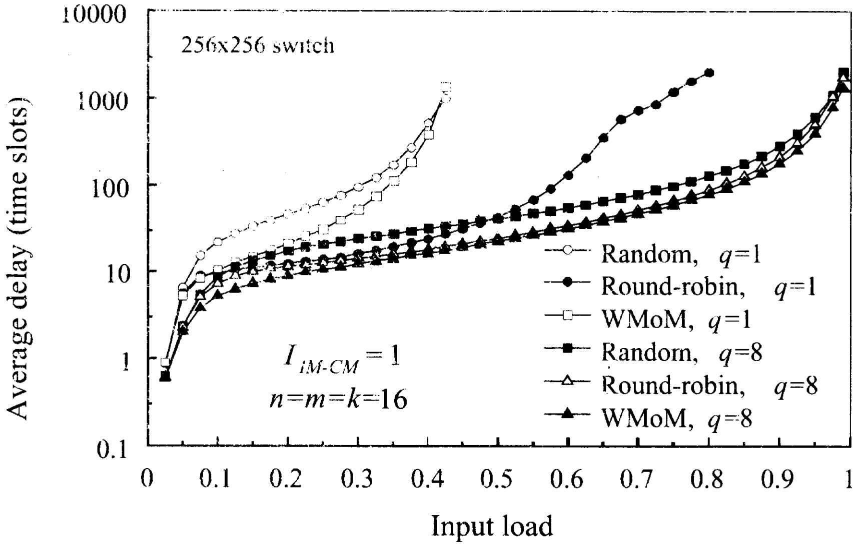

Figure 5.3 WMoM in a $n=m=k=16$ switch under Bernoulli uniform traffic.

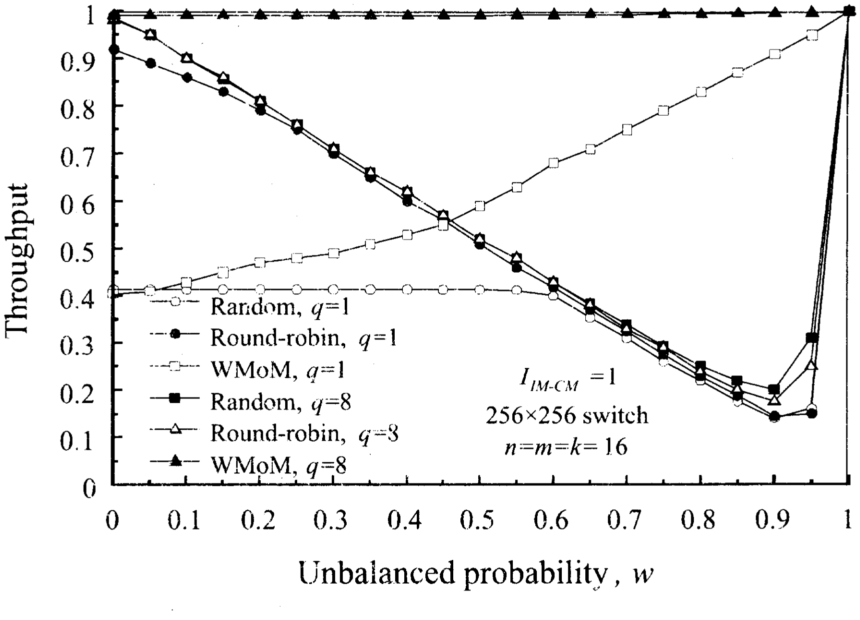

Figure 5.4 WMoM in a $n=m=k=16$ switch under Bernoulli unbalanced traffic.

and once modules are matched, the VOQs with large occupancy wait for the following opportunity to send a cell.

The simulation measured the throughput of WMoM with q=*8*, under Chang's, asymmetric, and diagonal traffic models. WMoM delivers close to 100% throughput under Chang's traffic, 91% throughput under asymmetric, and 87% throughput under diagonal. Furthermore, WMoM was tested with larger r values and it was noted that the switching performance does not increase significantly under these traffic patterns, making $I_{IM-CM} = 1$ sufficient in these cases, and therefore, greatly simplifying the configuration of CMs. However, for traffic models with a hot spot distribution, an $I_{IM-CM} = k$ may be necessary. Also, q=*8* is a large number of iterations; however, these are performed in-chip.

5.6 Scalability

Because an IM is only matched to a single OM, then all CMs have the same configuration at a given time slot. Therefore, the information coming from the module scheduler to all CMs is the same.

Figure 5.5 shows an example of configurations of a 9×9 three-stage Clos-network switch after the MoM process with a single iteration. All CMs use the same configuration obtained through module matching. In this example, $IM(0)$ is matched to $OM(1)$, $IM(1)$ is matched to $OM(2)$, and $IM(2)$ is matched to $OM(1)$. Also, $IP(0,0)$ is matched to $OP(1,2)$, $IP(1,0)$ is matched to $OP(1,1)$, and $IP(2,0)$ is matched to $OP(0,1)$. Because port matching involves only those IM-CM pairs, the configuration for such match can be done at the IMs only. As shown in this examples, OMs would be using always the same configuration (no reconfiguration, independently of the matching result), and therefore switching is not performed by them. Therefore, the three-stage switch used in the matching process is only a reference, and a two-stage Clos-network switch (whose modules are indicated by the bold line) suffices.

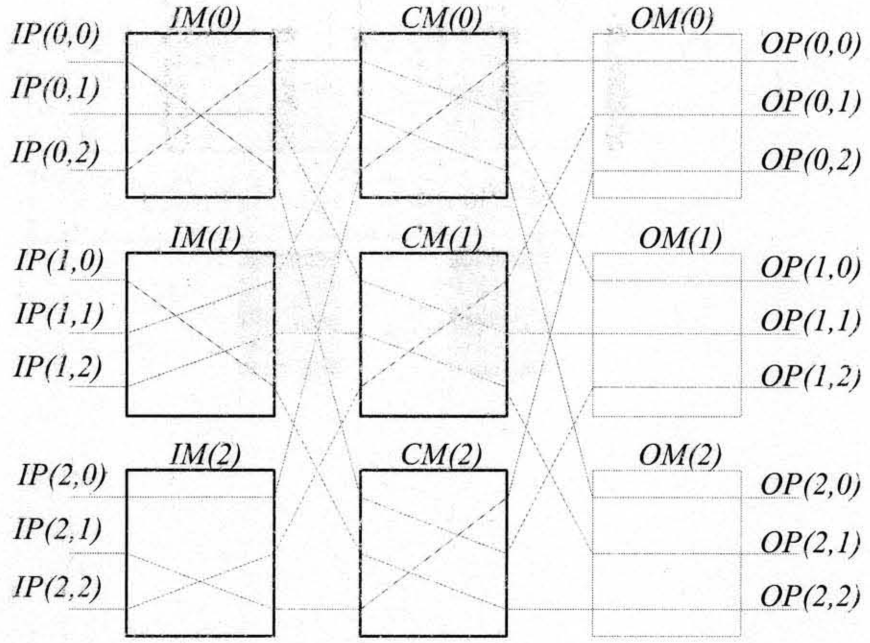

Figure 5.5 Example of configuration of the Central Modules.

The module and port arbiters might have counters to retain the number of cells in an input and VOQ, respectively. In this way, a single request can be sent from each input and VOQ to the module and port schedulers, respectively.

The reduction of scheduler sizes by module matching allows to consider the implementation of large switches. The two different strategies are considered: a) with $n = k = m$, and b) with a more flexible selection of n and m values. Table 5.1 shows an example of the component size for switches with $n = k = m$. Here, the size of the IMs/OMs and CMs are denoted by $|IM|$ and $|CM|$, respectively. The number of module scheduler is always one, and the number of port scheduler is k, and it is denoted by PS. The sizes of the module and port schedulers are denoted by $|MS|$ and $|PS|$, respectively.

Here, the maximum matching size is 64 to reduce hardware and time complexities is considered. Since the implementation issues related to cabling and distribution of a large number of chips is out of the scope of this paper, that large quantities of such elements may be acceptable.

Table 5.1 Example of Scheduler Sizes of Clos-network Switches with $n = k = m$

N	n=m=k	$\|IM\|$ = $\|CM\|$	PSs	$\|MS\|$	$\|PS\|$
256	16	16x16	16	16	16
1024	32	32x32	32	32	32
4096	64	64x64	64	64	64
16384	128	128x128	128	128	128

Table 5.2 Example of Scheduler Sizes in a Switch with Flexible Configuration

N	n	k	m	MSs	PSs	$\|MS\|$	$\|PS\|$
256	64	4	4	1	4	4	64
512	64	8	8	1	8	8	64
1024	64	16	16	1	16	16	64
2048	64	32	32	1	32	32	64
4096	64	64	64	1	64	64	64
8192	64	128	128	1	128	128	64

For switches with $n = k = m$, the number of size possibilities is rather reduced, so it can be considered a more flexible selection of n and m as Table 5.2 shows.

By the two tables above, it is clear that the switch size is limited to 4096 ports with a matching size of 64 (i.e., 64×64 schedulers). A larger number of ports would increase the size of module schedulers and the CMs, beyond the restricted value in our example. However, the module matching principle can be applied to nested Clos-network switches, used to reduce the CM sizes.

Nested Switches and Recursive Module-First Matching

Nested Clos-network switches can be seen as a recursive application of the Clos-network configuration directly into any module (e.g., IM, CM, or OM in a three-stage switch, and IM and CM in a two-stage switch) of a switch. For the sake of simplicity, let's consider that nesting is applied to CMs, and that only two levels are used (i.e., a CM has one Clos-network configuration within and the modules inside are only single-stage switches), as Figure 5.6 shows. This figure also shows the order the matching process follows, first the module matching of the internal modules in CM (modules with bold lines), then the IM-OM modules external to the CMs (bold dashed lines), and finally the port matching among matched IM-OM pairs (bold dashed ports at $IM(k-n-1)$ and $OM(n-1)$. The nested two-stage Clos-network switch is inside the large rectangle in this figure. Therefore, the architecture of CMs can use a Clos-network configuration. To apply module-first matching to nested Clos-network switches, the module-first matching among the IM and OMs that are inside a CM of the reference three-stage Clos-network switch are processed, each denoted as CM_{IM} and CM_{OM}, respectively.

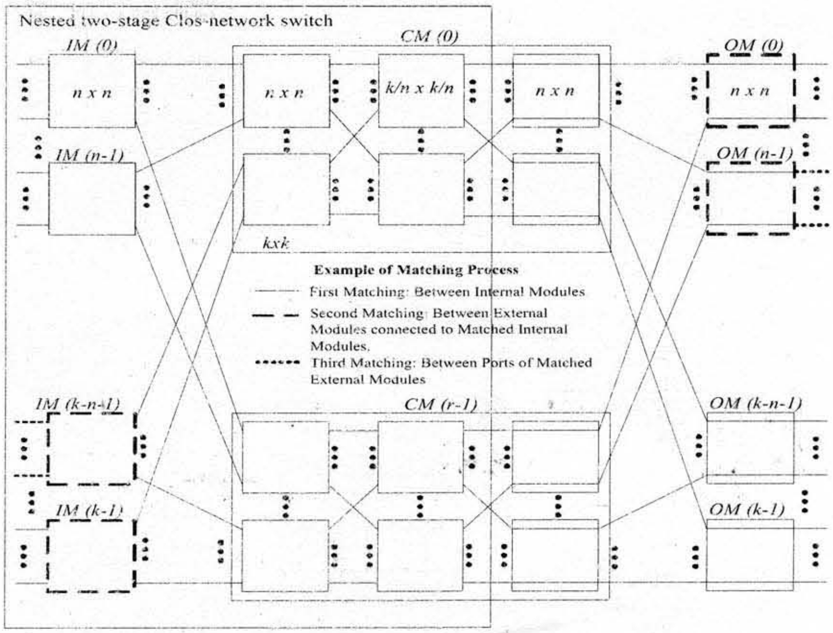
Nested two-stage Clos-network switch
IM (0)
n x n
n x n
CM (0)
k/n x k/n
n x n
OM (0)
n x n
IM (n-1)
OM (n-1)
kxk
Example of Matching Process
First Matching: Between Internal Modules
Second Matching: Between External Modules connected to Matched Internal Modules.
Third Matching: Between Ports of Matched External Modules
IM (k-n-1)
CM (r-1)
OM (k-n-1)
IM (k-1)
OM (k-1)

Figure 5.6 Nested Clos-network switch for large scale packet switches.

Table 5.3 Scheduler Sizes of Nested Clos-network Switches

N	8192	16384	32768	65536	262144
n	64	64	64	64	64
k	128	256	256	1024	4096
m	64	64	64	64	64
g_{MS}	2	4	8	16	64
$\|g_{MS}\|$	64	64	64	64	64
MSs	2	4	8	16	64
PSs	128	256	512	1024	4096
$\|MS\|$	64	64	64	64	64
$\|PS\|$	64	64	64	64	64

Table 5.3 shows the number hardware on nested Clos-networks using module-first matching. Here, it can be seen with a restricted scheduler size of 64, the maximum port count is up to 262,144. In a switch with 160 Gbps ports, module matching would allow to configure a 40 Ebps (Exabit per second) switch, in three phases.

5.7 Conclusions

A configuration scheme for input-queued Clos-network switches,called MoM, which performs module matching before port matching to reduce the configuration (and matching) complexity, was proposed. A weight-based MoM scheme, called WMoM, based on the selection of the longest VOQ occupancy first is used to describe MoM scheme and to show the obtainable performance.

The scheduler and configuration complexities for large-size switches can be reduced to $O(N^{1/2})$, where N is the number of ports. This complexity is smaller than any of the schemes previously proposed. For example, a 1024×1024 match by MoM requires parallel and independent 32×32 schedulers while other schemes require 1024×1024 schedulers.

A two-stage Clos-network switch is proposed for scalable IQ Clos-network packet switches. The reduction of the original Clos-network switch proposed for circuit switching was allowed due to the packet switching fashion. This novel two-stage switch uses the MoM scheme to reduce the configuration complexity of the very scale switches, of up to Exabit capacity.

The switching performance of several MoM schemes is presented: round-robin, random, and WMoM in a switch with *N=256*. The result showed that WMoM delivers higher performance than the other schemes under uniform and nonuniform traffic models. WMoM, using longest occupancy-queue first, provides 100% throughput under Bernoulli uniform traffic, above 99% throughput under unbalanced and Chang's traffic, 91% under asymmetric traffic, and 87% under diagonal traffic.

CHAPTER 6

SCALABLE AND PRACTICAL SPACE-SPACE-MEMORY CLOS-NETWORK PACKET SWITCHES

6.1 Introduction

The design of the switch modules for an IQC switch is rather simple, however, the required configuration process may be complex as output contention and path routing need both to be resolved for every time slot before the transmission of cells occurs. The previously configuration schemes to reduce the scheduler size and configuration complexity by grouping ports that belong to a module and performing module matching have been proposed in Chapter 5 [37, 38, 54]. These schemes perform module matching first and port matching thereafter. However, these schemes require more than one iteration in the module matching process to achieve an acceptable performance. This number of iterations can accumulate a long processing delay as in each module matching iteration the matching information (e.g., requests and grants) travels from IMs to CMs and back to IM (for port matching) in an scheduler implementation that follows a distributed approach. As IMs and CMs are located in the different places, the exchange of information between arbiters in different modules requires long resolution delay, which can jeopardize the switch scalability.

In this chapter, the reduction of iterations of information exchange between IMs and CMs is addressed by proposing a novel three-stage Clos-network switch architecture that uses buffers in the crossbars of the third-stage modules. This switch is called space-space-memory (SSM) Clos-network switch. The SSM Clos-network switch can reduce the complexity of the configuration process as output contention is resolved by the crosspoint buffers in the output modules. Speedup is not required for the memory in the third-stage modules of the SSM Clos-network switch [32, 55, 56].

In addition, two weighted configuration algorithms for this SSM Clos-network switch are proposed. One is the weighted module-first and none-port matching scheme (WMF-NP) scheme, the other is the weighted central modules' link matching (WCMM) scheme. The two approaches reduce the configuration complexity and the number of IM-CM iterations as more IM-CM iterations produce long resolution delay. As these schemes are designed for the SSM Clos-networ switch, the two schemes include output arbitrations over the buffers in the third-stage modules, and have no matching processes between input and output ports. This is an advantage over the previously proposed weight-based module-first matching scheme for IQC switches [38]. Furthermore, the simulation results show the high performance of the proposed approaches using weight-based schemes under uniform and nonuniform traffic.

The remainder of this paper is organized as follows. Section 6.2 presents the switch architecture used in this chapter. Section 6.3 describes the proposed configuration schemes, WMF-NP and WCMM, and the examples of WMF-NP and WCMM. Section 6.4 discusses implementation details. Section 6.5 presents the performance evaluation. Section 6.6 presents the conclusions.

6.2 SSM Clos-Network Switch Architecture

The SSM Clos-network switch is a three-stage switch architecture with crosspoint buffers in the third-stage modules and virtual output queues (VOQs) at the input ports as shown in Figure 6.1. The following terminology is used, as in [33]: $IM(i)$: $(i+1)$th input module, where $0 \leq i \leq k-1$, $CM(r)$: $(r+1)$th central module, where $0 \leq r \leq m-1$, $OM(j)$: $(j+1)$th output module, $0 \leq j \leq k-1$, n: number of input/output ports in each IM/OM, k: number of IMs/OMs, m: number of CMs, $IP(i,g)$: $(g+1)$th input port (IP) at $IM(i)$, where $0 \leq g \leq n-1$, $OP(j,h)$: $(h+1)$th output port (OP) at $OM(j)$, where $0 \leq g \leq n-1$, $L_I(i,r)$: $(r+1)$th output link at $IM(i)$ that is connected to $CM(r)$, $L_C(r,j)$: $(j+1)$th output link that is connected to $OM(j)$, $VOQ(i,g,j,h)$: virtual output queue at $IP(i,g)$

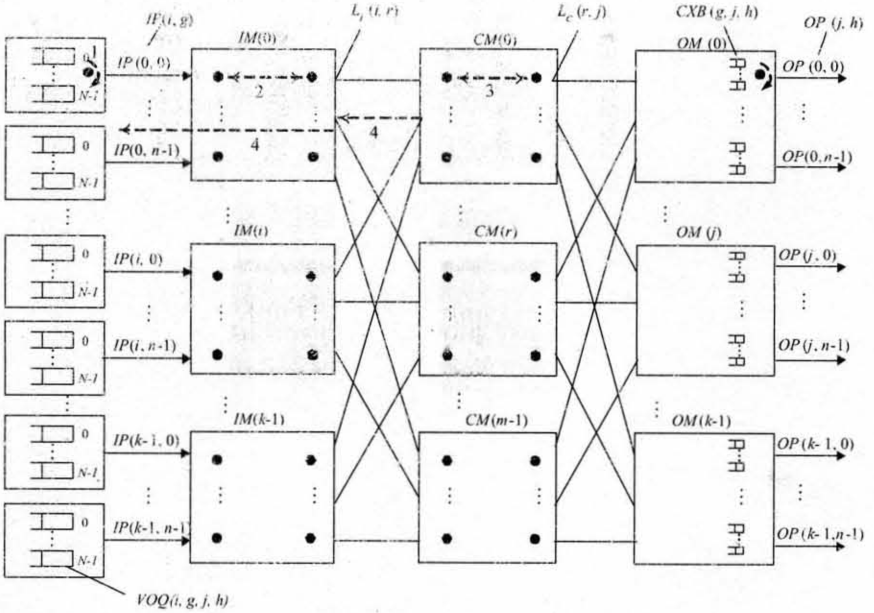

Figure 6.1 The SSM Clos-network switch architecture.

that stores cells destined for $OP(j,h)$, and $CXB(g,j,h)$: crosspoint buffer at $OM(j)$ that stores cells at $CM(r)$ destined to $OP(j,h)$. Each $IP(i,g)$ has $N = n \times k$ VOQs to avoid head-of-line (HOL) blocking [8]. Each $OM(j)$ has $N = n \times k$ crosspoint buffers.

6.3 Weighted Module-First and None-Port Matching (WMF-NP) and Central Modules' Link Matching (WCMM) Schemes for SSM Clos-Network Switch

In this section, two weight-based matching schemes for the SSM Clos-network switch are introduced. One is the weighted module-first and none-port matching (WMF-NP) scheme, the other is the weighted central module's link matching (WCMM) scheme. The WMF-NP scheme uses a scheduler S_M for module matching and arbiters at IM output-links, CM output-links, inputs, and outputs to associate the matching processes. WCMM also has arbiters as in WMF-NP and uses m CM-link schedulers instead of a module-matching scheduler. These two schemes use the longest output module queue-occupancy first (LOQF) as the selection policy. This scheme is similar to iLQF [13], however, the selection policy is executed at the output-module level. A flow control mechanism is used to indicate VOQs about the available room at $CXB(g, j, h)$ (or CXB occupancy) and to determine which VOQ can be considered for arbitration and for forwarding a cell to it.

6.3.1 WMF-NP Matching Scheme

WMF-NP uses the module-matching first approach [38], using weights assigned by the occupancy of an input module for an output module. To determine the weight for the $IM(i) - OM(j)$ matching, a VOQ module counter $VMC(i, j)$ is used to count the number of cells in $IM(i)$ that are destined to $OM(j)$ (i.e. $VMC(i, j) = \sum_{g=0}^{n-1} \sum_{h=0}^{n-1} |VOQ(i, g, j, h)|$), where VOQ occupancy is denoted as |VOQ|. The matching process follows a request-grant-accept approach. In general, the switch performs multiple iterations in the matching between inputs and IM output-link arbiters, and between IM and OM modules. Each $L_I(i, r)$ has an available/matched flag $FL_I(i, r)$ and each $L_C(r, j)$ has an available/matched flag $FL_C(r, j)$. These flags indicate whether a link (and therefore the configuration of $CM(r)$) is selected or not. These flags are used to define eligibility of an OM in the module-matching phase. $OM(j)$ is considered eligible to match $IM(i)$ if at least there is one path (and $L_I(i, r_1)$ and $L_C(r_2, j)$, where $r_1 = r_2$) is available connecting these two modules. The following is the description of WMF-NP matching scheme:

First iteration of WMF-NP (I_{IM-CM})

Part 1: Module matching: first iteration

- Step 1. *Request.* Each VMC whose count is larger than zero sends a request to the destined and eligible output-module arbiter at the S_M. Requests include the number of cells for an output module as weight.

- Step 2. *Grant.* If an unmatched output module arbiter at the S_M receives any requests, it chooses the one with the largest weight. Ties are broken arbitrarily.

- Step 3. *Accept.* If an unmatched input module arbiter at the S_M receives one or more grants, it accepts the one with the largest occupancy. Ties are broken arbitrarily. The FL_I and FL_C flags of the matched links are set as matched.

qth iteration of module matching

- Step 1. Each unmatched VMC sends a request to all unmatched output-module arbiters at the S_M, as in the first iteration.

- Steps 2 and 3. The same procedure is performed as in the first iteration among unmatched VMCs and unmatched output-module arbiters.

Part 2: VOQ selection and Matching within IM

The VOQ selection and the Matching in IM are processed in parallel (or else, VOQ is selected first and IM matching thereafter). After Part 1 is complete, each input arbiter selects a non-empty VOQ for the matched $OM(j)$ by using LQF selection policy. Each L_I is matched to an input. Step 1. Each input with cells to $OM(j)$ sends a request to all (k) $L_I(i,r)$ arbiters. Step 2. Each L_I arbiter selects the request of an input whose weight is the largest and sends a grant to the input. Step 3.Each input accepts one grant.

This matching needs to perform m iterations if LQF is used, among those unmatched L_I and inputs.

I_{IM-CM}th iteration of WFM-NP

Perform module matching (Part 1), and VOQ selection and matching within IM (Part 2) with those modules that have one or more unmatched and non-empty input ports and available paths between unused IM output-link $L_I(i,r_1)$ and CM output-link $L_C(r_2,j)$, for instance, r_1 of IM output-link $L_I(i,r_1)$ is equal to r_2 of CM output-link $L_C(r_2,j)$, where r_1 and r_2 are the index of CM (i.e., $CM(r)$).

Figure 6.2 shows an example of the available paths between an unused IM output-link $L_I(i,r_1)$ and a CM output-link $L_C(r_2,j)$ at the I_Mth iteration in an $n=m=k=3$ SSM Clos-network switch. The dark circles represent the IMs and CMs, and the blank

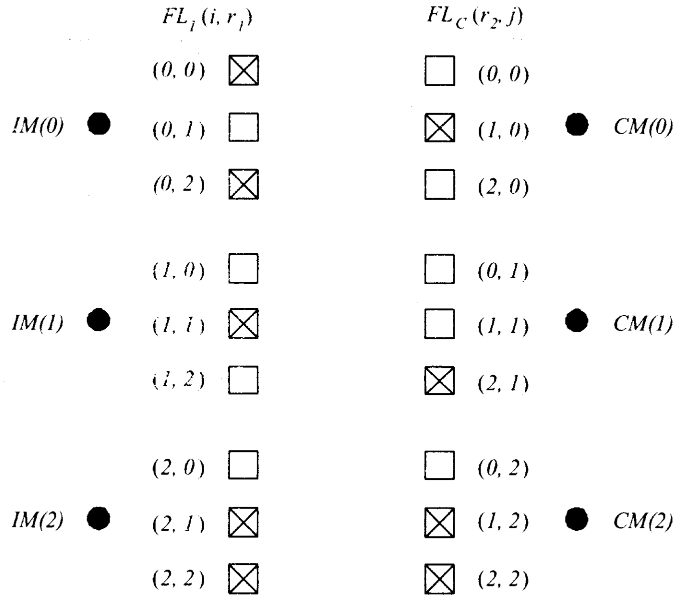

Figure 6.2 The example of the available paths between unused IM output-link L_I and CM output-link L_C in a (n=m=k=3) SSM Clos-network switch.

and crossed squares denote available and matched flags $FL_I(i, r)$ of L_I and $FL_C(r, j)$ of L_C, respectively. This example shows an available path between $L_I(0, 1)$ and $L_C(1, 1)$, therefore, then $IM(0)$ can be matched to $OM(1)$.

Output selection

Output arbiters at each output port in OMs use the LQF policy to select a buffered cell among non-empty crosspoint buffers to forward a cell to the output port.

6.3.2 WCMM Matching Scheme

The WCMM matching scheme includes two matching phases. In Phase 1, a matching process between input arbiters and IM output-link arbiters $A^E_{IM}(i, r)$ is performed within the IM. WCMM employs an iterative matching by using LOQF selection to assign an input to an IM output link. In Phase 2, WCMM performs an iterative matching between IM output-link arbiters $A^I_{CM}(r, i)$, where I indicates the ingress link of $CM(r)$ connected to $IM(i)$, and CM output-link arbiters $A^E_{CM}(r, j)$, where E indicates that is the egress link of $CM(r)$ connected to $OM(j)$, within the CM. The selection policy in WCMM is also based on LOQF. To determine the weight for $IP(i,g)-L_I(i,r)$ matching within the IM and $L_I(i,r) - L_C(r,j)$ matching within the CM, a VOQ module counter $VOMC(i,g,j)$(i.e. $|VOMC(i,g,j)| = \sum_{h=0}^{n-1} |VOQ(i,g,j,h)|$), where the VOQ occupancy is denoted as |VOQ|, which counts the number of cells in $IP(i,g)$ that are destined to $OM(j)$. Each of the matching processes follows a request-grant-accept approach. The following is the description of WCMM.

Phase 1: Matching within IM: first iteration

- Step 1. Each inputs selects the non-empty VOMC with the largest occupancy and sends a request to every output-link L_I arbiter $A^E_{IM}(i,r)$, where each request indicates the number of cells of the selected VOMC as weight value.
- Step 2. If an output-link arbiter A^E_{IM} receives any requests, it chooses a request with the largest weight. Ties are broken arbitrarily.
- Step 3. If the input arbiter receives any grants, it accepts a grant with the largest occupancy. Ties are broken arbitrarily.

qth iteration of matching within IM

- Step 1. Each unmatched inputs sends another request to all unmatched output-link arbiters.
- Step 2 and 3. The same procedure is performed as in the first iteration for matching between unmatched inputs and unmatched output links.

Phase 2: Matching within CM: first iteration

- Step 1. After Phase 1 is complete, each $VOMC(i,g,j)$ in $IP(i,g)$ matched with $L_I(i,r)$ sends a request to its destined output link L_C arbiter $A^E_{CM}(r,j)$. Requests include the number of cells for an output link L_C.

- Step 2. If an L_C arbiter $A^E_{CM}(r, j)$ receives any requests, it chooses a request with the largest occupancy. Ties are broken arbitrarily.

- Step 3. If the L_I arbiter $A^I_{CM}(r, i)$ receives any grants, it accepts a grant with the largest occupancy. Ties are broken arbitrarily.

qth iteration of matching within CM

- Step 1. Each unmatched VOMC sends another request to all unmatched output-link arbiters $A^E_{CM}(r, j)$.

- Step 2 and 3. The same procedure is performed as in the first iteration for matching between unmatched VOMCs and unmatched output links L_C.

Input and output selection

After Phase 1 and 2 are completed, each input arbiter selects a VOQ with largest occupancy from a matched VOMC and sends a cell to the $CXB(g, j, h)$. Then, an output arbiter uses the LQF policy to select a buffered cell among non-empty crosspoint buffers to forward a cell to the output port.

Figure 6.3 shows that L_I links in group A from different IMs can only be matched to L_C links at a same CM. This can be resolved by separating groups that perform matched in parallel. The small circles are denoted as L_I and L_C arbiters at CMs.

Figure 6.4 shows an example of the first matching phase in the WCMM scheme. The VOMC values show the sums of VOQ cells in $IP(i, g)$ that are destined to $OM(j)$. Each input selects the largest VOMC to send a request to every L_I arbiter with VOMC value. In $IM(0)$, $IP(0, 0)$ and $IP(0, 1)$ select the $VOMC(0, 0, 0)$ and $VOMC(0, 1, 1)$, respectively, and send a request to $L_I(0, 0)$ and $L_I(0, 1)$ arbiters with VOMC value of 5 and 6 cells, respectively. In $IM(1)$, $IP(1, 0)$ select the $VOMC(1, 0, 0)$ and sends a request to $L_I(1, 0)$ and $L_I(1, 1)$ arbiters with VOMC value of 3, respectively. The L_I and inputs arbiters select a request or grant by the largest VOMC values among all requests (grants)

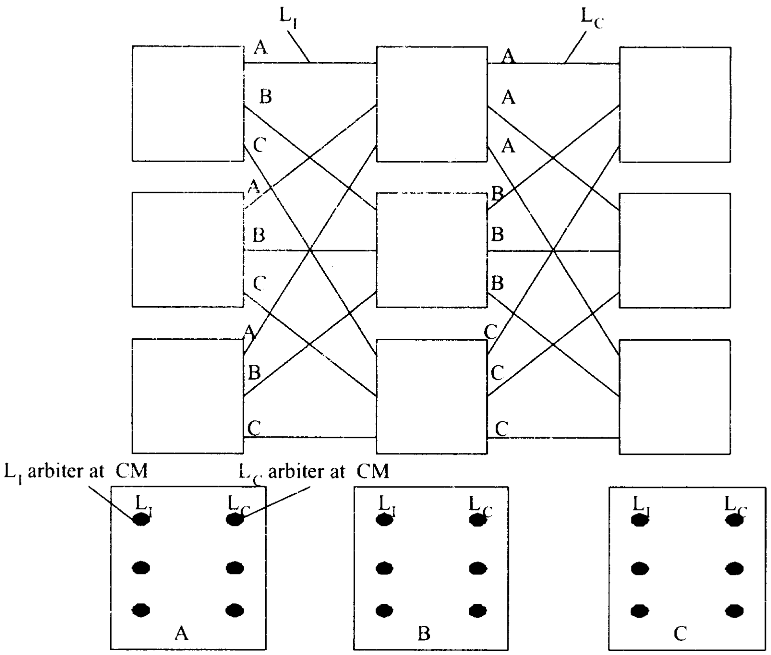

Figure 6.3 The example of matching process of the WCMM scheme in a (n=m=k=3) SSM Clos-network switch.

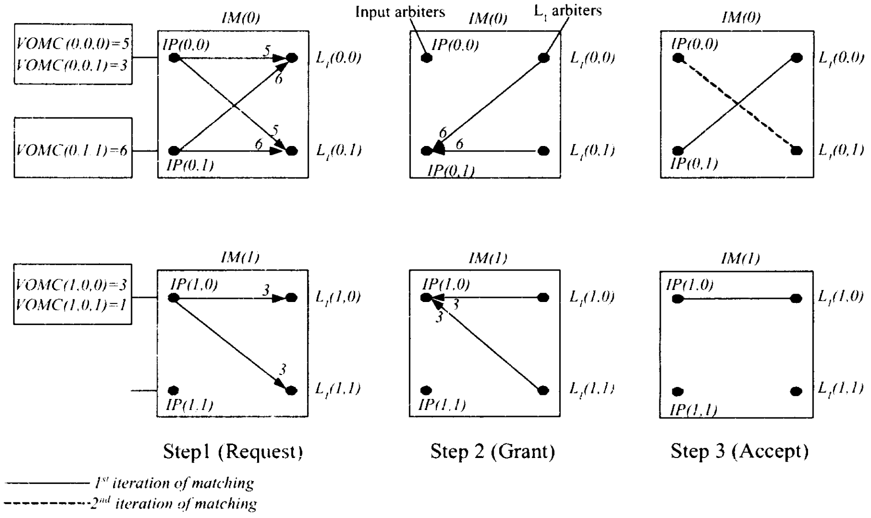

Figure 6.4 The example of matching within IM for the WCMM scheme in a (n=m=k=2) SSM Clos-network switch.

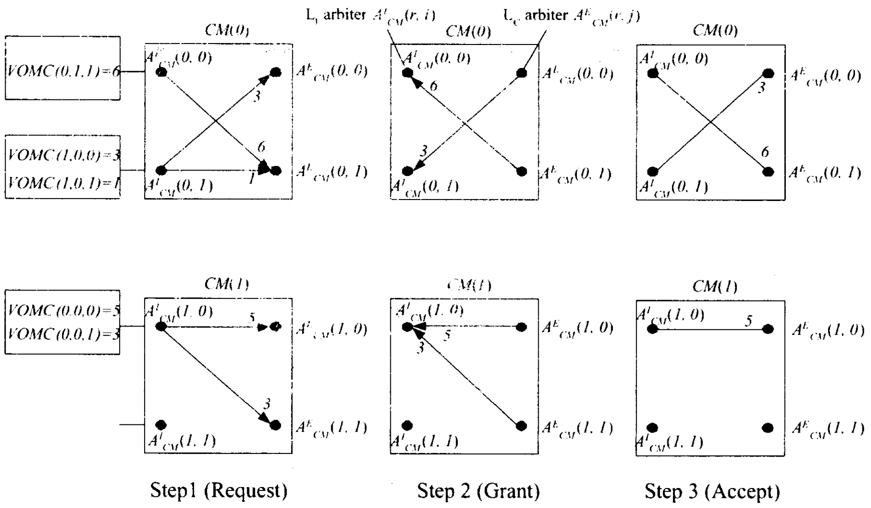

Figure 6.5 The example of matching within CM for the WCMM scheme in a (n=m=k=2) SSM Clos-network switch.

of each arbiter, as shown by the grant and accept steps. $L_I(0,0)$ selects the request with VOMC value of 6 from $IP(0,1)$ over the request with VOMC value of 5 from $IP(0,0)$, and $L_I(1,0)$ selects the request with value of 3 from $IP(1,0)$. In the second iteration, the same procedure is performed as in the first iteration for matching between unmatched inputs and unmatched output links. $L_I(0,1)$ is matched with $IP(0.0)$ in the second iteration, as shown in the accept step, indicated by the dashed line.

After Phase 1 is completed, WCMM performs Phase 2: matching within CM, as shown in Figure 6.5. In the request step, $VOMC(0,1,1)$, $VOMC(1,0,0)$, $VOMC(1,0,1)$, $VOMC(0,0,0)$, and $VOMC(0,0,1)$ send requests to their destined output link output link L_C arbiter $A^E_{CM}(r,j)$, respectively. In the grant step, $A^E_{CM}(0,1)$ receives two requests, and selects $A^I_{CM}(0,0)$ because that it has larger VOMC value according LOQF selection policy. $A^E_{CM}(0,0)$, $A^E_{CM}(1,0)$, and $A^E_{CM}(1,1)$ receive a single request, therefore, the requests are granted. In accept step, $A^I_{CM}(1,0)$ selects $A^E_{CM}(1,0)$ by using LOQF policy. $A^I_{CM}(0,0)$ accepts the single grant issued by $A^E_{CM}(0,1)$. $A^I_{CM}(0,1)$ accepts the single grant issued by $A^E_{CM}(0,0)$. After Phase 1 and 2 are completed, input arbiters select the largest number of VOQ cells from the matched VOMCs.

6.4 Implementation of WCMM

Figure 6.1 shows the input, output, L_I output-link, and L_C output-link arbiters as small circle in inputs, OMs, IMs, and CMs, respectively, in a distributed-approach implementation. The WCMM process is as follows: 1) the input arbiter selects a largest $VOM(i, g, j)$ and sends a request to each L_I output-link arbiter, 2) the input and L_I output-link arbiters perform the matching in IMs with q iterations, 3) the L_I and L_C output-link arbiters perform the matching in CMs with q iteration, and 4) the CM send the grant to IMs and inputs. Figure shows these numerated steps indicated by dashed-line arrows. The processes are indicated with a number below the arrows, and the arrows indicate in what direction the information flows. A bidirectional arrow represents an iterative matching process.

6.5 Performance Evaluation

The performance of the proposed scheme is studied by using computer simulation. The performance of the WMoM [38] in a bufferless Clos-network switch, and of the WMF-NP and WCMM in an SSM Clos-network switch are compared. The number of iterations for module matching, which mean that the information travels between arbiters in IMs and CMs, is denoted as I_{IM-CM}. Due to the same number of iterations for port matching, the matching within IM, and the matching within CM, the indication for those iterations in the following figures is denoted as q. The crosspoint buffer size in OMs is denoted as B. It is considered a 256×256 Clos-network switch with $n = m = k = 16$ to show the performances of these schemes. The traffic models considered have Bernoulli arrivals with destinations with uniform and nonuniform distributions. The simulation does not consider the segmentation and re-assembly delays for variable size packets. Simulation results are obtained with a 95% confidence interval, not greater than 5% standard error for the average cell delay.

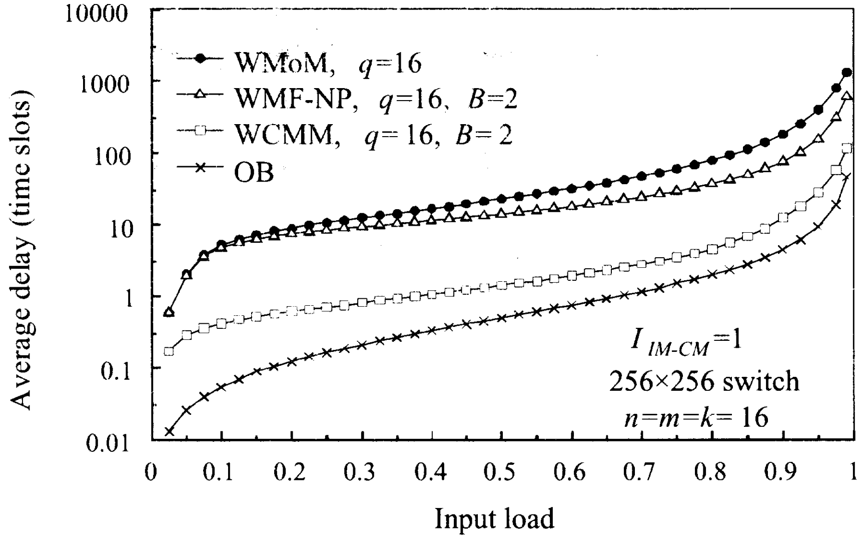

Figure 6.6 Average delay of WMoM, WMF-NP and WCMM schemes *(n=m=k=16)* and an OB switch (256 × 256) under Bernoulli uniform traffic.

6.5.1 Uniform Traffic

Figure 6.6 shows the simulation results of 256 × 256 switches with WMoM, WMF-NP, and WCMM schemes and of an output-buffered (OB) switch, all under uniform traffic with Bernoulli arrivals. This figure shows that the WCMM and WMF-NP can achieve 100% throughput with $q = 16$ and $B = 2$, as WMoM does, under uniform traffic. The figure also shows that the average delay performance of WCMM under this traffic is close to that of an OB switch, and WMF-NP is lower than that of WMoM. The reason for this improvement is that WCMM and WMF-NP remove port matching to reduce configuration complexity.

6.5.2 Nonuniform Traffic

The WMoM, WMF-NP, and WCMM schemes with multiple iterations are simulated under two different nonuniform traffic patterns: unbalanced [32] and diagonal [46].

Figure 6.7 shows the throughput performance of WMoM, WMF-NP, and WCMM under unbalanced traffic. This figure shows that the throughput of WMF-NP and of WCMM is above 99% with $q = 16$ and $B = 2$ as that of WMoM under the complete range of w. The high throughput of WMF-NP and WCMM under this traffic model is the product of

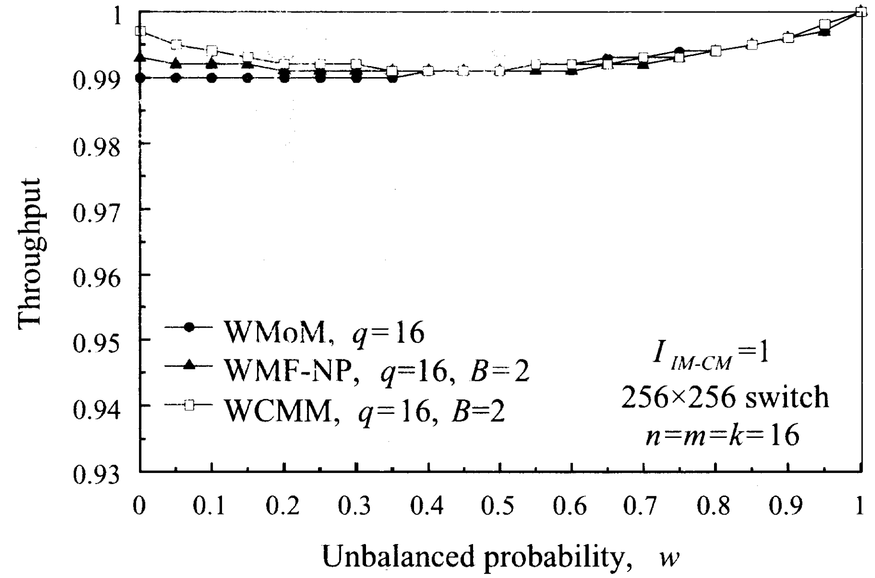

Figure 6.7 Throughput of WMoM, WMF-NP, and WCMM schemes n=m=k=16 switch under unbalanced traffic.

considering the VOQ and CXB occupancy and of providing multiple paths between IMs and OMs.

The last nonuniform traffic pattern considered here is the diagonal traffic model. Figure 6.8 shows that WCMM, using a single IM-CM iteration and $B = 2$, and WMF-NP, using $I_{IM-CM} = 2$ and $B = 2$, deliver higher throughput than WMoM, using $I_{IM-CM} = 16$, under the complete range of x. One of the reasons for this improvement is that WMF-NP and WCMM schemes for SSM Clos-network switches use the input and output port arbiters separately instead of ports matching with multiple iterations, that is, the information doesn't travel from CM to IM for ports matching.

6.6 Conclusions

In this chapter, a novel SSM Clos-network switch is proposed. The SSM Clos-network switch uses crosspoint buffers in output modules and the memory used in the crosspoint buffers needs no speedup. Two configuration schemes for SSM Clos-network switches, called the weighted module first matching and weighted central modules' link matching,

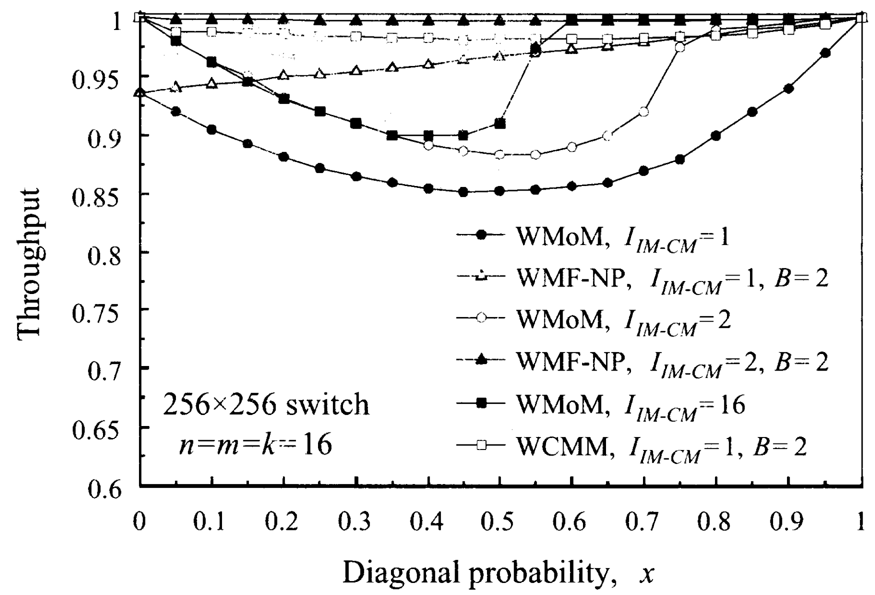

Figure 6.8 Throughput of WMoM, WMF-NP and WCMM *n=m=k=16* under diagonal traffic.

WMF-NP and WCMM, respectively, are also proposed. These schemes reduce the configuration complexity by removing port matching in both schemes and by WCMM reducing the number of iterations in the exchange of information between IMs and CMs in WCMM. Both schemes require small and feasible size schedulers. The reduction of the number of IM-CM iterations is of major importance in three-stages Clos-network switches as the input modules are located in different physical locations from the central modules in a large switch to make their implementation feasible.

The switching performance of three schemes is studied: WMoM of a 256 × 256 bufferless Clos-network switch, and WMF-NP and WCMM of a 256 × 256 SSM Clos-network switch. The results showed that WMF-NP and WCMM provide 100% throughput, as WMoM does, under Bernoulli uniform traffic and above 99% throughput under unbalanced traffic and no memory speedup in the crosspoint buffers at the third-stage modules. WMF-NP and WCMM use a smaller number of IM-CM iterations than WMoM to deliver higher performance under diagonal traffic.

CHAPTER 7

CONCLUSION

High-speed and large-capacity switches are in demand because of the speedy growth of the Internet. In this dissertation, a new efficient, implementable, and low complexity schemes for single-stage IQ switches were proposed. These schemes, called uFORM and uFPIM, are examples of weightless-based matching schemes, that offer a solution for low-complexity and fast matching for IQ packet switches, based on a nonblocking crossbar fabric. The matching schemes use a proposed capture-frame concept and are based on round-robin and random selections, respectively. These schemes provide high throughput under several admissible traffic patterns, including uniform and those with nonuniform distributions, without recurring to speedup nor multiple iterations. The throughput improvement achieved by uFPIM is demonstrated. Furthermore, the proposed captured-frame concept is shown to be scalable as the throughput performance increases as the switch size increases.

uFORM and uFPIM can provide high performance for large IQ switches. However, smaller switches are less sensitive to the unlimited frame-size occupancy values. As an application to different switch sizes, a variation of uFORM is developed, this is called FORM. This scheme, which captures the limited frame-size occupancy values, also provides high throughput under uniform and unbalanced traffic patterns for small switch sizes with different limited frame-size values. The uFPIM, uFORM, and FORM schemes do not need to compare the status among those contention VOQs. The hardware and timing complexity of these schemes is low. This makes them efficient and implementable schemes.

A Clos-network architecture is efficient considered, as switch scalability is required in high-capacity switches. A solution for dispatching schemes on the MSM Clos-network switches is presented. In this dissertation, the study of the captured-frame concept is extended into dispatching schemes for MSM Clos-network switches. The framed concurrent round-robin dispatching (FCRRD) scheme and the framed random dispatching (FRD) scheme, both using a single iteration in the matching between first- and second-stage modules, can achieve higher throughput than those schemes without the captured-frame size under several nonuniform traffic patterns without placing buffers in the second stage modules and without expanding the internal bandwidth. The result also showed that FCRRD use fewer IM iterations than CRRD to keep 100% throughput under uniform traffic. The reduction of the number of iterations is important in MSM Clos-network switches as the input modules are located in different physical locations from the central modules in a large switch and schedulers may be implemented in similar distributed fashion.

To avoid the requirement of memory speedup that MSM switches may have, a novel configuration scheme for input-queued Clos-network (IQC) switches is proposed: the module-first matching (MoM) scheme. In a practical scenario, this scheme performs routing first and port matching thereafter. This approach reduces the scheduler size and the configuration complexity of IQC switches. For example, a 1024×1024 match by MoM requires parallel and independent 32×32 schedulers while other schemes require 1024×1024 schedulers. The high switching performance of the proposed approach using weight-based and weightless selection schemes under uniform and nonuniform traffic is presented. A weight-based MoM scheme, called WMoM, based on the selection of the longest VOQ occupancy first is used to provide high throughput under uniform and nonuniform traffic patterns. A two-stage Clos-network switch and module-first matching (MoM) scheme are proposed. It is shown that a very large scale switch is built and the configuration complexity of this scale switch is reduced by employing the two-stage Clos-network switch and MoM scheme.

The exchange of information between arbiters in different modules produces long resolution delay. A solution for the resolution delay of three-stage Clos-network switches is proposed: the Space-Space-Memory (SSM) Clos-network switch, the weighted module-first and none-port matching (WMF-NP) scheme, and the weighted central modules' link matching (WCMM) scheme. This SSM Clos-network switch that uses crosspoint buffers in the third-stage modules needs no memory speedup. The two configuration schemes provide high throughput under any admissible traffic patterns and reduce the configuration complexity for the SSM Clos-network switches.In addition, WCMM reduces the number of iterations needed in the exchange of information between IM and CM.

REFERENCES

[1] P. Newman, T. Lyon, and G. Minshall, "Flow labelled IP: A connectionless approach to ATM," *Proc. IEEE INFOCOM '96*, vol. 3, March 1996, pp. 1251–1260.

[2] G. Parulkar, D. Schmidt, and J. Turner, "AITPM: A strategy for integrating IP with ATM," *Proc. SIGCOMM. Computer Commun.*, vol. 25, December 1988, pp. 49–59.

[3] "White paper: ATM switching architecture." FORE systems, November 1993.

[4] H. Ahmadi and W. E. Denzel, "A survey of modern high-performance switching techniques," *IEEE J. Select. Area Commun.*, vol. 7, no. 7, September 1989, pp. 1091–1103.

[5] W. Fischer, O. Fundneider, E. H. Goeldner, and K. A. Lutz, "A scalable ATM switching system architecture," *IEEE J. Select. Area Commun.*, vol. 9, no. 8, October 1991, pp. 1299–1307.

[6] H. J. Chao, C. H. Lam, and E. Oki, Eds., *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers*, 1st ed. John Wiley and Sons, Inc., 2006.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proc. IEEE INFOCOM '00*, vol. 2, March 2000.

[8] M. Karol and M. Hluchyj, "Queuing in high-performance packet-switching," *IEEE J. Select. Area Commun.*, vol. 6, December 1988, pp. 1587–1597.

[9] C. Clos, "A study of nonblocking switching networks," *Bell Syst. Tech. J.*, March 1953, pp. 406–424.

[10] J. Turner and N. Yamanaka, "Architectural choices in large scale ATM switches," *IEICE Trans. Commun.*, vol. E81-B, no. 2.

[11] N. Chrysos and M. Katevenis, "Scheduling in non-blocking buffered three-stage switching fabric," *Proc. IEEE INFOCOM '06*, April 2006.

[12] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar, "Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset," *IEEE Commun. Mag.*, December 1997, pp. 44–53.

[13] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, August 1999, pp. 1260–1267.

[14] T. X. Brown and K. H. Liu, "Naural network design of a Banyan network controller," *IEEE J. Select. Area Commun.*, vol. 8, 1990, pp. 1289–1298.

[15] T. E. Anderson, S. S. Owicki, J. B. Saxe, , and C. P. Tacker, "High-speed switch scheduling for local area networks," *ACM Trans. on Computer Systems*, vol. 11, no. 4, November 1993, pp. 319–352.

[16] M. Karol, K. Eng, and H. Obara, "Improving the performance of input-queued ATM packet switches," *Proc. IEEE INFOCOM '92*, vol. 1, May 1992, pp. 110–115.

[17] C. Clos, "An $n^{5/2}$ algorithm for maximum matching in bipartite graphs," *Soc. Ind. Appl. Math. J. Comput.*, vol. 2, 1973, pp. 225–231.

[18] D. Shah, "Maximal matching scheduling is good enough," *Proc. IEEE GLOBECOM '03*, vol. 6, December 2003, pp. 3009–3013.

[19] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, no. 2, April 1999, pp. 188–201.

[20] H. J. Chao and J.-S. Park, "Centralized contention resolution schemes for a large-capacity optical ATM switch," *Proc. IEEE ATM Workshop 1998*, May 1998, pp. 11–16.

[21] E. Oki, R. Rojas-Cessa, and H. J. Chao, "PMM: A pipelined maximal-sized matching scheduling approach for input-buffered switches," *Proc. IEEE GLOBECOM '01*, vol. 1, November 2001, pp. 35–39.

[22] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," *Proc. IEEE INFOCOM '98*, vol. 2, no. 8, April 1998, pp. 792–799.

[23] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches," *Proc. IEEE HPSR '01*, May 2001, pp. 276–280.

[24] C. S. Chang, W. J. Chen, and H. Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," *Proc. IEEE INFOCOM '00*, vol. 3, March 2000, pp. 1614–1623.

[25] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, Part I: one-stage buffering," *Computer Commun.*, vol. 25, 2002, pp. 611–622.

[26] A. Bianco, M. Franceschinis, S. Ghisolfi, A. M. Hill, E. Leonardi, F. Neri, and R. Webb, "Frame-based matching algorithms for input-queued switches," *Proc. IEEE HPSR '02*, May 2002, pp. 69–76.

[27] S. Li and N. Ansari, "Input-queuing switching with QoS guarantees," *Proc. IEEE INFOCOM '99*, vol. 3, March 1999, pp. 1152–1159.

[28] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized scheduling algorithms for high-aggregate bandwidth switches," *IEEE J. Select. Area Commun.*, vol. 21, May 2003, pp. 546–559.

[29] D. Shah, P. Giaccone, and B. Prabhakar, "Efficient randomized algorithms for input-queued switch scheduling," *IEEE J. Micro.*, vol. 22, January-February 2002, pp. 10–18.

[30] N. McKeown, "Scheduling algorithms for input-queued cell switches," *Ph.D. Dissertation, Dept. Elect. Eng. Comput. Sci., University of California at Berkeley, Berkeley, CA*, 1999.

[31] R. Rojas-Cessa, E. Oki, and J. H. Chao, "Maximum weight matching dispatching scheme in buffered Clos-network packet switches," *Proc. IEEE ICC '04*, vol. 2, June 2004.

[32] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined input-one-cell-crosspoint buffered switch," *Proc. IEEE HPSR '01*, May 2001, pp. 324–329.

[33] E. Oki, Z. Jing, R. Rojas-Cessa, and H. J. Chao, "Concurrent round-robin-based dispatching schemes for Clos-network switches," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, December 2002.

[34] T. T. Lee and S. Y. Liew, "Parallel routing algorithm in Benes-Clos networks," *Proc. IEEE INFOCOM '96*, March 1996, pp. 279–286.

[35] K. Pun and M. Hamdi, "Distro: A distributed static round-robin schedulinf algorithm for bufferless Clos-network switches," *Proc. IEEE GLOBECOM '02*, vol. 3, 2002, pp. 2298–2302.

[36] P. Gupta and N. McKeown, "Design and implmentation of a fast crossbar scheduler," *IEEE Micro Mag.*, vol. 19, Feburary 1998, pp. 20–28.

[37] K. Y. Eng and A. S. Acampora, "Fundamental conditions governing TDM switching assignments in terrestrial and satellite networks," *IEEE Trans. Commun.*, vol. 40, no. 2, February 1992.

[38] C.-B. Lin and R. Rojas-Cessa, "Module matching schemes for input-queued Clos-network switches," *IEEE Commun. Lett.*, vol. 11, no. 2, February 2007.

[39] H. J. Chao, S. Y. Liew, and Z. Jing, "A dual-level matching algorithm for 3-stage Clos-network packet switches," *Proc. IEEE 11th Symposium on High Performance Interconnects*, August 2003.

[40] Y. Jiang and M. Hamdi, "A fully desynchronized round-robin matching scheduler for a VOQ packet switch architecture," *Proc. IEEE HPSR '01*, May 2001, pp. 407–411.

[41] Y. Li, S. Panwar, and H. J. Chao, "The dual round-robin matching switch with exhaustive service," *Proc. IEEE HPSR '02*, May 2002, pp. 58–63.

[42] S. Li and N. Ansari, "Chapter 1.3: Switching architecture and scheduling algorithms," *ATM Handbook (F. Golshani and F. Groom, Eds.), International Engineering Consortium*, 2000, pp. 37–54.

[43] S. Motoyama, D. W. Petr, and V. S. Frost, "Input-queued switch based on a scheduling algorithm," *Electronics Lett.*, vol. 31, July 1995, pp. 1127–1128.

[44] G. Nong, J. K. Muppala, and M. Hamdi, "Analysis of nonblocking ATM switches with multiple input queues," *IEEE/ACM Trans. Networking*, vol. 7, February 1999, pp. 60–74.

[45] R. Schoene, G. Post, and G. Sander, "Weighted arbitration algorithms with priorities for input-queued switches with 100% throughput," *Broadband Switches Symposium '99*, 1999. [Online]. Available: http://www.schoenen-service.de/assets/papers/Schoenen99bssw.pdf

[46] K. Pun and M. Hamdi, "Static round-robin dispatching schemes for Clos-network switches," *Proc. IEEE HPSR '02*, May 2002.

[47] Y. Jiang and M. Hamdi, "A 2-stage matching scheduler for a VOQ packet switch architecture," *Proc. IEEE ICC '02*, vol. 4, May 2002, pp. 2105–2110.

[48] M. M. Marsan, M. Ajmone, A. Bianco, E. Leonardi, and L. Milia, "RPA: A flexible scheduling algorithm for input buffered switches," *IEEE Trans. Commun.*, vol. 47, December 1999, pp. 1921–1933.

[49] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, Part II: multi-stage buffering," *Computer Commun.*, vol. 25, 2002, pp. 623–634.

[50] T. T. Lee and C. H. Lam, "Path switching- A quasi-static routing scheme for large-scale ATM packet switches," *IEEE J. Select. Area Commun.*, vol. 15, no. 5, 1997.

[51] W. Feng and M. Hamdi, "Scalable central-stage buffered Clos-network packet switches with QoS," *Proc. IEEE HPSR '06*, June 2006.

[52] X. Li, Z. Zhou, and M. Hamdi, "Space-memory-memory architecture for Clos-network packet switches," *Proc. IEEE ICC '05*, vol. 2, May 2005.

[53] R. Rojas-Cessa and C.-B. Lin, "Captured-frame eligibility and round-robin matching for input-queued packet switches," *IEEE Commun. Lett.*, vol. 8, September 2004.

[54] H. J. Chao, Z. Jing, and S. Y. Liew, "Matching algorithms for three-stage bufferless Clos-network switches," *IEEE Commun. Mag.*, vol. 41, October 2003.

[55] Y. Doi and N. Yamanaka, "A high-speed ATM switch with input and crosspoint buffers," *IEICE Trans. Commun.*, vol. E76, no. 3, March 1993, pp. 310–314.

[56] R. Rojas-Cessa, E. Oki, and H. J. Chao, "CIXOB-1: Combined input-crosspoint-output buffered packet switch," *Proc. IEEE GLOBECOM '01*, vol. 4, November 2001, pp. 2654–2660.