ABSTRACT

JOINT BUFFER MANAGEMENT AND SCHEDULING FOR INPUT QUEUED SWITCHES

by
Dequan Liu

Input queued (IQ) switches are highly scalable and they have been the focus of many studies from academia and industry. Many scheduling algorithms have been proposed for IQ switches. However, they do not consider the buffer space requirement inside an IQ switch that may render the scheduling algorithms inefficient in practical applications.

In this dissertation, the Queue Length Proportional (QLP) algorithm is proposed for IQ switches. QLP considers both the buffer management and the scheduling mechanism to obtain the optimal allocation region for both bandwidth and buffer space according to real traffic load. In addition, this dissertation introduces the Queue Proportional Fairness (QPF) criterion, which employs the cell loss ratio as the fairness metric. The research in this dissertation will show that the utilization of network resources will be improved significantly with QPF. Furthermore, to support diverse Quality of Service (QoS) requirements of heterogeneous and bursty traffic, the Weighted Minmax algorithm (WMinmax) is proposed to efficiently and dynamically allocate network resources.

Lastly, to support traffic with multiple priorities and also to handle the decouple problem in practice, this dissertation introduces the multiple dimension scheduling algorithm which aims to find the optimal scheduling region in the multiple Euclidean space.

JOINT BUFFER MANAGEMENT AND SCHEDULING FOR INPUT QUEUED SWITCHES

by
Dequan Liu

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 2003

APPROVAL PAGE

JOINT BUFFER MANAGEMENT AND SCHEDULING FOR INPUT QUEUED SWITCHES

Dequan Liu

Dr. Edwin Hou, Dissertation Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Nirwan Ansari, Dissertation Co-Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Symeon Papavassiliou, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Jiming Liu, Committee Member — Date
VP, CTO, ZTEUSA

BIOGRAPHICAL SKETCH

Author: Dequan Liu

Degree: Doctor of Philosophy

Date: January, 2003

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, USA, 2003

- Master of Science in Electrical Engineering,
  Beijing University of Posts & Telecommunications, Beijing, China, 1992

- Bachelor of Science in Electrical Engineering,
  Beijing University of Posts & Telecommunications, Beijing, China, 1989

Major: Electrical Engineering

Publications:

Dequan Liu, Nirwan Ansari, and Edwin Hou,
"QLP: A Joint Buffer Management and Scheduling Scheme for Input Queued Switches,"
IEEE Workshop on High Performance Switching and Routing, pp. 164-168, Dallas, TX, May, 2001.

Dequan Liu, Nirwan Ansari, and Edwin Hou,
"Fairness Criterion for Allocating Resources in Input Queued Switches,"
IEE Electronics Letters, Volume 37, Number 19, pp. 1205-1206, September, 2001.

Dequan Liu, Nirwan Ansari, and Edwin Hou,
"A novel fairness criterion for input queued switches,"
IEEE Military Communications Conference, Volume 2, pp. 1474-1478, McLean, VA, October, 2001.

Dequan Liu, Nirwan Ansari, and Edwin Hou,
"A Novel Algorithm for Resource Allocation for Heterogeneous Traffic,"
Conference on Information Sciences and Systems, pp. 382-385, Princeton, NJ, March, 2002.

To my beloved family

ACKNOWLEGEMENT

First of all, I would like to thank my academic advisor Professor Edwin Hou for his excellent guidance, invaluable advice, and constant encouragement throughout the years.

I also would like to express my deep thanks to my dissertation co-advisor Professor Nirwan Ansari for helping me delve into various networking areas. His invaluable academic insights in high-speed networks are the key for the completion of this dissertation.

I am deeply indebted to Professor Sirin Tekinay for helping me to acquire the necessary knowledge to be a successful researcher in data networks and wireless networks.

I would also like to thank Professor Symeon Papavassiliou for his professional guidance, and detailed comments on the drafts of this dissertation.

I would like to express my special thanks to Dr. Jiming Liu and Ms. Liwei Gao for their insightful suggestions on my research work and their kind help for my life.

I would also like to thank Dr. Jinghui Li for his invaluable discussion and help. I would like to thank my colleagues in the Advanced Networking Laboratory and NJ Wireless Center, especially, Ximin Zhang, Jiongkuan Hou, Zhicheng Ni, Jingxuan Liu, Yuanqiu Luo, and Hong Zhao.

Finally, I would like to acknowledge my parents who have always been there for their immeasurable devotion and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, the basic background of this dissertation, which includes the architectures of cell-based switches, functions of scheduling and buffer management schemes, and types of traffic to be supported, are introduced. In the last section, the outline of this dissertation is presented.

1.1 Structures of Input and Output Queued Switches

Many switching architectures have been considered for Asynchronous Transfer Mode (ATM) networks [71, 32]. Depending on the position of the buffer, a switch can be classified as input, output, and input-output queued.

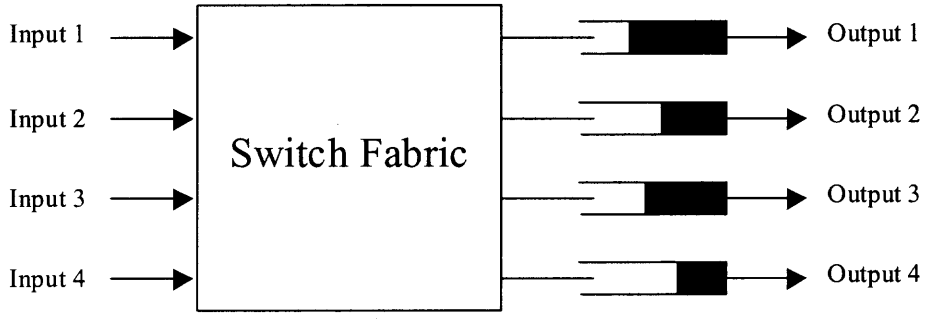

Figure 1.1 The structure of a 4 by 4 output queued switch.

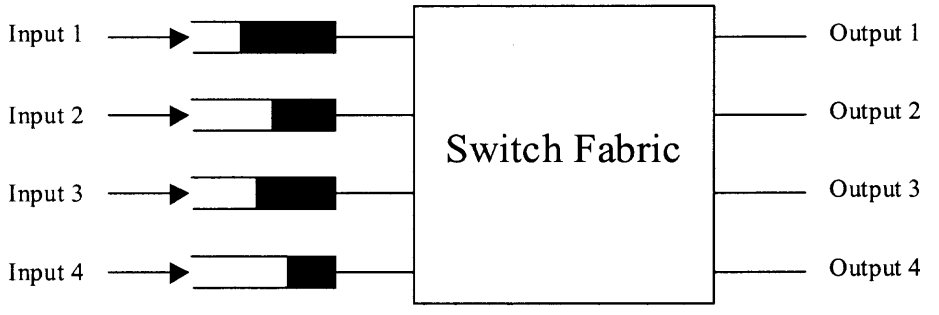

Figure 1.2 The structure of a 4 by 4 input queued switch.

Cells are immediately transmitted across the switching fabric upon arrival in an output queued (OQ) switch, as shown in Figure 1.1, and stored in the buffers at the output side. Scheduling algorithms for an OQ switch then decide which cell can be transmitted to the output lines. Although OQ switches can provide QoS guarantees, they are limited by the speedup requirement — the processing speed of data line inside the fabric and the rate to access the buffer can be, in the worst case, N times the output line rate for an N by N switch. In high-speed networks, with high-speed optical fiber being the transmission media, satisfying this requirement is becoming much more difficult.

In an input queued (IQ) switch as shown in Figure 1.2, there are buffers at the input side. Each input port can keep its own buffer or share a common one with other input ports. Cells are first stored in the buffers before they are selected by the scheduling algorithm to switch through the switching fabric.

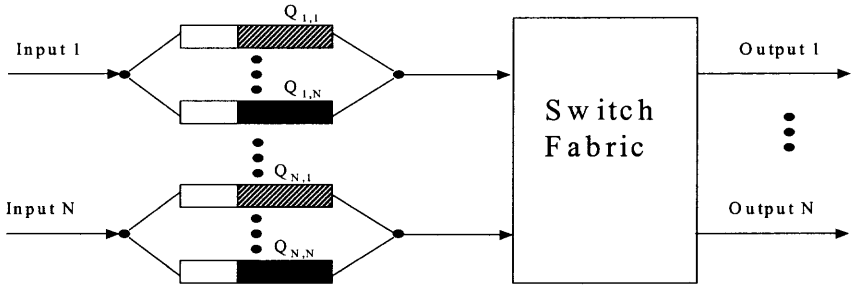

Figure 1.3 The structure of an IQ switch with virtual output queueing.

There is a well-known Head-of-Line (HOL) blocking problem [29] for IQ switches when each input port keeps a single queue for all output ports. HOL occurs when a cell is blocked by another cell, which queues ahead the blocked cell and destines to a different output. HOL blocking limits the throughput of an IQ switch to 58% under uniform independent and identical distribution (*i,i,d*) Bernoulli traffic for large N [29]. The problem is worse for bursty traffic and the throughput of a switch may be limited to

50% when the burstiness of traffic is high [1]. The HOL problem can be eliminated by using Virtual Output Queuing (VOQ), as shown in Figure 1.3, where each input maintains a separate virtual queue for each output [37, 54, 49, 48, 69]. In this dissertation, the IQ switch is assumed to have a separate buffer at each input port, and VOQ is adopted.

1.2 Buffer Management

Buffer management schemes address the problem of buffer space allocation among contenders. An ideal buffer management scheme should possess the following features. First, it should completely allocate the whole buffer space to reduce the overall cell losses. Namely, a cell is always allowed to queue in the buffer when there is empty space inside the buffer. Second, it should be able to regulate the sharing of the buffer to guarantee different QoS requirements imposed by different classes of traffic. Since each class may have its own Cell Loss Ratio (CLR) requirement, it may be necessary to maintain separate queue information for different classes. Third, it should guarantee fairness to isolate well-behaved flows from misbehaving flows. Fourth, it should be simple to implement in high-speed networks. Unfortunately, hardly any of the existing buffer management schemes have all these features. In Chapter 2, it is shown that several schemes meet some, but not all the features of the ideal buffer management scheme.

From the buffer perspective, an ATM switch can be classified as either completely shared memory or non-shared memory [71]. The latter can be further divided into the IQ and the OQ switch based on the location of the memory. For shared memory and OQ switches, the buffer access speed may be the bottleneck of the switch because, as

explained in Section.1.1, it may be required to run, in the worst case, N times the line rate. When shared memory and OQ switches are employed in the high speed network, they may not be able to support high speed links due to current memory techniques. Existing buffer management schemes are mostly proposed for the completely shared memory or OQ switches.

In this dissertation, for an IQ switch, the virtual buffer management scheme that employs the concept of the exchangeability of buffer space and bandwidth is presented. For example, for regulated traffic, network administrators can allocate either its Peak Cell Rate (PCR) with no buffer space or its Sustainable Cell Rate (SCR) with some buffer space to the traffic. From the perspective of Cell Loss Ratio (CLR), both allocation methods can achieve the same results. Thus, it is possible to allocate resources flexibly to find a good tradeoff between bandwidth and buffer space allocation to competing connections based on the current network condition.

1.3 Guaranteed Traffic Versus Best Effort Traffic

The number of Internet users has been increasing exponentially in the last several years. The original Internet Protocol (IP) was designed to support the best-effort traffic only: all traffics in the Internet are treated as the same class. IP Routers are work-conserving to transmit all packets. If the traffic load is more than their transmission capacity, packets will be dropped. RSVP and DiffServ have been proposed to support multi-class traffic and guarantee transmission rates for several classes of traffic in the Internet in the near future [75, 5]. ATM, on the other side, is a mature technology to support multi-class services and different QoS requirements. Many proposals have been suggested to support

IP over ATM network [58, 60, 61, 2]. It is clear that future network must support at least two kinds of traffic: the QoS aware traffic and the best-effort traffic. In this dissertation, two kinds of traffic (multi-class traffic can be considered similarly), which are guaranteed traffic, such as audio and video, and best-effort traffic, such as data, are considered. Connection Admission Control (CAC) is required for the guaranteed traffic before they are allowed to enter into the network, so that their required resources can be satisfied by the network. For the best-effort traffic, no CAC is preceded, and thus they may require more resource than what the network can support. Namely, no QoS can be guaranteed for the best-effort traffic. In this dissertation, these two kinds of traffic are considered together to obtain the maximum throughput of a switch, while satisfying the resource demands of the guaranteed traffic. In addition, the network resources are allocated fairly and efficiently among the best-effort traffic.

1.4 Outline of this Dissertation

Scheduling algorithms for input queued switches will be investigated in this dissertation. Based on the fact that buffers inside switches may not be able to accommodate all arriving traffic, the effect of scheduling algorithms on the buffers are studied. By jointly considering the buffer management schemes and scheduling algorithms, the main goal of this dissertation is to improve the utilization of both buffer space and bandwidth so as to accommodate more traffic with limited resources.

Chapter 2 presents existing scheduling algorithms for input queued switches. The scheduling algorithms are classified into three categories: bipartite graph matching algorithms, parallel iterative matching algorithms, and QoS features guaranteed

algorithms. Chapter 2 also briefly reviews existing buffer management schemes for the shared buffer structure. The packet discarding schemes in the gateway will be discussed in this chapter.

Chapter 3 proposes Queue Length Proportional (QLP) assignment algorithm, which allocates available bandwidth to the competing flows in proportion to their corresponding buffer occupancies. High utilization of both buffer space and bandwidth can be obtained so that both high throughput and low cell loss ratio can be achieved.

In Chapter 4, the cell loss ratio is employed as the fairness metric resulting in the Queue Proportional Fairness (QPF) criterion. Three types of fairness: intra-queue, intra-input and inter-input fairness, are considered for an input queued switch. By employing this new metric, the new criterion can achieve higher utilization of network resources than the traditional Max-Min fairness criterion.

Chapter 5 presents the Weighted Minmax (Wminmax) algorithm to support different QoS requirements of heterogeneous traffic. In Wmimax, heterogeneous and regulated traffics are grouped into several classes according to their negotiated QoS parameters. For each class, network resources are allocated in proportion to their corresponding weights.

In Chapter 6, the Multiple Dimension Scheduling (MDS) algorithm is proposed to support traffic with multiple priorities and to handle the decouple problem in practical implementation. In MDS, the criterion to select the packet to transmit is determined by multiple QoS parameters and, therefore, it is possible to find the optimal scheduling region in the multiple dimensional space.

Chapter 7 provides the conclusions and future work of this dissertation.

CHAPTER 2

EXISTING SCHEDULING ALGORITHMS FOR INPUT QUEUED SWITCHES AND BUFFER MANAGEMENT SCHEMES

In this chapter, existing scheduling algorithms for input queued switches are introduced. In the bipartite graph matching algorithms, several types of matching are considered based on their matching criteria. The parallel iterative matching algorithms aim to find a match by several iterations. Finally, four algorithms, which can guarantee some QoS features, such as delay and delay jitter, are presented. Also, in this chapter, existing buffer management schemes are reviewed. For fixed length cells with a common shared buffer, the existing buffer management schemes can be classified as push-out or non push-out policies based on whether push-out technique is involved or not. They can also be classified as dynamic or static threshold schemes based on whether the buffer partitioning is dynamic or static. In gateways, for variable-length IP packets, several discarding algorithms are also reviewed in this chapter.

2.1 Bipartite Graph Matching Algorithms

A bipartite graph matching [70, 1] G = (U, V, E, W), as shown in Figure 2.1, is used to depict the connection of an IQ switch. Nodes U,V represent input ports and output ports, respectively; edges E stand for possible transmission and edge weights W represent the transmission demands of each edge. A matrix W, as shown in Figure 2.2(a), is used to represent the weights of a switch, where each matrix element represents the corresponding weight from an input to an output. For example, $w_{1,2}$ represents the weight from input 1 to output 2. If all the weights are set as ones, the weight matrix W is

identical to the edge matrix E. Thus, from W, it can also know which inputs have traffic waiting for transmission. For example, $w_{1,4} = 0$ shows the weight from input 1 to output 4 is zero and, at the same time, means that there is no transmission demand from input port 1 to output port 4. Different scheduling algorithms will select different metrics as the weights. For example, Longest Queue First (LQF) algorithm [49] selects the queue length as the weight. Scheduling algorithms will try to find a subnet of edges $M \subset E$ such that each input is connected with no more than one output and vice versa. Let a matrix M represents the matched connections of inputs and outputs found by a scheduling algorithm. If an input and an output are matched, the corresponding element in M will be set to one, and it will be set to zero otherwise. Since an input can at most match one output at each time slot, there is no more than one element being one at each row and each column in M.

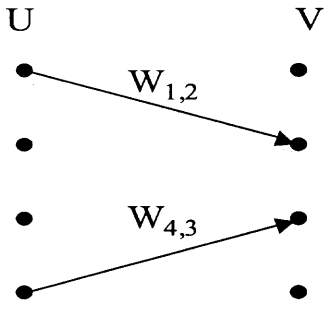

Figure 2.1 A bipartite graph matching.

$$\text{(a)} \begin{bmatrix} 2 & 4 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 1 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.2 The matrix of weights (a), and the matched connection matrix (b).

Four kinds of matching have been considered:

- Maximum Size Matching (MSM) aims at finding a maximum number of matching pairs. The complexity of the algorithm to find the MSM is $O(N^{2.5})$ [24].
- Maximum Weight Matching (MWM) algorithm wants to find a maximum aggregate weight with computation complexity $O(N^3 \log N)$ [70]. i.e., arg $\max_M[\Sigma_{i,j} w_{i,j} m_{i,j}]$. The MSM is a special case of the MWM with all weights being set to ones.
- Maximal Matching (MM) means that no extra matching pair can be added without changing current matches. The best algorithm to find a MM has computational complexity $O(N^2)$ [70]. It is clear that a MSM or a MWM is always a MM. However, a MM may not always be a MSM or a MWM.
- Stable Marriage Matching (SMM): given a weighted bipartite graph (U, V, E, W), a matching $M \subset E$ is a stable marriage matching [17] if for any edge $e \notin M$, there is an edge $e_M \in M$ such that they share a common node and $W(e_M) \geq W(e)$. The computation complexity to find a SMM is $O(N^2)$.

(a) $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$ (d) $\begin{bmatrix} 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$

Figure 2.3 Solutions to the bipartite graph in Figure 2.2 (a): A maximum size match (a), A maximum weight match (b), A maximal match (c), and A stable marriage match (d).

Figure 2.3 shows example solutions of the bipartite graph matching problem based on Figure 2.2(a) with: (a) a maximum size match, (b) a maximum weight match, (c) a maximal match, and (d) a stable marriage match.

Many scheduling algorithms have been proposed based on graph matching. Although the MSM can provide a high throughput for *i.i.d* traffic, it suffers several problems for non-uniform traffic. Under admissible traffic, MSM may lead to the problem of starvation and unfairness. For inadmissible traffic, MSM may lead to starvation, where some flows are not able to be serviced for very long time when the traffic is non-uniform.

Instead of finding a MSM, Longest Queue First (LQF) algorithm [49] aims to find a MWM with the queue length as the weight. [49] proved that LQF is stable under admissible and independent traffic pattern, i.e., the average queue length is finite. For a switch with a big buffer, it implies the LQF can achieve 100% throughput.

The iterative LQF (i-LQF) [50] is a simple version of LQF. By employing the simple iterative algorithm, i-LQF has a low computational complexity. The Longest Normalized Queue First (LNQF) [37] is an improved version of LQF. In LNQF, the queue length is normalized by its rate. Thus, the LNQF can avoid the starvation of the low traffic and achieve fair bandwidth allocation.

The Oldest Cell First (OCF) [52] algorithm was proposed to solve the starvation problem of the LQF algorithm. In OCF, the waiting time of the HOL cell is selected as the weight and a MWM is found based on the selected weights. Since the service metric is the waiting time of a cell, every cell can be serviced eventually after waiting for some finite time and OCF is a stable and starvation-free algorithm for all independent and admissible traffic.

The Longest Port First (LPF) [37] is a practical version of LQF by setting the weight to the aggregated queue length of each port. Also, the LPF matching is

implemented by modifying the Edmonds-Karp maximum size matching algorithm [12, 70] which has lower computational complexity than the MWM algorithm.

The weighted arbitration algorithm [63] can support traffic with different priorities by assigning larger weights to the higher priority traffic.

The shakeup technique [21] is a simple and randomized approach. The motivation of the shakeup technique is that the matching of the next time slot is most likely similar to the matching of the current time slot. In the unweighted shakeup technique, an unmatched port can force a matching for itself even if an existing matching has to be removed from an initial matching. In the weighted shakeup technique, a matching is selected with probability proportional to its queue size among competing ports. The explanation of the performance improvement of the shakeup technique is that when an existing matching is removed, it can cause a chain of shake-outs and lead to an augmenting path in a probability sense.

2.2 Parallel Iterative Matching Algorithms

Parallel Iterative Matching (PIM) algorithm [1] has been proposed by DEC System Research Center for their commercial AN2 switch, which is a 16-port, 16Gb/s switch. PIM aims at finding a MM by several iterations. A connection made in the previous iteration will not be removed in the next matching iteration. Therefore, a MM will be obtained instead of a MSM. Three steps, executing at each input or output, are required in each iteration.

Step 1. Request: Each unmatched input sends a request to every output if there are traffic waiting for the transmission (corresponding $w_{i,j}$ is not zero).

Step 2. Grant: If an unmatched output receives any requests, it randomly selects one to grant.

Step 3. Accept: If an input receives any grants, it randomly selects one to accept.

Repeating the above three steps in each iteration, PIM will be able to find a MM. The average number of iterations that the PIM algorithm will converge is log N. Each iteration may match, on average, at least ¾ of the unmatched pairs until a MM is found. For more than 99% of the times, a MM is able to be found by using PIM within 4 iterations for a 16×16 switch.

A closed form equation for the throughput of a switch using PIM algorithm under *i.i.d.* Bernoulli traffic is given in [19]. It was shown that more than 90% of the throughput can be reached within 3 iterations with PIM for a 16×16 switch.

However, PIM cannot allocate bandwidth flexibly among competing connections. Also, it cannot allocate the bandwidth fairly among competing connections [53]. Another problem of PIM is its complexity caused by implementing a random arbiter in high-speed networks [53].

Round-robin matching (RRM) algorithm [53, 51] is proposed to overcome the problem of complexity and unfairness in PIM by using the round robin arbiter. In RRM, each iteration also has three steps but is modified as follows for a N by N switch:

Step1. Request: Each unmatched input sends a request to every output if the corresponding $w_{i,j}$ is not zero.

Step 2.Grant: If an unmatched output receives any requests, it selects one with the highest priority to grant. The priority is decided by a priority pointer g_i. After each grant

the pointer will increase (ModuloN) to one location next to the granted one following the fixed round robin schedule.

Step 3. Accept: an unmatched input will accept the highest priority one among all grants. Each input keeps a fixed, round robin schedule to give the priority of each output port. The pointer a_i will increase to one location beyond the accepted one in each accept step.

Although RRM algorithm overcomes the complexity and unfairness of PIM, for uniform *i.i.d* Bernoulli traffic, RRM will be unstable for traffic load over 63% [51]. The poor performance of RRM is caused by the way it updates the pointer.

iSLIP algorithm [51] modifies RRM algorithm in the second step as:

Step 2. Grant: If an output receives any requests, it selects the input with the highest priority in a fixed, round robin schedule. The pointer g_j will go to the next location if and only if the grant is accepted in step 3.

This modification prevents arbiters from synchronization as in RRM, where some arbiters keep pointing to the same input.

It is shown in [51] with simulations that iSLIP can achieve 100% throughput for uniform traffic (destinations are uniformly distributed) with only one iteration. Under heavy traffic load, iSLIP behaves similarly as time division multiplexing.

Prioritized, threshold, and weighted iSLIP are also presented in [51]. In Prioritized iSLIP, a separate queue is kept for every priority level traffic. The lower priority level traffic will be served if there is no higher priority traffic waiting for transmission. In threshold iSLIP, a number of threshold levels are maintained to determine the corresponding priority level of a request. After the priority level of a

request is decided, scheduling will follow the prioritized iSLIP. Weighted iSLIP is proposed to avoid the starvation of low priority traffic as in the prioritized iSLIP, where lower priority traffic will not be served until there is no higher priority traffic. An output port distributes its bandwidth among competing input ports in proportion to their corresponding weights.

The Iterative Round Robin with Multiple Classes (IRRM-MC) [56, 57] algorithm is similar to the prioritized iSLIP. Starting from the highest priority, IRRM-MC executes different iterations in different classes. High priority traffic will obtain more resources than low priority traffic by using the IRRM-MC algorithm.

The Simplified PIM (SPIM) [55] is a simple version of PIM, where each input is only allowed to make at most one request in each iteration. Therefore, the accept step is not needed anymore. Thus, the implementation of SPIM is easier than PIM. The performance of SPIM is almost the same as PIM [55].

The weighted PIM (WPIM) [67] can provide bandwidth guarantee in an IQ switch. In WPIM, every VOQ is masked or unmarked depending on whether or not it transmits more traffic than its weighted assignment. A request from a masked VOQs is ignored by output ports. By allowing an input to accept multiple grants belonging to different time slots in each iteration, Enhanced PIM (EPIM) [39] is able to achieve a maximal match in a few iterations.

FCFS In Round Robin Matching (FIRM) [64] algorithm is proposed to improve the delay performance of iSLIP by modifying the second grant step as:

Step 2. Grant: If an unmatched output receives any requests, it selects the input with the highest priority among all requested inputs. If the grant is not accepted, the round robin pointer is kept on this request until it is served eventually.

In the worst case, a request in FIRM can be served in N^2 time slots while a request will wait for $N^2+(N-1)^2$ time slots to be served in iSLIP.

2.3 QoS Features Guaranteed Algorithms

Several scheduling algorithms for IQ switches are proposed to guarantee cell delay for time sensitive traffic, such as voice.

Both the Slepian-Duguid [1] and Store-Sort-and-Forward (SSF) [36] algorithm can be referred to as frame-based schemes, where the time axis is divided into frames. Each frame has a fixed number of time slots. The arriving cells are first stored in the buffers, and the algorithm will rearrange these stored cells in next frame. It is shown in [1], if the total number of stored cells is no more than the number of time slots in each frame, a schedule can always be found. Both algorithms have the problem of rate granularity. Smaller frame can have low delay guarantee but with high rate granularity while larger frame suffers high delay but keeps low rate granularity.

Based on the Birkhoff and Von Neumann theorem for decomposing the rate matrix into permutation matrices, the Birkhoff-Von Neumann (BVN) algorithm [6, 7] was proposed to guarantee cell delay bound if traffic is leaky bucket constrained. The main contribution of BVN is that it decomposes the multiple dimension problem of the scheduling algorithm for an IQ switch into one dimension problem. BVN can achieve 100% throughput for both uniform and non-uniform traffic.

Several linear complexity $O(N^2)$ algorithms are proposed in [26]. These algorithms can be classified into the MWM category, where the weights are selected as the queue length, the waiting time of the oldest cells, and the outstanding credit. The stable marriage match is selected as the matching algorithm for selected weights. Mathematical analysis, based on the Lyapunov functions, shows that these algorithms can support 50% bandwidth reservation. Since the Lyapunov functions give very loose bounds, simulations show that the bandwidth reservation can actually be up to 90%.

2.4 Shared Memory Management Algorithms

Existing buffer management schemes are proposed for completely shared memory or OQ switches. They can be classified as push-out or non-push-out policies. Push-out is a technique to support multi-priorities and fairness by replacing an existing cell with a new incoming one when the buffer is full. Push-out policies are efficient but difficult to implement.

Kamoun and Kleinrock analyzed several buffer sharing schemes in [27] for OQ switches. Complete Sharing (CS) policy allows cells enter into the buffer until it is full. This policy can perform very well under light load, but it can cause severe unfairness under asymmetrical or heavy loading condition because heavy connections can occupy the whole buffer and starve other connections. Complete Partitioning (CP) policy, on the other hand, divides the buffer into separate sections, and each of them can be accessed only by a particular connection. If one connection reaches its threshold, cells from this connection are not allowed to enter into the buffer. This policy guarantees fairness among all connections but may incur high cell loss ratio. Sharing with a Minimum Allocation

(SMA) reserves a minimum number of buffers and the remaining buffers are shared among all output ports. In sharing with a minimum queue lengths (SMXQ) scheme, the number of buffers allocated to each output port is limited to some level. Sharing with a minimum queue lengths and minimum allocation (SMQMA) is a combination of SMXQ and SMA schemes. The traffic pattern in [27] is assumed to be independent Poisson arrivals and exponential service times, and a closed product form solution is obtained in [27].

The existence and the structure of an optimal sharing policy (in the sense of minimum packet loss or maximum throughput) have been investigated in [16]. The policies are called coordinate-convex policies because they have a coordinate-convex state space. In coordinate-convex policies, a packet will never be dropped once it is admitted in the buffer, and thus they belong to the non-push-out policies. For independent Poisson arrivals with exponential service time, it is shown in [16] that the optimal coordinate-convex policy will limit the queue length of output port to some fixed level in an OQ switch with two output ports.

The DoD (Drop-on-Demand) policy is suggested in [73] which allows the drop of accepted packets, and thus it belongs to the class of push-out policies. According to DoD, an arriving packet can always be accepted when there is empty space in the buffer. If a packet destined for an output, which has more packets than any other ports, finds the buffer full, then the arriving packet is dropped. Otherwise the arriving packet will be accepted and a packet that belongs to the longest queue will be pushed-out.

Wu and Mark proposed a strategy called Complete Sharing with Virtual Partition (CSVP) [74] for buffer management at a multiplexer or an output port of an OQ switch.

In CSVP, the total buffer space is partitioned based on the relative traffic loads (measured or estimated). Virtual partition allows a newly arriving cell belonging to an oversubscribed flow to enter into the buffer, and to be overwritten when necessary. The main contribution of the CSVP scheme is that it has the same performance as the CP mechanism but maintains fair allocation to all participating flows.

The Push-out with Threshold (POT) was proved to be the optimal policy in terms of the overall cell loss ratio for a system with two output ports in [11, 65]. Although the overall loss probability of POT policy has no significant improvement comparing to the coordinate-convex policy, the POT policy can keep the loss probability of an individual output constant. Also it can guarantee fairness among competing connections.

Guerin *et al.* presented a special scheme in [22] to provide QoS features through buffer management only.

Cheung and Pencea suggested Pipelined Sections (PS) [8] buffer management method that divides the buffer space into N prioritized sections. Each flow with QoS requirements is assigned some buffer space in each section. Arrivals first enter into the last section and only packets in this section can be transmitted. Remaining sections are used to store and re-organize packets. It is shown that PS method can provide rate guarantee to a leaky bucket constrained flow using significantly less buffer reservation than the technique in [22].

Static Threshold (ST) is simple but does not adapt to traffic conditions, while Push-out technique is efficient but difficult to implement. Choudhury and Hahne proposed Dynamic Threshold (DT) scheme [9] to adaptively change the allocated buffer space to meet the traffic condition. The key idea is that the maximum permissible length

is proportional to the unused buffering in the switch. A queue whose length equals or exceeds the current threshold may accept no more arrivals. However, the DT scheme will leave some amount of buffer space unused [9].

Fan *et al.* suggested a new DT scheme in [14] to improve the utilization of buffer space. A common threshold for all competing flows is dynamically updated based on the current traffic condition. When all traffic loads are low, a high common threshold is expected. On the other hand, if the total traffic loads are high, a low common threshold will be obtained to guarantee the fair allocation of the whole buffer space. The new DT scheme can achieve 100% buffer utilization.

2.5 Random Early Detection Algorithm and its Variants

To avoid congestion at gateways, several dropping policies have been proposed.

The Drop Tail (DT) [23] scheme is a simple one used in most routers today. In DT, when an arriving packet finds the queue full, it is dropped. Several problems exist in DT: (1) Fairness among flows is not considered in DT; (2) It will incur the global synchronization problem where some control windows synchronously decrease their window size leading to low throughput of a gateway.

Random Early Detection (RED) [15] is proposed to provide a high aggregate throughput while keeping the queue size small by dropping packets early before the queue is full. In RED, P_a, a probability function of the average queue size is employed for dropping purpose. When the average queue size is greater than the maximum threshold, all arriving packets are marked to drop. If the average queue size is between the minimum threshold and the maximum threshold, each arriving packet will be marked

with a probability P_a. By using the average queue size as the parameter to mark a packet, RED can be implemented without keeping status information of every connection. However, RED is sensitive to these parameters. Also RED does not guarantee the fairness among connections [47].

Rate and Queue Controlled Random Drop (RQRD) [28], which is based on the structure of the Core-Stateless Fairness Queueing (CSFQ) [68] distributed architecture, uses both the rate and queue information for the dropping. Thus, RQRD can achieve high throughput of a router as well as provide fairness among competing flows. Also, RQRD performs well for both UDP and TCP traffic.

CHAPTER 3

JOINT BUFFER MANAGEMENT AND SCHEDULING ALOGORITHM

In this chapter, a new algorithm is proposed to obtain both high throughput and low cell loss ratio for input-queued switches.

Although output queued switches can provide QoS guarantees, they are limited by the speedup requirement — the processing speed of data line inside the fabric and the rate to access the buffer may be required to run, in the worst case, N times the outside line rate for an N×N switch. In high-speed networks, this requirement is becoming much more difficult to be satisfied. The fabric and the memory of input queued switches, on the other hand, can run at the same rate as the outside line. The well-known Head-of-Line (HOL) blocking problem can be eliminated simply by using Virtual Output Queuing (VOQ), where each input maintains a separate virtual queue for each output.

A key issue related to input-queued switches is scheduling cells to obtain a high throughput as well as a low cell loss ratio. Mckeown *et al.* [54] presented a mechanism to achieve up to 100% throughput by finding a matching of a bipartite graph during every time slot. This algorithm performs very well when the traffic is admissible. However, the computational complexity is $O(N^{2.5})$ per time slot. Recently, a novel algorithm proposed by Chang *et al.* [6, 7] can guarantee not only a high throughput but also a bounded delay. The matching of a bipartite graph is computed over many time slots (e.g., 1000 time slots) rather than one time slot. Thus the new algorithm is "good on average" with much lower computational complexity per time slot. Other scheduling algorithms for input-queued switches such as the ones in [48, 49, 43] can also achieve a high throughput.

However, these scheduling algorithms do not consider the buffer space requirement inside a switch that may render these algorithms inefficient in practical applications.

Lapiotis and Panwar [33, 33] showed that joint buffer management and service scheduling for output-queued switches can improve the utilization of switch resources and accommodate more traffic in the network. This dissertation proposes to adopt this joint optimization concept for input queued switches, which are scalable, resulting in the Queue Length Proportional (QLP) assignment algorithm. In addition, this dissertation focuses on the condition, in which overloaded traffic will last for a long enough time. It is shown that appropriate joint assignment of both buffer space and bandwidth according to the real traffic load will lead to not only a high throughput, but also a low cell loss ratio. An intuitive explanation for this provision is that there is no benefit by assigning more bandwidth to a connection than its assigned buffer space can accommodate. Also, it is not necessary to assign more buffer space to a connection with low allocated bandwidth (rate), especially under heavy traffic condition.

3.1 Birkhoff-Von Neumann Algorithm (BVN)

To show how QLP (which concentrates on how the leftover bandwidth can be efficiently allocated to the best effort traffic) can work together with BVN (which provides enough bandwidth to the guaranteed traffic), the BVN algorithm is first reviewed in this section.

Let $\lambda = (\lambda_{i,j})_{N \times N}$ be the rate matrix of a switch with N input ports and N output ports. Here $\lambda_{i,j}$ denotes the rate demand from input i to output j. It is said to be *non-overbooking* if the following two inequalities are satisfied:

$$\sum_{i=1}^{N} \lambda_{i,j} \leq 1 \quad j = 1,..., N \tag{3.1}$$

$$\sum_{j=1}^{N} \lambda_{i,j} \leq 1 \quad i = 1,..., N \tag{3.2}$$

There exists a set of positive coefficients C_K and associated permutation matrices M_k, $k=1,...,K$ (K is the decomposition number which is less than N^2-2N+2) that satisfy:

$$\lambda \leq \sum_{k=1}^{K} C_k M_k, \quad \text{and} \quad \sum_{k=1}^{K} C_k = 1.$$

After obtaining the coefficients and the permutation matrices, it can set the connection of a switch according to the permutation matrices with the connection duration proportional to the coefficients. BVN can be implemented by the following three algorithms:

3.1.1 The Converting Algorithm

The rate matrix λ is called doubly substochastic if it satisfies conditions (3.1) and (3.2). If both (3.1) and (3.2) are equalities, then the matrix is called doubly stochastic.

Algorithm 1 derives a doubly stochastic matrix R from the doubly substochastic matrix λ by the following steps:

Step 1: Randomly find an element at (i,j) position in λ satisfying $\Sigma_j \lambda_{i,j} < 1$ and $\Sigma_i \lambda_{i,j} < 1$.

Step 2: Let $\varepsilon = 1\text{-}max[\Sigma_j \lambda_{i,j}, \Sigma_i \lambda_{i,j}]$. Then add ε to the element at (i,j) in λ.

Step 3: Repeat step 1 and 2 until the sum of all elements in λ is equal to N.

3.1.2 The Decomposition Algorithm

Let R be the doubly stochastic matrix derived from the original doubly substochastic λ by algorithm 1 and $(i_1, \ldots, i_N)$ be a permutation of $(1, \ldots, N)$ satisfying:

$$\prod_{k=1}^{N} R_{k,i_k} > 0 \tag{3.3}$$

Step1: Let M be the permutation matrix corresponding to $(i_1, \ldots, i_N)$ and $C = min(R_{k,i_k})$ for $1 \le k \le N$. Construct a new matrix R_s by: $R_s = R - C M$

Step 2: If $C < 1$, the matrix $R_s/(1- C)$ is doubly stochastic, and a new permutation of $(i_1, \ldots, i_N)$ satisfying (3.3) can be obtained. Continue step 1 to find a new matrix. If $C = 1$, the representation is completed.

3.1.3 The Scheduling Algorithm

Assign a class of tokens for each permutation matrix M_k, $k=1, \ldots, K$.

Step1: First, a token is generated for each class. The virtual finishing time of the first class k token is: $V_k^1 = \frac{1}{C_k}$, $k=1, \ldots, K$.

Step 2: The switch serves the current class of tokens with the smallest virtual finishing time first.

Step 3: Once the K tokens are served, the next class of K token is generated by:

$$V_k^i = V_k^{i-1} + \frac{1}{C_k}, \quad k=1, \ldots, K, \quad i \ge 2.$$

Repeat step 2 and step 3 until all permutation connections have been served with connection duration proportional to their coefficients.

For example, consider the following rate matrix:

$$\lambda = \begin{bmatrix} 0 & 0.3 & 0.2 & 0.4 \\ 0.2 & 0.3 & 0 & 0.2 \\ 0.4 & 0.1 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0.3 \end{bmatrix}, \tag{3.4}$$

where each row represents an input port and each column represents an output port.

Algorithm 1 may obtain the following doubly stochastic matrix:

$$R = \begin{bmatrix} 0 & 0.4 & 0.2 & 0.4 \\ 0.4 & 0.4 & 0 & 0.2 \\ 0.4 & 0.2 & 0.4 & 0 \\ 0.2 & 0 & 0.4 & 0.4 \end{bmatrix}. \tag{3.5}$$

Algorithm 2 may result in the following decomposition:

$$R = 0.4\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + 0.4\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + 0.2\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Then, the connection of the switch can be set according to the permutation matrices obtained above with connection duration proportional to the corresponding coefficients.

3.2 Queue Length Proportional Algorithm (QLP)

The integrated algorithm QLP is introduced in this section along with its mathematical analysis.

3.2.1 Problem Statement

Let λ be the rate matrix of the guaranteed traffic, and the derived doubly stochastic matrix R be the assigned rate matrix by the scheduling algorithm for both the guaranteed traffic and the best-effort traffic. $R \geq \lambda$ implies that the rate demand of the guaranteed

traffic is satisfied, and R-λ is the bandwidth assigned to the best-effort traffic. The actual traffic rate matrix B, which can be estimated on line, is the aggregated rate of the real guaranteed traffic plus the best-effort traffic.

If the rate matrix B satisfies the non-overbooking conditions (3.1) and (3.2), the scheduling scheme can simply follow the three algorithms introduced in the previous section by deriving the assigned rate matrix R directly from B, and no further steps are needed. Unfortunately, since there is no admission control for the best-effort traffic, B may fail to satisfy conditions (3.1) and (3.2).

Since algorithm 1 derives the assigned rate matrix R from the rate matrix of guaranteed traffic λ only, it may not be able to allocate the leftover bandwidth fairly and efficiently while maintaining a high throughput. For example, from Equation (3.4), the rate demand of the guaranteed traffic from input port 2 to output port 1 is $r_{2,1} = 0.2$, but the assigned rate is 0.4 as shown in Equation (3.5). On the other hand, the rate demand of the guaranteed traffic from input port 3 to output port 1 is $r_{3,1} = 0.4$, and the assigned rate is 0.4. In other words, a rate of 0.2 is assigned to the best-effort traffic from input 2 to output 1, and none from input 3 to output 1. It is possible that the best-effort traffic from input 2 to output 1 may not need all of the 0.2 bandwidth, and thus the assigned bandwidth is wasted. On the other hand, the best-effort traffic from input 3 to output 1 cannot be transmitted because no bandwidth is assigned for it.

Chang *et al.* suggested a solution, referred here as the Max-Min algorithm, for this problem in [7] by applying the Max-Min fairness criterion to allocate the bandwidth. The Max-Min fairness was originally proposed for flow control [3], and it is a rate based, light traffic prioritized criterion [26, 30, 46]. The basic idea is to try to allocate as much

network resource as possible to the connection that has the minimum requirement among all connections. The Max-Min fairness can be reached by the filling procedure where rate allocation for all input-output pairs increases linearly until the minimum one reaches its rate limitation. Other pairs continue to increase their rates similarly until all bandwidths are allocated. The rate limitation of each pair, which is an element of the actual rate matrix B, can be estimated on line [7]. Since the Max-Min algorithm derives the assigned rate matrix R from not only the rate matrix of the guaranteed traffic λ but also the estimated rate matrix B, this algorithm can obtain a high throughput by avoiding possible mismatch between the assigned bandwidth and the real traffic load. However, under overloaded conditions, this method can incur a high cell loss ratio if the buffer space is not large enough. For example, consider a 2×2 switch, the available bandwidth of output 1 for the best-effort traffic is 0.5, and actual rates of the best-effort traffic on the two inputs which is destined for output 1 are $r_{1,1} = 0.5$ and $r_{2,1} = 0.2$, respectively. To achieve Max-Min fairness, the assigned rate for these two inputs should be: $r_{1,1} = 0.3$ and $r_{2,1} = 0.2$. The traffic that cannot be transmitted for the two input ports is $0.2T$ and none, respectively. T is the time interval of the bandwidth allocation procedure. If this condition persists for a long time, the buffer of input 1 is likely to overflow, and the buffer of input 2 is surely under utilized. Another possible problem is the on-line measurement errors may influence the performance of the Max-Min algorithm.

3.2.2 QLP for a Single Output

To achieve high throughput as well as to improve the utilization of the buffer space, this dissertation proposes the Queue Length Proportional (QLP) Assignment algorithm to

avoid the possible cell loss caused by the Max-Min algorithm proposed in [7] under heavy congestion condition.

The QLP algorithm assigns an input port with rate proportional to its buffer queue length. The matrix R may be obtained through the following optimization problem:

Maximize:

$$\sum_{i=1}^{N} R_{i,j} \qquad j = 1,\ldots,N$$

Subject to:

$$\sum_{i=1}^{N} R_{i,j} \le 1 \quad \textit{and} \quad \sum_{j=1}^{N} R_{i,j} \le 1 \tag{3.6}$$

For an $N \times N$ switch, let the buffer length of input n $(n=1,\ldots,N)$, which is destined to the same output m, $m \in (1,\ldots,N)$, be $L_1, L_2, \ldots, L_N$, respectively. Let $L_T = L_1 + L_2 + \ldots + L_N$ be the total virtual queue length for output m.

The available bandwidth of output m for the best-effort traffic is R_T. Let $R_1, R_2, \ldots, R_N$ be the assigned bandwidth by output m for input 1, 2, …, N, respectively. $R_T = R_1 + R_2 + \ldots + R_N$.

Definition 3. 1. *If the allocated rates satisfy the following equation:*

$$\frac{R_i}{L_i} = \frac{R_T}{L_T} = \frac{1}{\mu}, \quad i=1, 2, \ldots, N \tag{3.7}$$

(if $L_i = 0$, then $R_i = 0$), then the allocated rates are called the QLP rates. Otherwise, they are called non-QLP rates.

Definition 3. 2. *Equation (3.7) is called the proportional rule, and μ is called the time factor.*

First, the QLP algorithm for a single output port is discussed. Consider the above example. Let the current queue lengths for the two inputs be 500 and 200 cells, respectively. Following the proportional rule, the assigned rates are: $r_{1,1} = 0.36$ and $r_{2,1} = 0.14$. The traffic that cannot be transmitted after 1000 cell slots for input 1 and 2 are 140 and 60 cells, respectively. If the Max-Min algorithm is used, the traffic that cannot be transmitted for input 1 and 2 are 200 cells and none, respectively. Thus, by using the QLP algorithm, the traffic that cannot be transmitted is balanced between input 1 and 2 to avoid overflow of the buffer at input 1.

It is very interesting to note that although QLP does not specify any explicit rules for buffer management, based on the above example, QLP inclusively completes the function of buffer sharing. By using QLP, it seems that the input port with the heavy traffic load can *steal* the buffer space from the ones with the light traffic load by transmitting more cells from its port and delaying the transmission of cells from other ports. It is also one of the reasons why QLP can have lower cell loss ratio than a non-QLP one given a limited buffer space.

Theorem 3. 1. *The policy for bandwidth assignment for an output port that follows the proportional rule can maximize the throughput of best effort traffic.*

Proof:

Let $\underline{L} = [L_1, L_2, \ldots, L_N]^T$ be the set of virtual buffer lengths of input ports destined for a same output port.

$\underline{R} = [R_1, R_2, \ldots, R_N]$ be the set of the QLP rates.

$\underline{R}(i, j, \varepsilon) = [R_1, \ldots, (R_i + \varepsilon), \ldots, (R_j - \varepsilon), \ldots, R_N]$ be the set of non-QLP rates with a mismatch rate ε happened at input i and j, respectively, where $i,j = 1, \ldots, N$, $0 \leq \varepsilon \leq R_j$.

Let $T \geq 1$ be the current time interval for the bandwidth allocation procedure. If $T = 1$, the bandwidth allocation is performed per time slot. Also, let β be the allowed transmission time of the best-effort traffic, and $T-\beta$ be the transmission time of the guaranteed traffic. During each time interval T, the total best-effort traffic transmitted by using QLP and non-QLP is S_Q and S_{NQ}, respectively:

Case 1: $\beta < \mu$. In this case, not enough bandwidth is available for the best-effort traffic.

$$S_Q = R_T \beta$$

$$S_{NQ} = (R_T - R_i - R_j)\beta + \min(L_i, (R_i + \varepsilon)\beta) + (R_j - \varepsilon)\beta$$

$$\text{If } L_i < (R_i + \varepsilon)\beta \Rightarrow \varepsilon > L_i / \beta - R_i \tag{3.8}$$

$$S_Q > S_{NQ}$$

Otherwise: $S_Q = S_{NQ}$

Case 2: $\beta = \mu$. In this case, there is an exact bandwidth for the best-effort traffic.

$$S_Q = R_T \beta = L_T$$

$$S_{NQ} = (R_T - R_i - R_j)\beta + \min(L_i, (R_i + \varepsilon)\beta) + \min(L_j, (R_j - \varepsilon)\beta)$$

$$= (R_T - R_i - R_j)\beta + R_i\beta + (R_j - \varepsilon)\beta < S_Q$$

Case 3: $\beta > \mu$. In this case, there is more than enough bandwidth for all best-effort traffic.

$$S_Q = R_T \mu = L_T$$

$$S_{NQ} = (L_T - L_i - L_j) + L_i + \min(L_j, (R_j - \varepsilon)\beta)$$

$$\text{If } L_j > (R_j - \varepsilon)\beta \Rightarrow \varepsilon > R_j - L_j / \beta \tag{3.9}$$

$$S_Q > S_{NQ}$$

Otherwise: $S_Q = S_{NQ}$

Thus, a non-QLP algorithm cannot transmit more traffic than a QLP one. ■

Corollary 3. 1. *The further time delay caused by a non-QLP algorithm compared to the QLP one is decided by the mismatch rate ε, the time factor μ, transmission time of the best-effort traffic β, and related queue lengths L_i or L_j.*

Proof:

Let δ be the further time delay caused by a non-QLP algorithm. Consider the same cases as in the above theorem.

Case1:

$$\delta = ((R_i + \varepsilon)\beta - L_i)/(R_i + \varepsilon) = \beta - \mu/(1 + \mu\varepsilon / L_i) \tag{3.10}$$

subject to Equation (3.8).

Case 2:

$$\delta = \varepsilon\beta/(R_j - \varepsilon) = \beta/(L_j/(\mu\varepsilon) - 1) \tag{3.11}$$

Case 3:

$$\delta = (L_j - (R_j - \varepsilon)\beta)/(R_j - \varepsilon) = \mu/(1 - \mu\varepsilon / L_j) - \beta \tag{3.12}$$

subject to Equation (3.9).

From Equations (3.10)-(3.12), it can be concluded that the further time delay is decided by ε, β, μ, L_i or L_j. ■

3.2.3 QLP for a Switch

Although QLP maximizes the throughput of the best effort traffic, unfortunately, the QLP rates may not always be achieved for a switch that has multiple output ports as they are limited by condition (3.6). Thus, the working area of the optimal bandwidth assignment must be obtained in term of the throughput for a switch. Figure 3.1 and Figure 3.2 show

the rate assignment for a 2×2 switch under the condition of $\beta \leq \mu$ and $\beta > \mu$, respectively. The x- axis and y-axis represent the rates assigned to input port i and j, respectively.

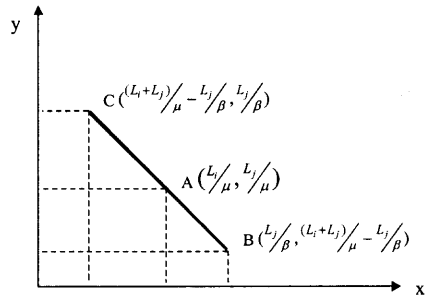

Figure 3.1 Rate assignment that maximizes best-effort traffic throughput for $\beta \leq \mu$.

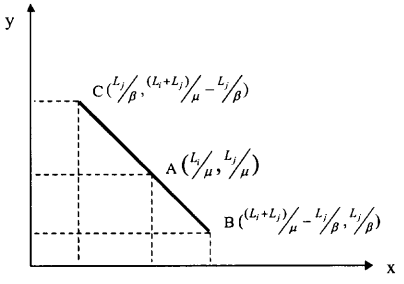

Figure 3.2 Rate assignment that maximizes best-effort traffic throughput for $\beta \geq \mu$.

As shown in Figure 3.1 and Figure 3.2, the rates that maximize the throughput take on values on line AB (or AC). Line AB implies a mismatched rate of $\varepsilon \geq 0$ is added to R_i and subtracted from R_j; Line AC, on the other hand, implies that ε is added to R_j and subtracted from R_i. Point A represents the proportional rates. Although the assigned rates taken on point B (or C) can also obtain a maximum throughput, they cannot approach a high utilization of buffers, e.g., taking rate values on point B, the buffer for input j may be full, and the buffer for input i will be under utilization.

3.2.4 Fairness of QLP

QLP follows the Queue Proportional Fairness (QPF) criterion [44] instead of the Max-Min fairness. QPF criterion, which employs the cell loss ratio as the fairness metric, is proposed to efficiently allocate both buffer space and bandwidth to the best effort traffic.

Although a buffer management scheme such as POT can prevent misbehaving users from hogging the whole buffer space at each input port, it still need to limit overloaded users from occupying too much bandwidth from users in other input ports by setting a maximum length threshold L_M. If $L_i > L_M$, L_i is set to be equal to L_M , $i=1,...,N$, in Equation (3.7).

3.3 Evaluation of the QLP and Max-Min Algorithms

A 4×4 non-blocking crossbar switch is used to evaluate the performance of QLP and Max-Min algorithm. The POT is used as the buffer management policy to manage the buffer at each input port for both algorithms. It is assumed that all the input ports and output ports have the same transmission rates, and it is necessary to normalize the rates of the input ports by dividing them by that of the output ports.

The traffic is generated at each input port as a fully loaded Bernoulli traffic (*i.e.*, p $\approx$ 1). To evaluate both algorithms under severe overloaded condition, the dissertation assumes that all traffic from input 1 goes to output 1; 50% of traffic from input 2 goes to output 1, and another 50% goes to the output 2; 80% of traffic from input 3 goes to output 3, and the other 20% goes to output 4; half of the traffic from input 4 goes to output 3 and another half goes to output 4. Note that there is not enough bandwidth for output 1 and 3.

Figure 3.3 shows the required buffer space for no cell loss in input port 1 using the Max-Min algorithm and the QLP algorithm, respectively. It is shown that, with the same traffic condition, the switch requires less buffer space at each input port by using the QLP algorithm than that by using the Max-Min algorithm.

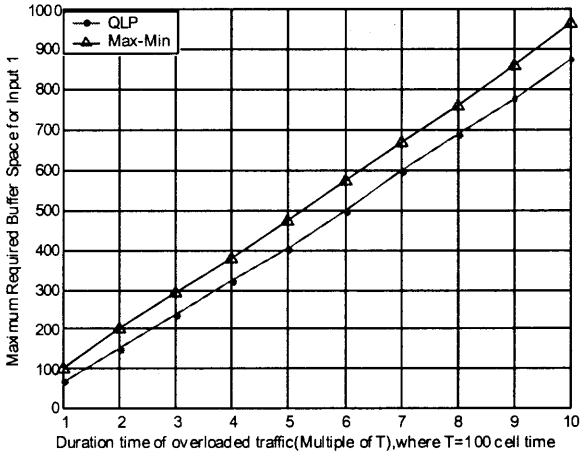

Figure 3.3 Comparison of the maximum required buffer space using the QLP and Max-Min algorithms for input 1.

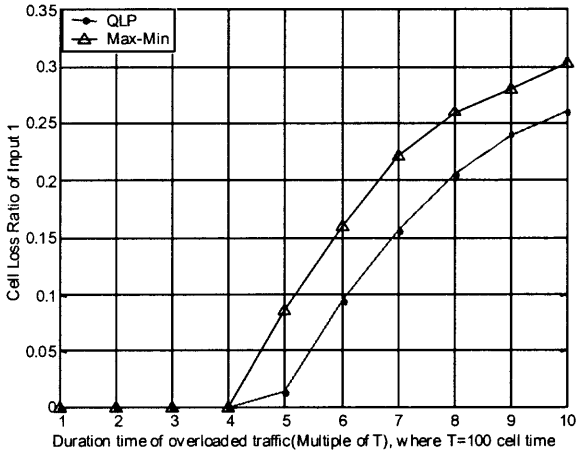

Figure 3.4 Cell loss ratio of input port 1 using the QLP and Max-Min algorithms.

As shown in Figure 3.4, if the buffer space for each input port is limited to 400 cells, the cell loss ratio using the QLP algorithm is around 25% to 100% lower than that using the Max-Min algorithm.

The throughput of QLP as shown in Figure 3.5 has improved by about 6% as compared to that of the Max-Min algorithm.

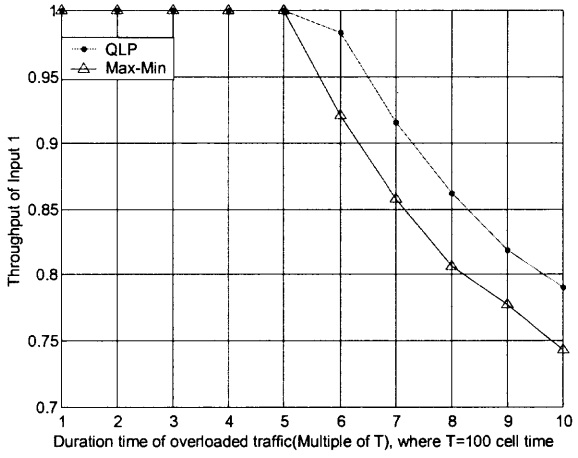

Figure 3.5 Throughput of input port 1 using the QLP and Max-Min algorithm.

3.4 Discussion

In QLP, the allocation of bandwidth is based on the real traffic queue length as well as the available bandwidth so that neither buffer space nor bandwidth will be wasted for possible mismatch between them. Since QLP considers the finite buffer space inside a switch, it can achieve not only a high throughput but also a low cell loss ratio.

Another salient feature for QLP is that the heavy load traffic in an input port can logically share buffers of other input ports although there are no physical connections among them. QLP is most suitable for handling congestion caused by the bursty traffic, hot-spot traffic, and malicious users.

CHAPTER 4

FAIRNESS ISSUES

A novel fairness criterion, Queue Proportional Fairness (QPF), which facilitates a universal fairness criterion for allocating both buffer space and bandwidth, is proposed in this chapter. In QPF, cell loss ratio is employed as the fairness metric instead of the bandwidth (rate) as in the well-known Max-Min fairness for the best effort traffic.

Fairness is a universal concept that is required for flow control, buffer management, and scheduling [72, 66, 7, 74]. Regardless of how differently fairness is defined, there is a common agreement that traffic with the same priority should be treated in the same way. However, many different allocation metrics exist such as throughput, response time, and fraction of demand, indicating that fairness criteria are rather application specific [25].

In the area of flow control of the ABR (Available-Bit-Rate) services, the ATM Forum, for example, has defined a number of fairness criteria, which provides vendors the freedom to implement any one of these definitions in their products to control the transmitting capacity of sources [72]. Jain *et al.* also suggested the fairness index to quantitatively measure fairness [25].

To define a fairness criterion for allocating network resources in an ATM switch, the following factors should be considered. First, there are many different switching architectures [71] for an ATM switch. For example, in an OQ switch, contention may happen at output ports. Therefore, it should apply the fairness criterion to allocate resources at the output ports. Second, fairness is required in allocation of buffer space as

well as bandwidth. Third, the switch has to handle different classes of traffic, and the traffic in each class are further classified into several priorities according to the corresponding QoS requirements.

In an IQ switch, buffer management handles the fair allocation of buffer space among all competing connections before they can be switched inside the switch fabric. If each input port keeps a separate buffer, buffer space is essentially allocated among connections that enter into the same input port. One function of scheduling algorithms is to allocate bandwidth as fair as possible among different input ports and connections contending for the same output port.

Guaranteed traffic, such as voice, is provisioned by the Connection Admission Control (CAC) function at set up. With neither bandwidth nor buffer space guaranteed from the network, the best effort traffic, on the other hand, may not be able to obtain enough resources especially under the overloaded condition. Thus, cell losses or long time delay may be inevitable for the best effort traffic.

In a switch, it should first allocate resources to satisfy the requirements of the guaranteed traffic. The leftover resources are then allocated fairly among the competing best effort traffic. Most previous works [7, 26] allocate bandwidth of an input queued switch to competing connections according to the well-known Max-Min fairness criterion, which is originally proposed for flow control [3]. The basic idea of Max-Min fairness is to allocate as much bandwidth as possible to the connection that has the minimum requirement among all connections. The Max-Min fairness can be reached by the filling procedure—bandwidth allocation for all input-output pairs increases linearly until the minimum one reaches its bandwidth limitation. Other pairs continue to increase

their bandwidths until all bandwidths are allocated. The Max-Min fairness criterion is essentially a rate-based, light traffic prioritized criterion [26, 30, 46]. Although Max-Min fairness criterion can achieve a fair bandwidth allocation, it is at the expense of the buffer space utilization. In other words, it is desired to define a new universal fairness criterion that can achieve a fair and efficient allocation of both bandwidth and buffer space, and thus can handle more traffic in a switch by improving the utilization of network resources. Furthermore, estimating the rate of the best effort traffic requires a complex procedure, and the estimated errors can greatly degrade the performance of the switch.

4.1 Fairness Criteria for ABR Services

The ATM Forum suggested several fairness criteria for the flow control of ABR services. For ABR service, PCR (Peak Cell Rate) and MCR (Minimum Cell Rate) are decided when a connection is setup. PCR specifies the maximum rate at which traffic can be transmitted on the connection, and MCR is the minimum rate guaranteed by the network.

Although these suggested fairness criteria were originally proposed for flow control of ABR services, some of them can serve as valuable references for handling best effort traffic in an IQ switch. Below are several examples of fairness criteria for ABR services recommended by the ATM Forum:

1. Max-Min: $R(i)=R/N$
2. MCR+ equal share: $R(i)= R_M(i)+R/N$
3. Maximum MCR or Max-Min: $R(i)= MAX(R_M(i),R/N)$
4. Proportional to MCR: $R(i)= R(R_M(i)/R_M)$
5. Weighted allocation: $R(i)= R(W(i)/W)$

6. Weighted + MCR share: $R(i)= R_M(i)+(R- R_M)(W(i)/W)$

where

R: bandwidth to be shared by connections;

N: the number of connections;

$R(i)$: allocated bandwidth to connection i;

$R_M(i)$: MCR of connection i;

R_M : sum of MCR of all connections;

$W(i)$: weight for connection i;

W: sum of weights.

Nho *et al.* suggested the proportional fairness [59]:

$$R(i) = R_M(i)+(R-R_M)(P(i)-R_M(i)) / (P- R_M) \tag{4.1}$$

where

$P(i)$: PCR of connection i;

P: sum of PCR of all connections.

By proportional fairness, allocated bandwidths of ABR services are in proportion to their non-predetermined transmitting capacity (PCR-MCR). Both uni-cast and multi-cast connection cases are considered in [59].

4.2 Fairness Index

To compare the fairness quantitatively, Jain *et al.* [25] proposed the fairness index:

$$f(x) = \frac{(\sum_{i=1}^{N} x_i)^2}{N\sum_{i=1}^{N} x_i^2} \tag{4.2}$$

where $x_i \geq 0$ is the normalized service rate of connection i. The fairness index is bounded between 0 and 1. If every connection has the same share, the fairness index is equal to 1. Several fairness metrics are employed depending on the applications [25]:

1. Response time—for interactive traffic;
2. Throughput—for file traffic;
3. Power—for traffic consisting of both file traffic and terminal traffic;
4. Fraction of demand—for systems with different demands of resources.

Kim *et al.* [31] proposed a modified fairness index based on utilization of an ATM switch. The modified fairness index is identical to Equation (4.2) except that $x_i = s_i / m_i$, where s_i and m_i are the average and maximum service rate of entity i, respectively. The value of x_i indicates the utilization of service rate of entity i. The modified fairness index based on utilization can improve the throughput of the switch. Parameters of three types of fairness: inter-queue, inter-input, and inter-output fairness are obtained by using three scheduling algorithms and compared in [31]. Inter-queue fairness denotes the fairness among queues in the same input port. Inter-input and inter-output fairness denote the fairness among input ports and output ports, respectively.

4.3 Queue Proportional Fairness (QPF)

The novel fairness criterion, QPF, is presented here. The contributions of QPF are twofold. First, the fairness metric employed in QPF is cell loss ratio, which is more suitable for the best effort traffic than the bandwidth as employed in the Max-Min fairness criterion. Second, it integrates a universal fairness criterion for allocation of both

buffer and bandwidth so that it can guarantee not only fairness in term of cell loss ratio, but also high utilization of network resources.

4.3.1 Problem Statement

The fabric and buffers of IQ switches can run at the same rate as that of the line rate. Thus, input queued switches are preferable especially in high speed networks. The switch fabric is one with no internal blocking, such as a crossbar.

Consider a rate matrix λ representing rate demands of guaranteed traffic:

$$\lambda = \begin{bmatrix} 0 & 0.1 & 0.2 & 0.4 \\ 0.2 & 0.3 & 0 & 0.2 \\ 0.1 & 0.1 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0.3 \end{bmatrix}, \tag{4.3}$$

where each row represents an input port, and each column represents an output port. $\lambda_{i,j}$ denotes the normalized rate demand from input port i to output port j. For the guaranteed traffic, it is said to be non-overbooking if the following two inequalities are satisfied:

$$\sum_{i=1}^{N} \lambda_{i,j} \le 1 \text{ and } \sum_{j=1}^{N} \lambda_{i,j} \le 1.$$

Let R be the allocated rate matrix, which is the sum of the rates provided to guaranteed traffic and best effort traffic. $R_{i,j} - \lambda_{i,j}$ is the rate allocated for the best effort traffic from input i to output j. The allocated rates should satisfy:

$$\sum_{i=1}^{N} R_{i,j} \le 1 \text{ and } \sum_{j=1}^{N} R_{i,j} \le 1. \tag{4.4}$$

Equation (4.4) indicates that each input and output port cannot transmit more traffic than its transmission capacity.

To achieve the Max-Min fairness, the leftover bandwidth 0.1 in output port 4, for example, has to be shared among input 1, 2, 3 and 4. Namely, $R_{i,4} - \lambda_{i,4} = 0.1/4 = 0.025$, i

= 1, 2, 3, 4. Without considering current buffer occupancy, it is possible that some inputs may need bandwidth more than 0.025 while others may need less than 0.025. In other words, some bandwidth may be wasted due to lack of traffic for transmission in some inputs while traffic in other inputs cannot be transmitted due to lack of bandwidth. To overcome the mismatch of bandwidth and traffic caused by the Max-Min fairness criterion, a new fairness criterion is needed to improve utilization of network resources.

4.3.2 Queue Proportional Fairness

In an IQ switch, three types of fairness, intra-queue, intra-input and inter-input fairness, are defined.

Definition 4. 1. *In an IQ switch, Intra-queue fairness is defined as the fairness among connections that enter into the same input port and destine to the same output port.*

Definition 4. 2. *In an IQ switch, Intra-input fairness is defined as the fairness among aggregated connections that enter into the same input port and destine to a different output port.*

Definition 4. 3. *In an IQ switch, Inter-input fairness is defined as the fairness among different input ports that contend for the same output port.*

For example, in Figure 1.3, intra-input fairness is the fairness among $Q_{1,1}$, $Q_{1,2}$,..., and $Q_{1,N}$ and inter-input fairness is the fairness among $Q_{N,1}$, $Q_{N,2}$,..., and $Q_{N,N}$.

From the perspective of an IQ switch, although the intra-queue fairness can affect the throughput of each individual connection, it has little effect on the total throughput of the switch if any working-conserving scheduling algorithms are employed. However, a proper definition of the intra-input and inter-input fairness can notably improve the total

throughput of a switch as shown in this dissertation. Normally, both buffer management and scheduling are involved in intra-queue and intra-input fairness, and scheduling should consider the inter-input fairness.

It is clear that existing buffer management schemes (e.g. [74]) can only handle the allocation of resources among flows sharing the same buffer, for example, intra-queue and intra-input fairness, but it can hardly affect inter-fairness for each input port keeping its own buffer.

For output queued switches, a fairness definition is given in [20] as:

$$\left| \frac{W_i(\tau)}{R_i} - \frac{W_j(\tau)}{R_j} \right| \leq K$$

where, K is determined by the maximum packet length. $W_i(\tau)$ and $W_j(\tau)$ are services received during time duration τ for session i and session j, respectively. R_i and R_j are the service sharing of session i and session j, respectively. Note that the service sharing is the portion of available bandwidth, which each session is supposed to obtain, rather than the input rate of each session.

For an input queued switch, no more than one cell will be received at each output port during one cell slot. Therefore, the received cell can be transmitted immediately upon arrival, and it is not necessary to consider fairness issue at its output ports.

In summary, in an IQ switch, intra-input and inter-input fairness are the key criteria that can completely affect the performance of the switch. This dissertation will thus focus on them.

Let $Q_{i,j}^k, R_{i,j}^k$ be the queue length and allocated bandwidth of connection k originated from input i and destined to output j, respectively. Here, $k = 1, 2, \ldots, K_i$, and i,

$j = 1, \ldots, N$. Let $Q_{i,j}$ and $R_{i,j}$ be the aggregated queue length and the allocated bandwidth of connections from input i to output j, respectively.

$$Q_{i,j} = \sum_{k=1}^{K_i} Q_{i,j}^k \qquad and \qquad R_{i,j} = \sum_{k=1}^{K_i} R_{i,j}^k$$

And, let Q_j and R_j be the total virtual queue length and the available bandwidth of output j, respectively.

$$Q_j = \sum_{i=1}^{N} Q_{i,j} \qquad and \qquad R_j = \sum_{i=1}^{N} R_{i,j}$$

Let ϕ_j be the set of inputs, which have backlogged traffic destined to output j, and θ_i be the set of outputs at which input i has backlogged traffic to be transmitted.

Definition 4. 4. *For each input-output pair, the Queue Proportional Fairness is satisfied if at least one of the following equalities is satisfied.*

$$\frac{R_{i,j}}{Q_{i,j}} = \frac{R_{h,j}}{Q_{h,j}} \qquad i,h \in \phi_j \quad and \quad j = 1,\ldots,N \tag{4.5}$$

$$\frac{R_{i,j}}{Q_{i,j}} = \frac{R_{i,l}}{Q_{i,l}} \qquad j,l \in \theta_i \quad and \quad i = 1,\ldots,N \tag{4.6}$$

Equation (4.5) and (4.6) give us a criterion for allocating bandwidth to guarantee intra-input and inter-input fairness in an IQ switch.

Note that a non-compliant connection in one input port can still take too much bandwidth from a compliant connection in another input port if Equations (4.5) and (4.6) are satisfied without further constraints. For example, both a non-compliant connection (Con1) from input port 1 and a compliant connection (Con2) from input port 2 are destined to output 1. If input port 1 currently has only one connection, buffer management will not push-out cells from Con1 even its maximum threshold having been

violated until the whole buffer space is taken. Thus, it is still unfair to use the current queue length ($Q^c_{i,j}$) of Con1 to calculate its bandwidth sharing. A virtual maximum threshold of queue length Q_T should be applied to each connection. Q_T can be statically designed according to tariffs or dynamically changed according to the network condition. The reason that Q_T is called the virtual maximum threshold of queue length is that it is only used to calculate the bandwidth sharing rather than being used as the real push-out threshold in the buffer management scheme. The queue length used in Equations (4.5) and (4.6) to calculate the bandwidth sharing should be:

$$Q_{i,j} = \min\{Q^c_{i,j}, Q_T\} \quad i, j = 1,..., N \tag{4.7}$$

Also, QPF can be satisfied by the water filling procedure [3] with the increasing speed of the bandwidth being the queue length in Equation (4.7). In the water filling procedure, rate (bandwidth) allocation for all input-output pairs increases linearly until the minimum one reaches its rate limitation. Other pairs continue to increase their rates similarly until all bandwidths are allocated. Note that QPF is identical to the weighted Max-Min fairness if tariffs are replaced by queue lengths [30, 46].

The following Lemma shows that QPF is a cell loss ratio based criterion. Let $L_{i,j}$ be the cell loss ratio of the aggregated connections from input i to output j, then:

Lemma 4. 1. *For each input-output pair, at least one of following equalities is satisfied:*

$$L_{i,j}=L_{k,j},\ i,k\in\phi_j$$

$$L_{i,j}=L_{i,l},\ j,l\in\theta_i$$

Proof of Lemma 4.1 is a direct result from Definition 4.4.

By employing the QPF criterion, maximum throughput of an IQ switch can be obtained as stated in Lemma 4.2, which results from the intuition that bandwidth is not wasted if QPF is satisfied.

Lemma 4. 2. *The maximum throughput of an IQ switch can be obtained if QPF is satisfied.*

Proof of Lemma 4.2 follows the proof of Theorem 3.1 in Chapter 3 of this dissertation.

It is worthy to note that although QPF can obtain a maximum throughput of an input queued switch, it still cannot guarantee a minimum cell loss ratio under the fact that buffer space inside a switch is, sometimes, finite comparing to real traffic load.

4.4 Evaluation of QPF

To compare the performance of the QPF and Max-Min fairness criterion, bandwidth of each output port to competing input ports is allocated according to the QPF and Max-Min fairness criterion separately in a 2×2 IQ switch.

The Bernoulli source with probability $p \approx 1$ is generated at each input port. To generate the severe overloaded condition, this dissertation assumes, without loss generality, that 80% of traffic from input 1 goes to output port 1, and another 20% goes to output 2; half of the traffic from input port 2 goes to output port 1, and another half goes to output 2.

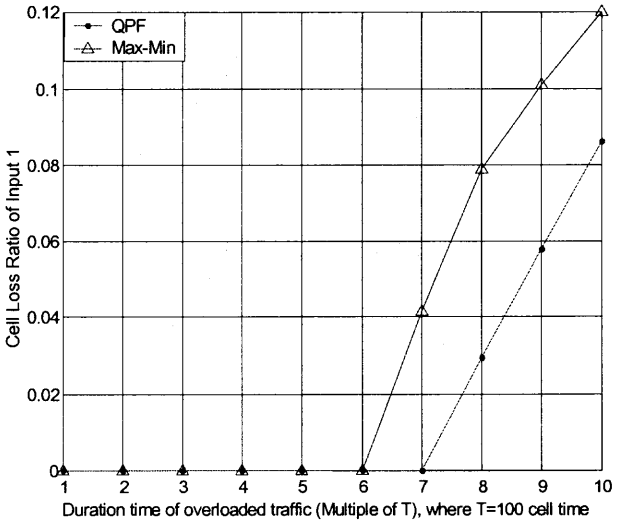

Figure 4.1 Cell loss ratio of input port 1 using the QPF and Max-Min fairness criterion.

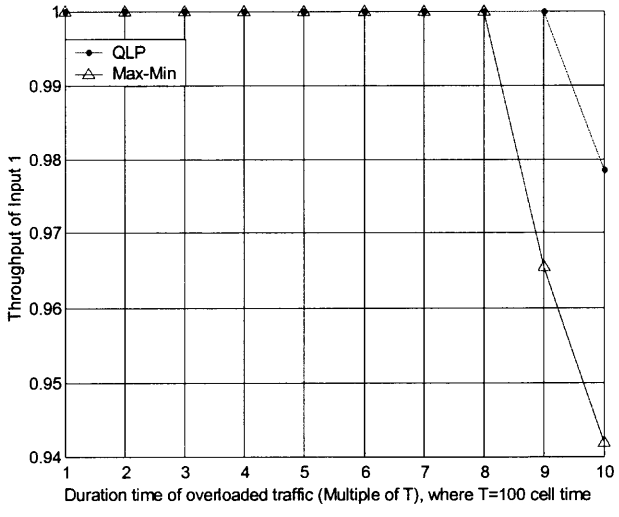

Figure 4.2 Throughput of input port 1 using the QPF and Max-Min fairness criterion.

Figure 4.1 shows that if the buffer space is limited to 400 cells, the cell loss ratio using the QPF criterion is improved by about 25% to 100% as compared to that using the Max-Min fairness criterion. The throughput using the QPF as shown in Figure 4.2 has improved by about 4%-6% as compared to that using the Max-Min criterion. The reason for the improvement is that the Max-Min fairness is a bandwidth based, light traffic

prioritized criterion while the QPF criterion inclusively considers the fair allocation of both bandwidth and buffer space.

The simulations assume that the rates of the best effort traffic are known exactly when the Max-Min fairness is applied. In practice, additional improvement is expected with QPF criterion because there are no inherent rate estimation.

4.4 Discussion

A novel fairness criterion, QPF, has been proposed in this dissertation to achieve high utilization of network resources as well as to guarantee fairness in terms of cell loss ratio. The main contribution of the QPF criterion is that it inclusively considers the allocation of both buffer space and bandwidth so that networks can accommodate more traffic than those by applying buffer management and scheduling algorithms separately.

The QPF criterion is the first attempt employing cell loss ratio as the fairness metric, which may be most suitable for the best effort traffic. Unlike the Max-Min fairness criterion, QPF is designed to handle the congestion caused by the busty traffic, the hot-spot traffic as well as the overloaded traffic. Although QPF criterion is designed for allocating resources in an input queued switch for the best effort traffic, it can also be used for flow control of the ABR services.

CHAPTER 5

RESOURCE ALLOCATION FOR HETEROGENEOUS TRAFFIC

In real networks, various classes of traffic exist and they may require different QoS services. Weighted Minmax algorithm (WMinmax) is proposed to efficiently and dynamically allocate network resources to heterogeneous and bursty traffic. In WMinmax, heterogeneous and regulated traffic are grouped into several classes according to their negotiated QoS parameters. For different classes, resources are allocated in proportion to their corresponding weights.

5.1 Introduction

To efficiently allocate network resources to bursty traffic, many dynamic resource allocation algorithms have been proposed [10, 35, 4]. The allocated bandwidth for each flow is proportional to its queue length in the Buffer–Population-Based Dynamic Slot Assignment algorithm [10] and the Generalized Longest Queue First (GLQF) algorithm [35]. In the Proportional Linear algorithm [4], in addition to the buffer occupancy, the instantaneous arrival rate is also included to calculate the bandwidth sharing of each flow. To allocate more bandwidth to the flow with a larger queue length, the proportional polynomial algorithm [4] was proposed, where the bandwidth allocation is in proportion to polynomials of the sum of the queue length and the instantaneous arrival rate. To achieve the desired goal of fair long–term buffer occupancy, the Minmax algorithm [4] was proposed by minimizing the maximum queue length of all contending flows. The proportional exponential algorithm [4] was introduced to reduce the computational

complexity of using the Minmax algorithm. It was shown that the Minmax algorithm has the best performance among these algorithms in terms of cell delay, delay jitter and cell loss rate [4]. However, the Minmax algorithm is limited to handling homogeneous traffic only. To be able to allocate resources for heterogeneous traffic that can be classified according to their negotiated QoS parameters, the Weighted Minmax (WMinmax) algorithm, which is the generalized version of the Minmax algorithm, is proposed and analyzed in this dissertation.

5.2 Weighted MinMax Algorithm

In this section, the Wminmax algorithm is presented.

5.2.1 Weighted Minmax Algorithm

Although the proposed algorithm is not limited to Variable Bit Rate (VBR) traffic, VBR traffic is used as the traffic model to simplify the description of the proposed algorithm. VBR traffics are characterized by three parameters: Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS) [68].

Following the model in [4], N heterogeneous traffic streams are assumed to share a link that is divided into time slots along the time axis. Each traffic stream is allocated a fixed bandwidth according to its SCR, and the dynamic allocation of the leftover bandwidth R to each stream is considered based on its queue length and instantaneous rate. N traffic streams are classified into J classes according to their negotiated parameters, with identical parameters, PCR, SCR, and MBS, in each class. There are N_j streams in class j, $j = 1, ..., J$, and $\sum_{j=1}^{J} N_j = N$. In WMinmax, a weight is assigned to each

class to make the bandwidth allocation relative to their negotiated parameters. Let L_i, λ_i and R_i be the queue length, the arrival rate, and the allocated bandwidth of stream i, respectively, and $i = 1, \ldots, N$. Also, let $Q_i = L_i + \lambda_i$, which is the estimated bandwidth needed to transmit the total traffic of stream i in next time slot. Then, the WMinmax algorithm is equivalent to solving the following optimization problem:

$$\text{Minimize } \{\max\{(Q_i - R_i) / w_i\}\} \tag{5.1}$$

subject to:

$$\sum_{i=1}^{N} R_i = R, \quad 0 \le R_i \le Q_i$$

where, w_i is the weight of stream i.

Note that WMinmax is a generalized version of the Minmax algorithm, i.e., the Minmax algorithm is a special case of WMinmax with all the weights being set to the same value. Like the Minmax algorithm, WMinmax assumes that all traffic conform to their negotiated parameters. Equation (5.1) will be referred to as the normalization procedure. After the normalization procedure, WMinmax algorithm can be implemented the same way as the Minmax algorithm. It is clear that the larger the weight assigned to a stream, the more bandwidth will be allocated to that stream.

5.2.2 An Example

The scenario given in [4] will be used to illustrate the Wminmax algorithm. There are two streams with rates $\lambda_1 = 1$ and $\lambda_2 = 2$ sharing the same output link with the rate $R = 2$.

First, all weights are assumed to be ones, and the initial queue lengths are zero. Since all weights are ones, the Minmax algorithm in [4] can be applied directly. At the first time slot, the expected queue lengths of the two streams are $Q_1 = 1$ and $Q_2 = 2$. So

the allocated bandwidth for the two streams are *1/2* and *3/2*, respectively. In the next time slot, actual queue lengths without including the estimated arrival rates are $L_1 = 0 + 1 - 1/2 = 1/2$ and $L_2 = 0 + 2 - 3/2 = 1/2$. In the following time slot, the queue lengths are $L_1 = 1/2 + 1 - 1/2 = 1$ and $L_2 = 1/2 + 2 - 3/2 = 1$. Note that although the arrival rate of the second stream is two times the first one. From the above example, the allocated rate for the second stream is three times the first stream. This is unfair in terms of equal allocation of bandwidth. However, as argued in [4], from the perspective of the buffer content, the Minmax algorithm is fair for all streams.

Second, with the same scenario, this dissertation considers the case where the weights for the two streams are different: $w_1 = 1/3$ and $w_2 = 2/3$. Since the Minmax algorithm can only support homogeneous traffic, the WMinmax algorithm has to be used to achieve the different requirements for the heterogeneous traffic. In the first time slot, bandwidth *2/3* and *4/3* are allocated to the two streams, and the queue lengths turn to be $L_1 = 0 + 1 - 2/3 = 1/3$ and $L_2 = 0 + 2 - 4/3 = 2/3$. In the next time slot 2/3 and 4/3 are allocated to them, and the queue lengths are $L_1 = 1/3 + 1 - 2/3 = 2/3$ and $L_2 = 2/3 + 2 - 4/3 = 4/3$. From the above example, it can conclude that a stream with a higher weight will obtain more bandwidth.

Comparing the above results of the two algorithms, it is clear that by introducing the weights to the corresponding streams, the WMinmax algorithm achieves better performance in terms of both bandwidth and buffer occupancies for streams with different negotiated parameters.

5.2.3 Analysis of the Wminmax Algorithm

Let $S_i = L_i \ / \ w_i$, $i=1,...,N$, where w_i is the corresponding weight of stream i, and $\sum_{i=1}^{m} w_i = 1$, $m \leq N$. So S_i is the normalized queue length divided by its corresponding weight. Assume S_i be ordered as $S_1 \geq S_2 \geq ... \geq S_N$, and the WMinmax algorithm produces queue lengths of N traffic streams as: $L_1, L_2, ..., L_N$. Then it is easy to reach the following Lemmas:

Lemma 5. 1. *There exists m and n, $m \leq n \leq N$, satisfying: $S_1 = ... = S_m = ... = S_n \geq S_{n+1} \geq ... \geq S_N$*

Proof:

Lemma 5.1 can be proved directly from the WMinmax algorithm, which allocates the available bandwidth to the first m largest streams, and makes their queue lengths equal with respect to their corresponding weights. Note that all the available bandwidths are completely allocated among the first m streams, and none are allocated to the remaining n-m streams that have equal queue lengths as the first m streams though. ■

Lemma 5. 2. *For any $k \leq n$,*

$$S_k = (Q_1 + Q_2 + ... + Q_m - R)$$

Proof:

Based on the fact that all available bandwidths are completely allocated to the first m largest queues, and none to the remaining N-m queues, thus:

$$\sum_{i=1}^{m} S_i w_i = \sum_{i=1}^{m} Q_i - R$$

Since $\sum_{i=1}^{m} w_i = 1$ and $S_1 = ... = S_m$

The following equation holds:

$$S_1 = \ldots = S_m = (Q_1 + Q_2 + \ldots + Q_m - R)$$

Since all the largest n streams have the same queue lengths,

$$S_1 = \ldots = S_m = \ldots = S_n = (Q_1 + Q_2 + \ldots + Q_m - R)$$ ■

Lemma 5. 3. *There exists m and n, $m \le n \le N$, such that for $k = 1, \ldots, n$,*

$$(Q_1 + Q_2 + \ldots + Q_m - R) < S_{k+1} \quad \text{if } k < m \tag{5.2}$$

$$(Q_1 + Q_2 + \ldots + Q_m - R) = S_{k+1} \quad \text{if } m \le k < n \tag{5.3}$$

$$(Q_1 + Q_2 + \ldots + Q_m - R) > S_{k+1} \quad \text{if } k = n \tag{5.4}$$

Proof:

For Equation (5.2), if there is a $k < m$ satisfying $(Q_1 + Q_2 + \ldots + Q_m - R) > Q_{k+1}$, then at least one stream $k = m - 1$ will not be allocated bandwidth. This contradicts the results of the WMinmax algorithm. Therefore, Equation (5.2) holds for all k smaller than m.

Equations (5.3) and (5.4) can be proved directly from Lemma 5.2 and Lemma 5.1, respectively. ■

Lemma 5. 3 provides a way to implement the WMinmax algorithm.

Lemma 5. 4. *If the queue lengths of N traffic streams with zero initial queue lengths yielded by the WMinmax algorithm are: $(L, \ldots, L, L_{n+1}, \ldots, L_N)$, $L > L_{n+1}$, then for any initial queue lengths in time slot j,* $\lim_{j \to \infty} \frac{L}{S_i^j} = w_i$, *if* $i \le n$

Proof:

First, the condition that all the weights are set to one is considered. Then, the N streams are divided into J subsets, and there are N_j streams in each subset with equal

arrival rates. Here, $\sum_{j=1}^{J} N_j = N$. For different subsets, the corresponding arrival rates are different and ordered as: $\lambda_{j1} < \lambda_{j2}$, for $j1 < j2$. Also, let L_i^j be the buffer content of the ith stream in subset j, it is necessary to show that for any subsets j_1 and j_2, $\forall i \in j_1$ and $\forall k \in j_2$, if $j_1 < j_2$, then $Q_i^{j_1} < Q_k^{j_2}$.

The following four cases will happen at the current time slot:

Case 1. Bandwidths are allocated to both stream i and k. Then, at the next time slot, the difference in queue length between the two streams remains the same as the previous time slot.

Case 2. Bandwidth is allocated to stream i and none to stream k (this case happens when $Q_i \geq Q_k$). The difference between the two is non-positive at the next time slot.

Case 3. Bandwidth is allocated to stream k and none to stream i (this case happens when $Q_i \leq Q_k$). The difference between Q_i and Q_j decreases at the next time slot.

Case 4, Bandwidth is not allocated to either stream. Since $\lambda_i > \lambda_j$, the difference between Q_i and Q_j decreases at the next time slot.

Thus, Q_i will be no less than Q_j after several time slots.

Second, from Lemma 5. 3, the difference between the largest mth streams and the $(m+1)$th stream will be negative at the next time slot. Therefore, all queue lengths will eventually become equal after a period of time.

Let us assign weights w_i and w_j to stream i and j, respectively. To keep the ratio of their queue lengths to be w_i / w_j, the allocated bandwidth should be different for $w_i \neq w_j$. Thus, the stream with the same arriving rates but different weights will have different queue lengths respective to their corresponding weights. ■

Since S_i is divided by its corresponding weight, it is the normalized queue length of stream i. From Lemma 5.4, it can conclude that the traffic with a larger weight will obtain more bandwidth than those with smaller weights so that the heterogeneous traffic can be efficiently supported by the WMinmax algorithm. The SCR of each stream can be selected as its corresponding weight.

5.3 Evaluation

For the simulation, a common link with *2*M bandwidth is used, and two ON-OFF traffic streams contend for the same out link. To find the worst-case performance, the traffic models are assumed to be extreme ON-OFF traffic [58].

In the first scenario, two streams have the same arrival traffic characters, i.e., both of them have the On time interval: $T_{on} = 0.5$ and Off time interval: $T_{off} = 0.5$. So, the average load $\rho = 0.5$. Peak rate $R_p = 5M$, and their weights are assigned to be *0.5* and *0.5*, respectively. In the second scenario, the two streams have the same On/Off time and the traffic load, but different peak rates: $R_{p1} = 6.5M$ and $R_{p2} = 3.5M$, and their weights are assigned to be *0.65* and *0.35*, respectively.

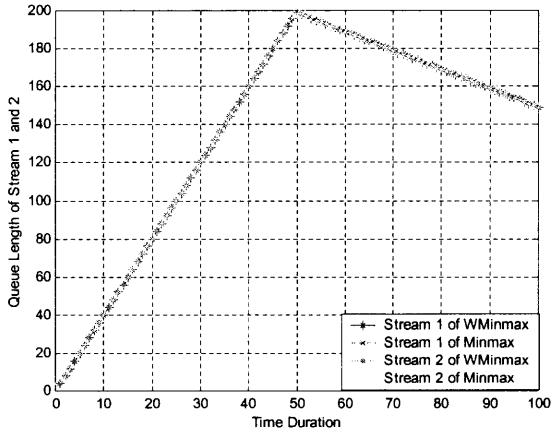

Figure 5.1 Queue lengths of stream 1 and 2 via WMinmax and Minmax algorithm for the first scenario.

As shown in Figure 5.1, if all traffic streams have the same traffic parameters and the same weights, the WMinmax and Minmax algorithms have the same performance. However, as shown in Figure 5.2, by assigning different weights to the traffic with different traffic characteristics, the WMinmax algorithm can more efficiently allocate bandwidth among the contending traffic.

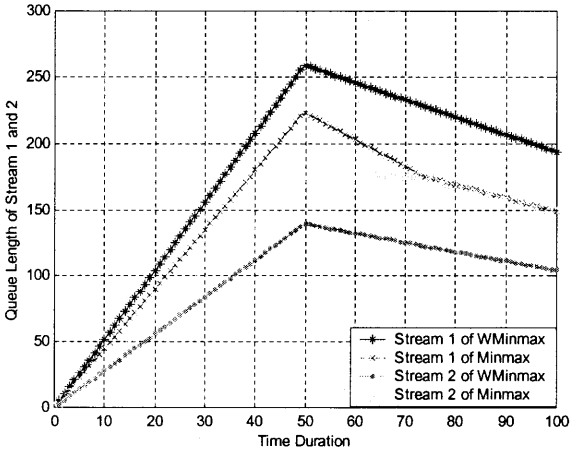

Figure 5.2 Queue lengths of stream 1 and 2 via WMinmax and Minmax algorithm for the second scenario.

5.4 Summary

To satisfy different QoS requirements of heterogeneous traffics, the WMinmax algorithm is proposed in this dissertation. Each class of traffic is assigned a weight, which reflects their respective bandwidth requirements. From the simulations, it can be concluded that, by using the WMinmax algorithm, different bandwidth requirements can be satisfied for the contending heterogeneous traffics.

CHAPTER 6

MULTIPLE DIMENSION SCHEDULING ALGORITHM

To efficiently support traffic with multiple priorities in integrated networks, a Multiple Dimension Scheduling (MDS) algorithm is proposed in this dissertation. In MDS, the criterion to select the packet to transmit is determined by multiple QoS parameters required by the traffic, and, therefore, it is possible to find the optimal scheduling region in the multiple dimension space.

6.1 Multiple QoS Priorities

Supporting traffic with diverse characteristics and requirements is a critical issue in integrated networks. Traffic requirements are usually represented by the Quality of Services (QoS) parameters, such as bandwidth, delay, delay jitter and packet loss ratio. The objective of the scheduling algorithms is to guarantee these QoS parameters by properly and efficiently selecting packets to transmit according to the criterion defined in the scheduling algorithms. Numerous scheduling algorithms have been proposed to achieve their desired features by selecting the corresponding criterion. By selecting the packet with the smallest deadline to transmit first, the Earliest Deadline First (EDF) algorithm is the optimal scheduling scheme to best meet the deterministic delay requirements for various classes of traffic [35]. The Longest Queue First (LQF) algorithm, on the other hand, can achieve the smallest cell loss among the proposed scheduling algorithms by tracking the information of queue lengths [53]. However, in real integrated networks, traffic may have several priorities. For instance, voice services are sensitive to the delay but can tolerate some packet losses while the data services can

tolerate longer delay but are sensitive to packet losses. Another practical issue involved in scheduling algorithms is the decouple problem, where traffic may require delay requirements completely unrelated to their transmission rates, and, therefore, rate based algorithms may not work efficiently under such scenario [17, 26]. This dissertation proposes a multiple-dimension scheduling algorithm that can efficiently solve these practical problems in integrated networks. Since the criteria to select the next transmission packet in EDF and LQF are only based on one category parameter, they are called One-Dimension algorithms. The proposed Multi-Dimension algorithm utilizes parameters from different categories to define the optimization criterion so that multiple priority requirements can be best satisfied. In MDS, all required QoS parameters are linearly integrated into one parameter. Based on the integrated parameters, the next packet for scheduling can be selected.

6.2 Multiple Dimension Scheduling Algorithms

To simplify the description of the proposed algorithm, it is first assumed that all traffics have only two different priority requirements, specifically, delay and packet loss ratio. Later, the algorithm will be extended to the case where more than two priority requirements are required by the traffic.

6.2.1 Two-Dimension Algorithm for Output-Queued Switches

In this subsection, the MDS algorithm is first presented for an Output-Queued (OQ) switch. Note that, for an N by N OQ switch, the processing speed of the data line inside the switching fabric and the rate to access the buffer may be required to run, in the worst case, N times the outside line rate. To support different classes of traffic, virtual queues

are required for all classes. Let L^k and D^k be the packet loss and delay requirements of class K, respectively. Let $\overline{L^k}$ be the normalized difference of the required and the current packet loss ratio, and $\overline{D^k}$ be the deadline of the head-of-line packet in each virtual queue. An integrated parameter $\overline{I^k}$ should be defined as:

$$\overline{I^k} = \overline{L^k} \times \mu + (D - \overline{D^k}) \tag{6.1}$$

where μ is the linear factor, and D is a constant larger than any $\overline{D^k}$. Two thresholds for either $\overline{L_{i,j}^k}$ or $D - \overline{D_{i,j}^k}$ are also necessary along with Equation (6.1) as the criteria to select the next transmission packet. They are used to prevent the violation from either the delay or the packet loss ratio. The packet with either parameter smaller than the corresponding threshold is considered as the emergent packet and the scheduler will first selects the emergent packet as the next one to transmit. If there are no emergent packets, the scheduler will select the packet with the largest $\overline{I_{i,j}^k}$ as the next one to transmit.

6.2.2 Two-Dimension Algorithm for Input-Queued Switches

Without the speedup constraint, input-queued switches are more suitable to be employed in high-speed network than OQ switches. The Head-of-Line (HOL) blocking problem can be simply eliminated by employing the Virtual Output Queuing (VOQ) scheme, where each input maintains a separate virtual queue for each output. Several algorithms have been proposed for matching the inputs to the outputs. To avoid the starvation problem involved in the Maximum Size Matching (MSM), the Maximum Weighted Matching (MWM) scheme is proposed to find a matching with a maximum sum of weights rather

than a maximum size [53]. The model of an IQ switch in this dissertation is assumed to be VOQ, and each class has its own virtual queue in each input port to support multiple classes of services. Also let $L_{i,j}^{k}$ and $D_{i,j}^{k}$ be the packet loss and delay requirements of class k originating from input i and destined to output j, respectively. Here, $k=1, 2, \ldots, K_i$, and $i, j = 1, \ldots, N$. K_i is the total number of classes from input i to output j, and N is the size of the switch. Let $\overline{L_{i,j}^{k}}$ be the normalized difference of the required and the current packet loss, and $\overline{D_{i,j}^{k}}$ be the deadline of the head-of-line packet in each queue. Then, an integrated parameter $\overline{L_{i,j}^{k}}$ can be obtained according to the following equation:

$$\overline{I_{i,j}^{k}} = \overline{L_{i,j}^{k}} \times \mu + (D - \overline{D_{i,j}^{k}}) \tag{6.2}$$

where μ is the linear factor, and D is a constant larger than any $\overline{D_{i,j}^{k}}$. Selecting the integrated parameters as the weights, a matching can be found based on the existing MWM algorithms. Note that the packet loss ratio and the time of the head of line packet at each queue are needed to find the proper matching. However, the buffer management and the matching algorithm should be executed by different agents to reduce the computation complexity. Also, the pipeline and critical links schemes may be the effective solutions to further reduce the implementational computation complexity.

Lemma 6. 1. *As compared to MWM, for the worst case, 50% capacity of the IQ switch can be reserved for each QoS category if a two-dimension scheduling algorithm is employed.*

Proof:

In the worst case, two categories are completely orthogonal, and therefore, each category has to reserve its requirement independent to another one. Therefore, 50% of the capacity of the IQ switch can be reserved for each category. ■

Lemma 6. 1 is useful for the admission control of QoS aware traffic such as voice and video. The best-effort traffic, however, are not considered in Lemma 6.1.

Note that if the parameter from one category can be expressed linearly by the parameter from another category, the MDS algorithm turns into an one-dimension algorithm. Also, the MDS algorithm aims to find the optimal scheduling region in terms of both cell loss and delay requirements for any given traffic conditions rather than an admission control region.

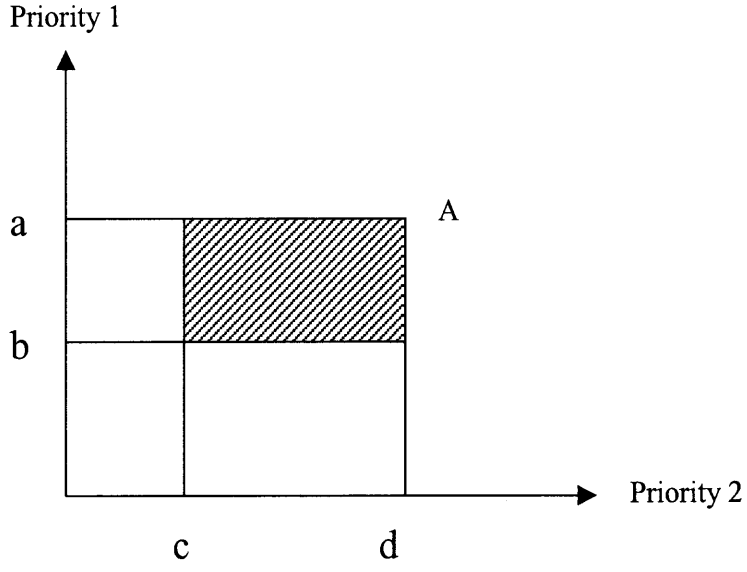

Figure 6.1 Scheduling region for two-priority traffic.

Figure 6.1 shows the optimal scheduling region for the traffic with two priorities. The shaded area *abcd* is the scheduling region, and point A is the optimal point.

The algorithm to support traffic with more than two priorities can be implemented by extending Equation (6.2) into

$$\overline{I_{i,j}^{k}} = \sum_{m=1}^{n} C_m \overline{A_{i,j}^{m,k}} \tag{6.3}$$

where, $\overline{A_{i,j}^{m,k}}$ and C_m are the urgency factor (the larger, the higher priority) of class m and the corresponding linear factor, respectively. Together with Equation (6.3), using the integrated parameter as the weights, the MWM can be obtained for scheduling the packets.

6.3 Conclusions

The multiple dimension scheduling algorithm, which aims to find the optimal scheduling region in the multiple dimensional space, has been proposed in this Chapter. The algorithm supports traffic with multiple priorities and is also able to handle the decouple problem in practice.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

Scheduling algorithms for input queued switches have been investigated in this dissertation. Based on the fact that buffers inside switches may not be able to accommodate all traffic, scheduling algorithms with limited buffer space have been studied. By considering the buffer management schemes and scheduling algorithms together, the main goal of this dissertation is to improve the utilization of both buffer space and bandwidth so that more traffic can be accommodated with limited resources.

Chapter 2 presented existing scheduling algorithms for input queued switches. Existing scheduling algorithms are classified into three categories: bipartite graph matching algorithms, parallel iterative matching algorithms, and QoS features guaranteed algorithms. Also, chapter 2 briefly reviewed existing buffer management schemes for the shared buffer structure. The packet discarding schemes, which are implemented at the gateway of current Internet networks, were discussed in this chapter.

Chapter 3 presented Queue Length Proportional (QLP) assignment algorithm, which allocates the available bandwidth to the competing connections in proportion to their corresponding buffer occupancies. High utilization of both buffer space and bandwidth can be obtained by using the QLP algorithm so that both high throughput and low cell loss ratio can be achieved.

In Chapter 4, cell loss ratio is employed as the fairness metric resulting in the Queue Proportional Fairness (QPF) criterion. Three types of fairness are introduced for an input queued switch: intra-queue, intra-input and inter-input fairness. As shown in this

chapter, by employing the new metric, the QPF can achieve higher utilization of network resources than the traditional Max-Min fairness criterion.

Chapter 5 presented the Weighted Minmax (Wminmax) algorithm to support different QoS requirements of heterogeneous traffic. In Wmimax, heterogeneous and regulated traffics are grouped into several classes according to their negotiated QoS parameters. For each class, network resources are allocated in proportion to their corresponding QoS parameters.

Chapter 6 presented Multiple Dimension Scheduling (MDS) to support traffic with multiple priorities and to handle the decouple problem in practice. In MDS, the criterion to select the packet to transmit is determined by multiple QoS parameters and, therefore, it is possible to find the optimal scheduling region in the multiple Dimension space.

The future work of this dissertation will focus on how to obtain the closed form relationship between queue lengths and buffer size for any 2 by 2 input queued switches. For switches with size larger than 2, the scheduling algorithm may be heavily dependent on the traffic pattern, and thus it is difficult to obtain the closed form solution. In the MDS algorithm, the selection of the linear factors is one of the critical steps that would affect the performance of the algorithm. Note that, the integrated parameters are based on different categories of QoS parameters in the MDS algorithm, research on the effect of any inaccuracies in the QoS parameters are important for improving the performance of the MDS algorithm.

REFERENCES

1. T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. Computer Systems*, Vol. 11, No. 4, pp. 319-352, November, 1993.

2. Peter Ashwood-Smith, Daniel Awduche, *et al.*, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," *IEFT draft-many-gmpls-architecture-00.txt*, February, 2001.

3. D. Bersekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

4. S. Biswas and R. Izmailov: "Design of a Fair Bandwidth Allocation Policy for VBR Traffic in ATM Networks," *IEEE/ACM Trans. On Networking*, Vol. 8, No. 2, pp. 212-223, April, 2000.

5. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC2475*, December, 1998.

6. C.S. Chang, W.J. Chen, and H.Y. Huang, "On service guarantees for input buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann," *Proc. IEEE IWQoS'99*, London, U.K, pp. 79-86, 1999.

7. C.S. Chang, W.J. Chen, and H.Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," *Proc. INFOCOM'00*, Tel Aviv, Israel, pp. 1614-1623, March, 2000.

8. S. Cheung and C. Pencea, "Pipelined Sections: A new buffer management discipline for scalable QoS provision," *Proc. INFOCOM'01*, pp. 1530-1538, 2001.

9. A. Choudhury and E. Hahne, "Dynamic queue length thresholds for shared-memory packet switches," *IEEE/ACM Trans. Networking*, Vol. 6, No. 4, pp. 130-140, April, 1998.

10. S. Chowdhury and K. Sohraby: "Alternative Bandwidth Algorithm for Packet Video in ATM Networks," *Proc. INFOCOM'92*, pp.1061-1068, 1992.

11. I. Cidon, L. Georgiadis and R. Guerin, "Optimal Buffer Sharing," *Proc. IEEE INFOCOMM'95*, pp. 24-31, 1995.

12. T. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithm*, The MIT Press, Cambridge, Massachusetts, March, 1990.

13. A. Elwalid, D. Mitra, and R. Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node," *IEEE Journal on Selected Areas in Communications*, Vol. 13, Issue: 6, pp. 1115 –1127, August, 1995.

14. R. Fan, A. Ishii, B. Mark, G. Ramamurthy, Q. Ren, "An optimal buffer management scheme with dynamic thresholds," *Proc. IEEE GLOBECOM '99*, Vol.1, pp. 631-637, 1999.

15. S. Floyed and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking*, pp. 397-413, August, 1993.

16. G. Foschini, B. Gopinath and J. Hayes, "Optimum allocation of severs to two types of competing customers," *IEEE/ACM Trans. On Commun.*, Vol. 29, pp. 1051-1055, 1981.

17. A. Francini and F. Chiussi, "A Weighted fair queueing scheduler with decoupled bandwidth and delay guarantees for the support of voice traffic," *Proc. IEEE GLOBECOM'01*, Vol. 3, pp. 1821-1827, 2001.

18. D. Gale and L.S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, Vol. 69, pp. 9-15, 1962.

19. N. Ge, J.K. Muppala, and M. Hamdi, "Analysis of non-blocking ATM switches with multiple input queues," *Proc. IEEE GLOBECOM'97*, pp. 531-535, 1997.

20. S. Golestani, "A Self-clocked fairness queueing scheme for broadband application," *Proc. INFOCOM'94*, pp. 636-646, 1994.

21. M.W. Goudreau, S.G. Kolliopoulos, and S.B. Rao, "Scheduling algorithms for input-queued switches: Randomized techniques and experimental evaluation," *Proc. INFOCOM'00*, Tel Aviv, Israel, pp. 1634-1643, March, 2000.

22. R. Guerin, S. Kamat, V. Peris and R. Rajan, "Scalable QoS Provision Through Buffer Management," *Proc. IEEE SIGCOMM'98*, pp. 29-40, 1998.

23. E. Hashem, "Analysis of Random Drop for Gateway Congestion Control," *Report LCS TR-465, Lab. for Computer Science, MIT*, P. 103, 1989.

24. J.E. Hopcroft and R.M. Karp, "An $n^{2.5}$ algorithm for maximum matching in bipartite graphs," *Society for Industrial and Applied Mathematics J. Comput.*, Vol. 2, pp. 225-231, 1973.

25. R. Jain, D. Chiu and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," *DEC Tech. Report 301*, 1984.

26. A. Kam and K.-Y. Siu, "Linear Complexity Algorithms for QoS Support in Input-Queued Switches with no Speedup," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 6, June, 1999.

27. F. Kamoun and L. Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic conditions," *IEEE/ACM Trans. Commun.*, Vol. 28, pp. 992-1003, 1980.

28. M. Karam, F. Tobagi, "Rate and Queue Controlled Random Drop (EQRD): A Buffer management Scheme for Internet Routers," *Proc. IEEE GLOBECOM '00*, Vol. 1, pp. 316-322, 2000.

29. M. Karol, M. Hluchyi, and S. Mogan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, Vol. COM-35, pp. 1347-1356, December, 1987.

30. F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. On Telecomm.* Vol. 8, pp. 33-37, 1997.

31. H. Kim, K. Shim and J. Lim, "Fairness concept in terms of utilisation," *IEE Electronics Letters*, Vol. 36, No. 4, pp. 379-381, February, 2000.

32. K. Kou, "Realization of Large-capacity ATM Switches," *IEEE communication magazine*, pp. 120-123, December, 1999.

33. G. Lapiotis and S. Panwar, "Quality of service analysis of shared buffer management policies combined with generalized processor sharing," *Proc. IEEE GLOBECOM'99*, Vol. 1a , pp. 37-43, 1999.

34. G. Lapiotis, "Stochastic Analysis of Joint Buffer Management and Service Scheduling in High-Speed Network Nodes," *Ph. D Dissertation*, Polytechnic University, 1999.

35. D. Lee, "Generalized Longest Queue First: An Adaptive Scheduling Discipline fro ATM Networks," *Proc. INFOCOM'97*, pp. 318-325, 1997

36. S. Li and N. Ansari, "Input queued switching with QoS guarantees," *Proc. INFOCOM'99*, New York, NY, pp. 1152-1159, March, 1999.

37. S. Li and N. Ansari, "Provisioning QoS features for input-queued ATM switches," *IEE Electronics Letters*, Vol. 34, Issue:19, pp. 1826-1827, September, 1998.

38. S.Q. Li, "Performance of a nonblocking space-division packet switch with correlated input traffic," *Proc. GLOBECOM'89*, pp. 1754-1763, 1989.

39. S.Y. Liew, S.W. Cheng, and T.T. Lee, "An enhanced iterative scheduling algorithm for ATM input-buffered switch," *Proc. ATM Workshop*, pp. 103-108, 1999.

40. C. Liu and J. Layland, "Scheduling Algorithms for multiprogramming in a Hard Real-Time Environment," *Journal of the ACM*, Vol.20, No. 1, pp. 46-61. 1973.

41. D. Liu, N. Ansari, and E. Hou, "A novel fairness criterion for input queued switches," *IEEE Military Communications Conference*, Vol. 2, pp. 1474-1478, McLean, VA, October, 2001.

42. D. Liu, N. Ansari, and E. Hou, "QLP: A Joint Buffer Management and Scheduling Scheme for Input Queued Switches," *IEEE Workshop on High Performance Switching and Routing, 2001*, pp. 164-168, Dallas, TX, May, 2001.

43. D. Liu, N. Ansari, and E. Hou, "A Novel Algorithm for Resource Allocation for Heterogeneous Traffic," *Conference on Information Sciences and Systems*, pp. 382-385, Princeton, March, 2002.

44. D. Liu, N. Ansari, and E. Hou, "Fairness Criterion for Allocating Resources in Input Queued Switches," *IEE Electronics Letters*, Vol. 37, No. 19, pp. 1205-1206, September, 2001.

45. M. Marsan, A. Bianco, E. Leonardi and L. Milia, "RPA: A Flexible Scheduling Algorithm for Input Buffered Switches," IEEE Transaction on Communications, Vol. 47, No. 12, pp. 1921-1933, December, 1999.

46. L. Massoulie and J. Roberts, "Bandwidth Sharing: Objectives and Algorithms," *Proc. INFOCOM'99*, pp. 1395-1403, 1999.

47. M. May, J. Bolot, C. Diot, B. Lyles, "Reasons Not to Deploy RED," *Proc. IEEE IWQoS'99,* March, 1999.

48. A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in Input-Queued Switches," *Proc. INFOCOM'98*, San Francisco, CA, pp. 792-799, March, 1998.

49. A. Mekkittikul and N. McKeown, "A starvation-free algorithm for achieving 100% throughput in an input-queued switch," *Proc. ICCCN'96*, pp. 226-231, October, 1996.

50. N. McKeown, "Scheduling algorithm for input-queued cell switches," *Ph. D Dissertation*, Univ. of California, Berkeley, 1995.

51. N. Mckeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches," *IEEE/ACM Trans. Networking*, Vol. 7, No. 2, pp. 188-201, April, 1999.

52. N. Mckeown, A, Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE/ACM Trans. On Commun.*, Vol. 47, pp. 1260-1267, August, 1999

53. N. McKeown, P. Varaiya, and J. Walrand, "Scheduling Cells in an Input-Queued Switch," *IEE Electronics Letters*, Vol. 29, No. 25, pp. 2174-2175, December, 1993.

54. N. Mckeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Proc. INFOCOM'96*, San Francisco, CA, pp. 296-302, March, 1996

55. S. Motoyama, D.W. Petr, and V.S. Frost, "Input-queued switch based on a scheduling algorithm," *IEE Electronics Letters*, Vol. 31, No. 14, pp. 1127-1128, July, 1995.

56. S. Motoyama, L.M. Ono, and M.C. Mavigno, "An iterative cell scheduling algorithm ofr ATM input-queued switch with service class priority," *IEEE Communications Letters*, Vol. 03, No. 11, pp. 323-325, November, 1999.

57. S. Motoyama, L.M. Ono, and M.C. Mavigno, "Performance analysis of iterative scheduling algorithms for ATM input-queued switches," *Proc. ITS'98*, pp. 195-200. 1998.

58. P. Newman, G. Minshall, T. Lyon "IP Switching - ATM Under IP," *IEEE/ACM Trans. Networking*, pp. 117-129, April, 1998

59. J. Nho, Y. Lee and K. Kim, "Congestion control with a new fairness criterion for multicast ABR service in ATM networks," *IEE Proc. of Commun.*, Vol. 146, No. 3, pp. 181-184, June, 1999.

60. G. Parulkar, D. Schmidt, and J. Turner, "IP/ATM: a Strategy for Integrating IP with ATM," *Proc. ACM SIGCOM*, Cambridge MA, pp. 49-56, September, 1995.

61. Y. Pekhter, B. Davie, D. Katz, E. Rosen and G. Swallow, "Cisco System' Tag Switching Architecture overview," *IETF RFC 2105*, February, 1997.

62. F. Presti, Z. Zhang, J. Kurose and D. Towsley: "Source Time Scale and Optimal Buffer/Bandwidth Tradeoff for Heterogeneous regulated Traffic in a Network Node," *IEEE/ACM Trans. On Networking*, Vol. 7, No. 4, pp. 490-501, August, 1999.

63. R. Schoenen, G. Post, G. Sander, "Prioritized arbitration for input-queued switches with 100% throughput," *Proc. ATM Workshop*, pp. 253-258, 1999.

64. D.N. Serpanos and P.I. Antoniadis, "FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues," *Proc. INFOCOM'00*, pp. 548-555, 2000.

65. S. Sharma and Y. Viniotis, "Optimal Buffer Management policies for Shared-Buffer ATM Switches," *IEEE/ACM Trans. Networking*, Vol. 7, No. 4, pp. 575-587, August, 1999.

66. D. Stephens and H. Zhang, "Implementing Distributed Packet Fair Queueing in a Scalable Switch Architecture," *Proc. INFOCOM'98*, pp. 282-290, 1998.

67. D. Stiliadis and A. Varma, "Providing Bandwidth Guarantees in an Input Buffered Crossbar Switch," *Proc. INFOCOM'95*, Boston, MA, April, 1995.

68. I. Stoica, S. Shenker, H. Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks," *Proc. ACM SIGCOM'98,* 1998.

69. Y. Tamir and G.L. Frazier, "High-performance multi-queue buffers for VLSI communication switches," *Proc. 15th Annu. Int. Symp. Comput. Architecture*, Honolulu, HI, pp. 343-354, May, 1988.

70. R.E. Tarjan, *Data structures and network algorithms*, Society for Industrial and Applied Mathematics, Pennsylvania, November, 1983.

71. F. Tobagi, "Fast Packet Switch Architectures for Broad-band Integrated Services Digital Networks," *Proc. of the IEEE*, Col. 78, No. 1, pp. 133-167, January, 1990.

72. Traffic Management Specification Version 4.1, *af-tm-0121.000, The ATM Forum*, March, 1999.

73. S. Wei, E. Coyle, and M. Hsiao, "An optimal buffer sharing," *Proc. IEEE GLOBECOM'91*, pp. 924-928, December, 1991.

74. G. Wu and J. Mark, "A Buffer Allocation Scheme for ATM Networks: Complete Sharing Based on Virtual Partition," *IEEE/ACM Trans. Networking,* Vol. 3, No. 6, pp. 660-670, December, 1995.

75. L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine,* Vol. 7, pp. 8-18, September, 1993.