

國立交通大學

電信工程研究所

博士論文

具服務品質保證之
下世代光纖網路訊務控制機制

**Traffic Control Schemes for
Next-Generation Optical Networks with
QoS-Provisioning**

研究生：唐文祥

指導教授：張仲儒老師

中華民國 99 年 01 月

具服務品質保證之下世代光纖網路訊務控制機制

Traffic Control Schemes for Next-Generation
Optical Networks with QoS-Provisioning

研究生：唐文祥

Student: Wen-Shiang Tang

指導教授：張仲儒 博士

Advisor: Dr. Chung-Ju Chang

國立交通大學

電信工程研究所

博士論文

A Dissertation

Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University

in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in

Communication Engineering
Hsinchu, Taiwan

2010 年 01 月

具服務品質保證之下世代光纖網路訊務控制機制

研究生：唐文祥

指導教授：張仲儒 博士

國立交通大學電機工程學系博士班

摘要

本論文的主要專注在下一代光纖網路的訊務控制機制，其中討論的網路系統包含有光群聚交換骨幹網路(optical burst switching backbone network)、高速都會區域網路(metropolitan area network, MAN)中的封包彈性環(resilient packet ring, RPR)、以及橋接式封包彈性環(bridged resilient packet ring, BRPR)。

我們首先探討光群聚交換骨幹網路中的訊務控制機制。該交換系統兼具光電路交換系統(optical circuit switching, OCS)和光封包交換系統(optical packet switching, OPS)的優點，且所需相關的光處理器也已開發，因此此交換機制比較受青睞。在光群聚交換骨幹網路中，頻寬的分配只要是以預先保留(Reservation)的方式來處理巨集封包(Burst)，再加入了光的緩衝器(Fiber Delay Line)可以使用的比較晚到或具較低優先權的封包可以順利的傳送出去。在這樣架構下的考量，我們設計一種具光緩衝器分配的權限群聚排程演算法(priority burst scheduling with FDL assignment, PBS-FA)。其主要設計理念是想讓具較高優先權的群聚必要情況下可以強制取代已保留給低優先權群聚或群聚長度較短但高優先群聚的頻寬，之後再對被犧牲的群聚進行了補償。

在本篇論文的第二部份，探討高速都會區域網路中的彈性分封環(Resilient Packet Ring)。在彈性分封環中訊務控制所需考慮的議題主要希望可以達到公平性的頻寬分配並且可以快速穩定各訊務流。我們提出一個高效能之公平流速產生器(fuzzy local fairRate generator, FLAG)，藉著之晰運作機制產生一個準確的本地公平流速來抑制壅塞情況並且達成上述考量。所提出的機制，是由三個部份所組成，適應性公平流速計算器(adaptive fairRate calculator, AFC)、之晰壅塞偵測器(fuzzy congestion detector, FCD)、與之晰公平流速計算器(fuzzy fairRate generator, FFG)。適應性公平流速計算器產生一個評估過的公平流速而之晰壅塞偵測器根據次級傳輸緩衝器(STQ)的容納量與接收到的流量大小來指出當前的壅塞程度。之晰公平流速計算器經由考量兩項由適應性公平流速計算器與之晰壅塞偵測器輸出的結果來得到反映真實流量狀況的本地公平流速。藉由適應性公平流速計算器與之晰壅塞偵測器的使用，之晰公平流速產生器可產生較小的收斂時間，再者當與其它演算法相比，在不同大小的壅塞區域中皆獲得極好的效果。

最後，我們探討由橋接器(bridge)鍵連多個彈性分封環而成的橋接式封包彈性環(bridged resilient packet ring, BRPR)中的路由問題。在此環境中，我們基於載量均衡原則(the load balancing principle)提出一個智慧型跨環路由控制法。該智慧型跨環路由控制法不只同時考慮橋接器以及下游擱點壅塞的情況並且同時考量橋接器的服務速率以及訊務終點站與橋接器的

距離。此路由控制法主要包含三個部分：一個是模糊橋接器擁塞指示器(fuzzy bridge-node congestion indicator, FBCI)、一個是平行串列遞迴類神經網路下游擷點公平性預測器(pipeline recurrent neural networks (PRNN) downstream-node fairness predictor, PDFP)、一個是模糊由路控制器(fuzzy route controller, FRC)。從模擬結果來看，該智慧型跨環路由控制法明顯改善佇列長度閾值控制器(queue length threshold route controller, QTRC)以及最短路徑控制器(the shortest path route controller, SPRC)很多。



Traffic Control Schemes for Next-Generation Optical Networks with QoS-provisioning

Student: Wen-Shiang Tang

Advisor: Dr. Chung-Ju Chang

Department of Electrical Engineering
National Chiao Tung University

Abstract

This dissertation is aimed at traffic control issue in the next-generation optical network for the optical burst switching (OBS) core network, the resilient packet ring (RPR), which is a metropolitan area network (MAN), and the bridged resilient packet ring (BRPR).

First, we propose a priority burst scheduling with fiber delay line (FDL) assignment (PBS-FA) for the OBS core network. It allows not only high-priority bursts to preempt low-priority ones but also longer high-priority bursts to preempt shorter high-priority ones. Meanwhile it schedules or reschedules these bursts by using FDL assignment. Simulation results reveal that the PBS-FA achieves the higher system throughput and the less average system dropping probability less than a preemptive latest available unused channel with void filling (PLAUC-VF) scheme.

Second, we propose a local fairRate generator using fuzzy logics and moving average technique for the RPR to achieve the congestion control. The fuzzy local fairRate generator (FLAG) is designed to achieve both low convergence time and high system throughput, besides fairness. It contains three functional blocks:

an adaptive fairRate calculator (AFC) to properly pre-produce a local fairRate by moving average technique; a fuzzy congestion detector (FCD) to intelligently estimate the congestion degree of station; finally, a fuzzy fairRate generator (FFG) to precisely generate the local fairRate. Simulation results show that only the FLAG can stabilize all flows in parking lot scenarios with different finite traffic demands, compared to conventional the aggressive mode (AM) and distributed bandwidth allocation (DBA) fairness algorithms.

Finally, we propose an intelligent inter-ring route control, employed in the bridges which connect two resilient packet rings (RPRs), for the BRPR. The intelligent inter-ring route controller (IIRC) is designed according to the load balancing principle, where the IIRC considers not only the congestion degree of both bridge and its downstream nodes but also the service rate and the number of hops to destination. It contains three functional blocks implemented by fuzzy logic systems or pipeline recurrent neural networks (PRNN). A fuzzy bridge-node congestion indicator (FBCI) is to detect the congestion degree of the bridge, a PRNN downstream-node fairness predictor (PDFP) is to predict the mean received fairRate from downstream nodes, and a fuzzy route controller (FRC) is to determine a preference value of route according to the congestion indication, the predicted mean received fairRate, the service rate of the bridge, and the number of hops to destination. Simulation results show that the IIRC improves the performances in the packet dropping probability, the average packet delay, and the throughput over the queue length threshold route controller (QTRC) and the shortest path route controller (SPRC).

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor, Dr. Chung-Ju Chang, for the patient guidance and concern over the reach details and methodology. His attentive and professional attitude is always the quintessence of imitation.

Special thanks go my colleagues in the Broadband Communication Lab. and all of my friends, for their genuine encouragement, kind help, and sweet memories. Their assistance is always helpful and warm.

Finally, I am deeply indebted to my family for their love and understanding. This dissertation is dedicated to my parents. Without their wholehearted care and full support, it is impossible for me to exploit the life in an unburdened way.

Contents

Mandarin Abstract	i
English Abstract	iii
Acknowledgements	v
Contents	vi
List of Figures	ix
List of Tables	xii
1 Introduction	1
1.1 Motivation	1
1.2 Paper Survey	6
1.2.1 Burst Scheduling in Optical Burst Switching Networks	6
1.2.2 Traffic control in Resilient Packet Ring	9
1.2.3 Traffic control in Bridged Resilient Packet Rings	12
1.3 Dissertation Organization	13
2 Priority Burst Scheduling with FDL Assignment for Optical Burst	

Switching Networks	15
2.1 Introduction	15
2.2 Architecture of Intermediate OBS Node	17
2.3 Priority Burst Scheduling with FDL Assignment (PBS-FA)	19
2.4 Simulation Results	23
2.5 Concluding Remarks	25
3 FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Ring	26
3.1 Introduction	26
3.2 RPR System Model	30
3.3 Fuzzy Logic System	33
3.3.1 Fuzzy Inference System (FIS)	33
3.3.2 Mamdani Fuzzy Model	35
3.4 Fuzzy Local FairRate Generator	37
3.4.1 Adaptive fairRate Calculator (AFC)	38
3.4.2 Fuzzy Congestion Detector (FCD)	40
3.4.3 Fuzzy fairRate Generator (FFG)	43
3.5 Simulation Results and Discussions	46
3.6 Concluding Remarks	58
4 Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings	60
4.1 Introduction	60
4.2 System Model	63

4.2.1	Architecture of Bridge Node	63
4.2.2	Fairness Algorithm	65
4.3	Neural Network Controller	66
4.3.1	Neural Networks and its Learning Capability	66
4.3.2	Multilayer Feedforward Neural Networks	69
4.3.3	Radial Basis Function Neural Networks	72
4.4	Intelligent Inter-Ring Route Controller	75
4.4.1	Fuzzy Bridge-Node Congestion Indicator (FBCI)	76
4.4.2	PRNN Downstream-Node Fairness Predictor (PDFP)	81
4.4.3	Fuzzy Route Controller (FRC)	82
4.5	Simulation Results	85
4.6	Concluding Remarks	93
5	Conclusions and Future Works	95
	Bibliography	99
	Vita	107

List of Figures

1.1	RPR structure	4
1.2	BRPR structure	7
2.1	Architecture of the OBS node	18
2.2	The flowchart of the PBS-FA scheme	20
2.3	The flowchart of the PBS-FA scheme	22
2.4	The system throughput	23
2.5	The average system dropping probability	24
3.1	Resilient packet ring structure	31
3.2	RPR station structure	32
3.3	The basic structure of fuzzy inference system	35
3.4	An example of Mamdani fuzzy model	36
3.5	Definitions for functions $f(\cdot)$ and $g(\cdot)$	37
3.6	Functional blocks of FLAG	39
3.7	The membership functions of the term set (a) $T(L_s(n))$ (b) $T(A_s(n))$ (c) $T(D_c(n))$	41
3.8	The membership functions of the term set (a) $T(f_p(n))$ (b) $T(D_c(n))$ (c) $T(f_l(n))$	45

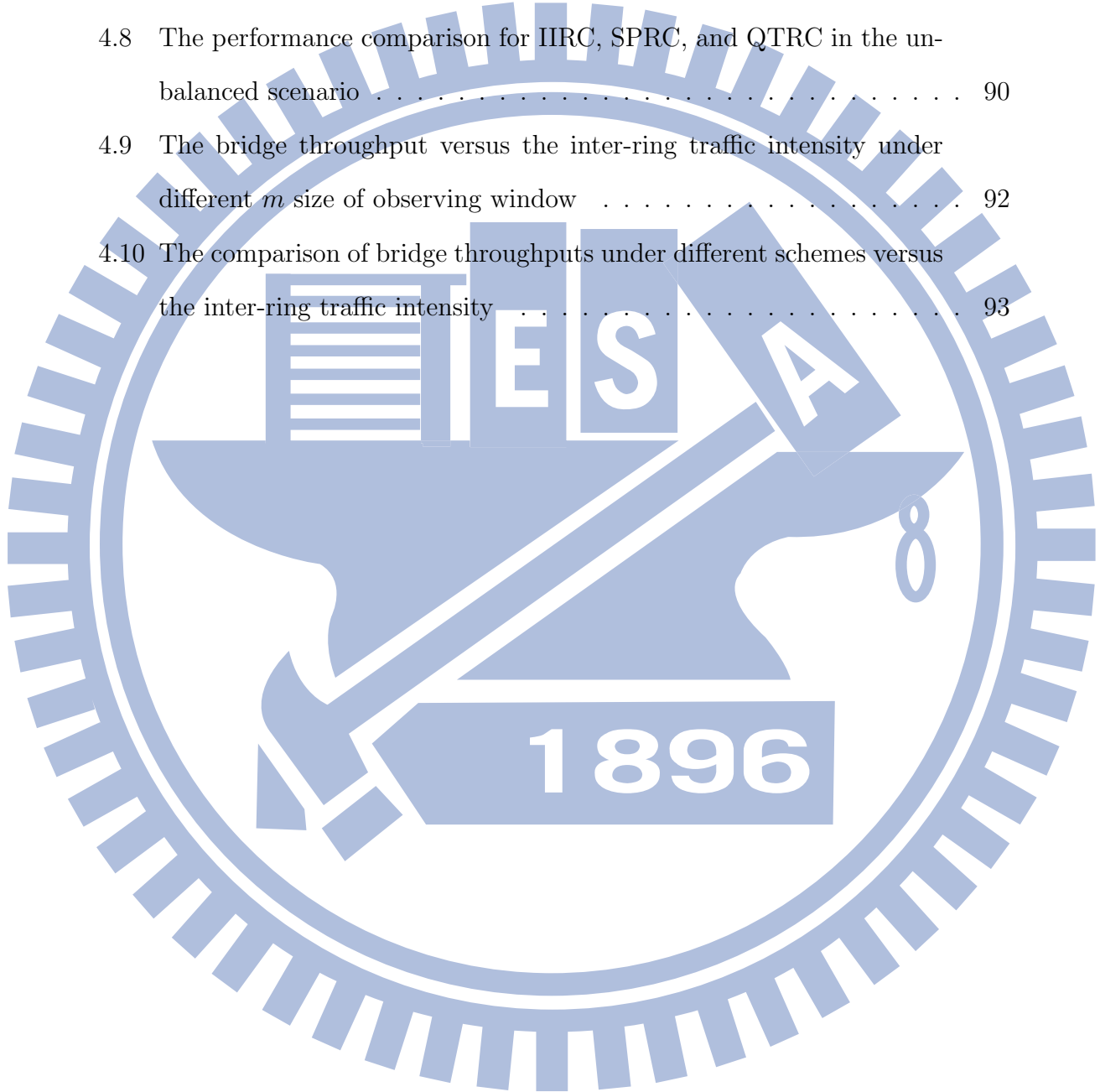
3.9	(a) Small parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DBA with moving average (DMA), and (e) FLAG.	48
3.10	(a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.	50
3.11	(a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG in a large parking lot scenario with various finite traffic flows.	51
3.12	The throughputs of (a) AM, (b) DBA, (c) DMA, and (d) FLAG in a large parking lot scenario containing 8 stations, where each flow is with truncated Pareto traffic model.	53
3.13	(a) Available bandwidth reclaiming scenario with finite traffic demand, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.	55
3.14	(a) Available bandwidth reclaiming scenario with finite traffic demand and two reuse traffic flows, and the throughput of (b) AM, (c) DBA, (d) DMA, (e) FLAG and (f) M-FLAG.	57
4.1	Architecture of the bridge node	64
4.2	Architecture of the interface	65
4.3	The basic structure of neural network	67
4.4	The structure of multilayer feedforward neural network	70
4.5	The structure of RBFN controller	73
4.6	Intelligent inter-ring route controller (IIRC)	76

4.7 The performance comparison for IIRC, SPRC, and QTRC in the balanced scenario 88

4.8 The performance comparison for IIRC, SPRC, and QTRC in the unbalanced scenario 90

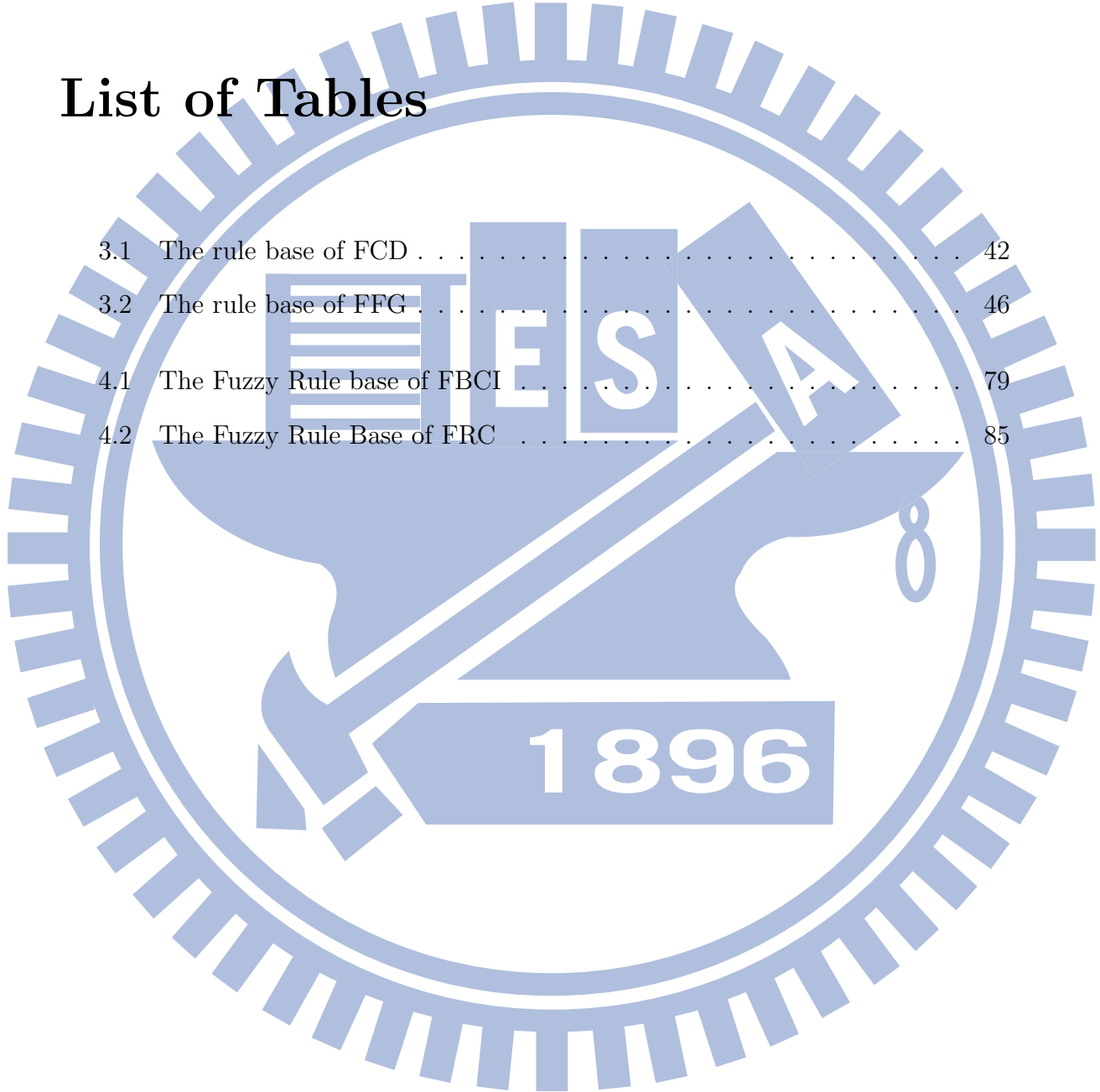
4.9 The bridge throughput versus the inter-ring traffic intensity under different m size of observing window 92

4.10 The comparison of bridge throughputs under different schemes versus the inter-ring traffic intensity 93



List of Tables

3.1	The rule base of FCD	42
3.2	The rule base of FFG	46
4.1	The Fuzzy Rule base of FBCI	79
4.2	The Fuzzy Rule Base of FRC	85



Chapter 1

Introduction

1.1 Motivation

In development of the Internet, the technology of the wavelength division multiplexing (WDM) has impacted the design and realization of the next generation network. From the point-to-point transport technology, the next generation network can be mainly divided into two types: long haul backbone (core) network and the metropolitan area networks (MAN). For the first network type, long haul backbone (core) network, the main challenge is how to keep data in the optical domain as much as possible. For the second network type, how to support the QoS, allocate bandwidth based on fairness, and avoid or solve congestion are the main problems.

Several approaches have proposed to take advantage of optical communication to develop the long haul backbone (core) network. Three of these approaches are the Optical Circuit Switching (OCS), the Optical Packet Switching (OPS), and the Optical Burst Switching (OBS) [1]-[10]. The main attraction of optical switching is that it should enable routing of optical data signals without the need for conversion to electrical signals and, therefore, should be independent of data rate and data

protocol. Also, the three optical switchings could promise for the gradual migration of the switching functions from electronics to optics. While OCS provides bandwidth at a granularity of a wavelength, OPS can offer an almost arbitrary fine granularity, comparable to currently applied electrical packet switching, and OBS lies between them.

Unfortunately, the Optical Circuit Switching (OCS) inevitably suffers from various shortcomings. For example, at the network edge, sophisticated traffic aggregation (or grooming) mechanisms are needed to support applications requiring only sub-wavelength bandwidth cost-efficiently by fully utilizing the optical pipes. If the number of client nodes connected to an optical network increases, the number of wavelengths required to provide a full mesh (i.e., all-to-all connectivity), as well as the corresponding size of the wavelength switches (or cross-connects), may exceed technological limits.

In the Optical Packet Switching (OPS), the data (payload) is sent along with its control (header). Because the natural statistics of the OPS shares the resources, it can efficiently support bursty traffic. Unfortunately, the Optical Packet Switching (OPS) still faces many cost and technological hurdles. More specifically, one major challenge is the current lack of optical random access memory. When the header is being processed, the payload needs to be buffered which requires O/E and E/O conversions along with electronic buffer or the use of fiber delay lines (FDLs). Notice that O/E conversion is used to convert optical signals to electrical form and, on the contrary, E/O conversion is used to convert electrical signals to optical form. Another major challenge is the stringent requirement for synchronization, both between multiple packets arriving at different input ports of an optical switching, and

between a packet header and its payload. It is due to the fact that the processing time at the intermediate nodes varies. There is still one problem in OPS that the size of the payload is usually too small when considering the high channel bandwidth of optical networks thus normally resulting in a relatively high control overhead.

The optical burst switching (OBS) is viewed as an optical switching paradigm to combine the best of optical circuit and packet switching while avoiding their shortcomings [3, 10]-[13]. The OBS has received considerable attention in the past few years, and various solutions have been proposed and analyzed in an attempt to improve its performance. However, in OBS, a key problem is thus to design efficient algorithms for scheduling bursts (or more precisely their bandwidth reservation). An ideal scheduling algorithm could process a control burst fast enough before the burst arrives, and yet find a suitable void interval (or a suitable combination of a FDL and a void interval) for the burst as long as there exists one. Otherwise, a burst may be unnecessarily discarded either that a reservation cannot be completed before the burst arrives or the scheduling algorithm is not smart enough to make the reservation.

The ring network with the natural advantage, such as simple architecture, easily adding or removing nodes, the fault tolerance property, and the needless routing property, is the prevalent topology used in metropolitan area networks (MANs). The resilient packet ring (RPR) is a dual-ring-based optical packet network, shown in Fig.1.1, and has been recently approved as the IEEE 802.17 Standard [17]. The resilient packet ring (RPR) is constructed by several pairs of two unidirectional links between stations. The RPR can provide guaranteed quality of service parameters and support service monitoring including performance management and fault

management [17, 18]. Besides, the RPR has some noticeable properties such as *spatial reuse*, *fair bandwidth allocation*, and *fast network failure recovery* to get rid of deficiencies of conventional high-speed Ethernet and SONET [19, 20]. Therefore, the RPR can not only achieve high bandwidth utilization and fast network failure recovery but also satisfy the requirements of MANs, such as reliability, flexibility, scalability, and large capacity [19, 20, 21]. The RPR is a superior candidate for MANs.

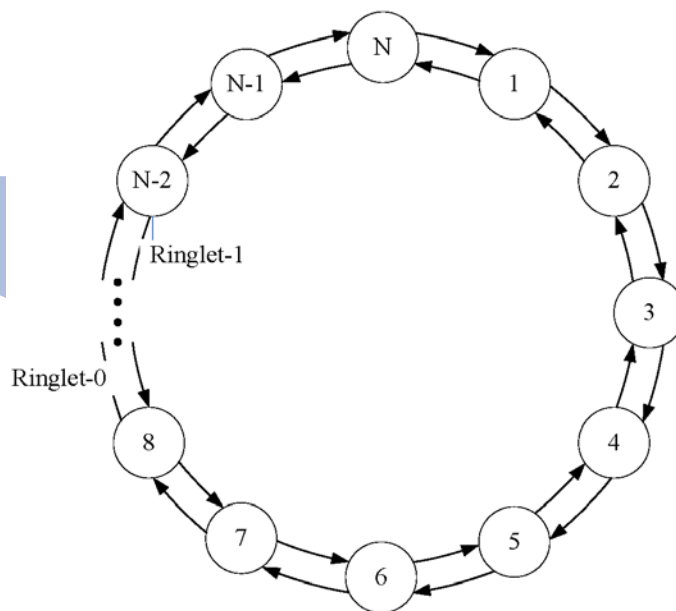


Figure 1.1: RPR structure

The spatial reuse allows a frame to be removed from the ring at its destination so that the bandwidth on next links can be re-used at the same time. Also, the fair bandwidth allocation avoids stations at upstream transmitting too many low-priority frames to cause stations at downstream system congestion. RPR needs

congestion control to enhance the fair bandwidth division in the congestion domain which is defined in the IEEE 802.17 [19, 22]. The congestion control implemented in each station should periodically generate an advertised fairRate to advertise its upstream station for regulating the added fairness eligible (FE) traffic flow defined in IEEE 802.17 [19, 22]. The advertised fairRate should be determined referring to the local fairRate, the received fairRate, and the congestion degree of the station. The local fairRate is generated by a fairness algorithm, and the received fairRate is the advertised fairRate from the downstream station.

Two key factors affect performance of the fair bandwidth allocation: congestion detection and fairness algorithm. If the congestion detection is too rough, it would lower the networks throughput or raise frame loss. The fairness algorithm should consider the most important performance issues of FE traffic flows: stability, fairness, convergence time, and throughput loss caused by the FE traffic flow oscillation. The stability would avoid the oscillation of regulated FE traffic flows, which would cause the throughput loss. If a fairness algorithm referees a ring ingress aggregated with spatial reuse (RIAS) fairness, it has been proved that the algorithm will achieve high system utilization [25]. It is because the RIAS has two key properties. The first property is that an ingress-aggregated (IA) flow fairly shares the bandwidth on each link, relating to other IA flows on the same link, where an IA flow is the aggregate of all flows originating from a given ingress station. The second property is that the maximal spatial reuse subjecting to the first property. Thus, the bandwidth can be reclaimed by IA flows when it is unused. In summary, the RIAS is a max-min fairness with traffic granularity of IA flow. The convergence time is the time interval between the instant of starting the congestion occurrence

and the instant that the amount of arriving specified traffic flow approaches the ideal fairRate which meets the the RIAS fairness. Therefore, a fairness algorithm should achieve not only high stability based on the RIAS fairness but also low convergence time and flow oscillation. There are two conservative modes (CM) [19, 25] and the aggressive mode (AM) [19, 20] fairness algorithms, which have been proposed in IEEE 802.17. Actually, the AM fairness algorithm performs better than the CM fairness algorithm. Unfortunately, the AM suffers from severe oscillations and bandwidth utilization degradation [19, 22]-[24]. It is due to the fact that the AM issues an un-limited fairRate, called FullRate, as its advertised fairRate when the station is released from congestion.

Multiple RPR rings can be bridged together to form a larger network, named bridged-RPR network (BRPR), by a *bridge* which forwards packets from one RPR to another RPR, shown in Fig.1.2. A spatially aware sublayer (SAS), which is a part of the MAC layer, in the bridge is used to decide which ringlet interface the packet should be routed to [17, 26]. Current research on SAS, including the IEEE 802.17b Working Group, is mainly focusing on how to modify this sublayer in order to avoid flooding the entire bridged network when transmitting inter-ring packets [17, 26]-[28].

1.2 Paper Survey

1.2.1 Burst Scheduling in Optical Burst Switching Networks

In OBS networks, there is a strong separation between the control and data planes, which allows for great network manageability and flexibility. In addition, the ingress

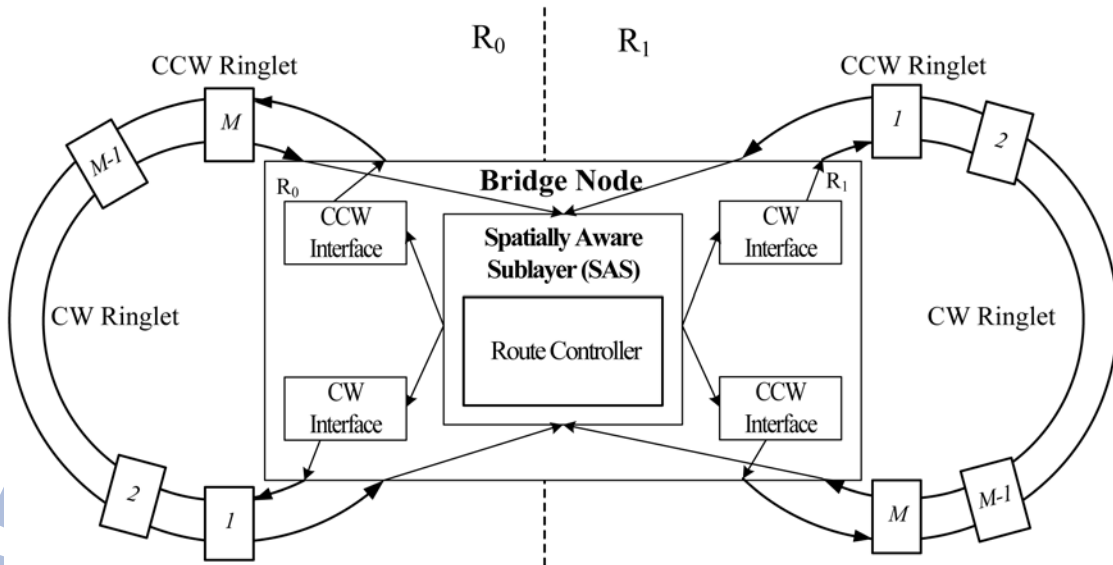


Figure 1.2: BRPR structure

node assembles a number of IP packets, which go to the same egress node, into a data burst (DB). For each DB, there is a control burst (CB) associated with it. In the following, we shall use a burst to indicate that it consists of CB and DB. In OBS networks, DBs are sent on data wavelengths which do not go through optical-to-electronic-to-optical conversions at any intermediate node, whereas their associated CBs are sent on one or more control wavelengths and converted to electronic signals for processing at every intermediate node. This could facilitate efficient electronic control while retaining the advantages of all optical communications such as alleviation of the bottleneck, and support for transparent data rates, coding formats, and protocols [4].

Two kinds signaling protocols, called just-enough-time (JET) and just-in-time (JIT), were proposed in [11, 12, 14], for OBS network. The JIT can be considered a variant of tell-and-wait signaling protocol as it requires each burst transmission

request to be sent to a central scheduler. The scheduler then informs each requesting node the exact time to transmit the data burst. Here, the term just-in-time means that by the time a CB arrives at an intermediate node, the switching fabric has already been configured. In the JIT protocol, the ingress node of a burst sends a CB to the next intermediate to make wavelength reservation. Such packet performs resource reservation at each node belonging to the burst path. As soon as the control packet arrives at a node, wavelength reservation and switch configuration are performed, and then the packet is forwarded to the next node. Since burst transmission needs to happen only when resources have been configured along the entire path, an initial transmission delay is necessary at the ingress node.

The JET is a reserve-a-fixed duration (RFD) scheme that reserves resources exactly for the transmission time of the burst. Particularly, the JET protocol is considered most effective, a control packet for each burst payload is first transmitted out-of-band, allowing each switch to perform just-in-time configuration before the burst arrives. In JET protocol, the CB is first transmitted to the next node to reserve the bandwidth and then the DB is sent after an offset time. The duration of offset time is dependent on the number of intermediate OBS nodes in its routing path, and the routing path is determined by the ingress node, which would be the shortest path from source to destination. Two new prioritized signaling protocols, called prioritized JET (PJET) [15] and preemptive prioritized JET (PPJET) [16], were proposed to provide quality of service. The PJET introduces a significant amount of delay, called extra-offset-time, to let the high-priority traffic isolate from the low-priority traffic by that the higher priority burst has the longer extra-offset-time. The PPJET serves different traffic classes on the basis of a strict priority order.

It makes high-priority bursts preempt low-priority bursts, even if the low-priority burst was scheduled, and does not need the excessive delay.

There are several burst scheduling schemes proposed and applied with PJET, which are the Horizon [29], the latest available unused channel with void filling (LAUC-VF) [48], and the efficient burst scheduling algorithms using geometric techniques [31]. The Horizon chooses a wavelength whose latest available time is the most close to the arrival time of the new burst. The LAUC-VF first chooses, among all the proper voids between two scheduled bursts, the one whose latest available time is the most close to the arrival time of the new burst and whose length is larger than the new burst length. If there is no available void, then the LAUC-VF works the same as the Horizon. The scheduling algorithms in [31] are similar to the Horizon and LAUC-VF except they use an efficient data structure to reduce the computing complexity. Another new prioritized signaling protocol, called preemptive prioritized JET (PPJET), was proposed in [16]. It serves different traffic classes on the basis of a strict priority order. A new scheduling scheme using PPJET, called preemptive latestavailable unused channel with void filling (PLAUC-VF), can provide better service for high-priority traffic by dropping reservations belonging to lower-priority traffic. It is similar to the LAUC-VF except it could let the high-priority burst preempt the low-priority burst.

1.2.2 Traffic control in Resilient Packet Ring

Since the Resilient Packet Ring (RPR), unlike legacy technologies, supports destination packet removal so that a packet will not traverse all ring nodes and spatial reuse can be achieved. However, allowing spatial reuse introduces a challenge to

ensure fairness among different nodes competing for ring bandwidth [25]. The RPR defines two fairness algorithms, conservative mode (CM) [19, 25] and the aggressive mode (AM) [19, 20] fairness algorithms, that specify how upstream traffic should be throttled according to downstream measurements, named an advertised fairRate. The upstream nodes would appropriately configure their rate limiters to throttle the rate of injected traffic to its fair rate. Unfortunately, both the two RPR fairness algorithms have a number of important performance limitations. First, they are prone to severe and permanent oscillations in the range of the entire link bandwidth in simple unbalanced traffic network environment, in which all flows do not demand the same bandwidth. Second, they could not fully achieve spatial reuse and fairness. Third, they must take much time to stabilize all flows [25, 32]. The operations of the two algorithms are described as follows.

In AM, the congested station also calculates and advertises a fairRate estimate periodically without waiting to evaluate the received traffic which is regulated by the previously transmitted advertised fairRate. Also, the calculation of the fairRate is based solely on preset parameters and the station's added rate which is the traffic added in ringlet. The frequent advertisement of new fairRate brings a more "aggressive" algorithm, thus more quickly attempts to adapt to changing traffic conditions. However, the faster response as compared to the conservative mode induces the risk of instabilities that flows oscillate permanently, when rate adjustments are made faster than the system is able to respond. In CM, the congested station transmits an advertised fair rate to upstream, and then waits to see the change in traffic from upstream stations. If the observed effect is not the fair division of rates, then the congested station calculates a new fair rate estimate again, and distributes it to

upstream.

Several fairness algorithms were proposed to solve this problem and some of them were designed based on the RIAS fairness [25, 32]-[37]. The distributed virtual-time scheduling (DVSR) [25] is proposed by Gambiroza et al. and it mainly computes a simple lower bound of temporally and spatially aggregated virtual time using per-ingress counter of packet arrival. The aggregated information propagates along the ring to let each station know the traffic condition of downstream stations. Therefore, each node is capable of limiting its output rate to satisfy RIAS fairness. Unfortunately, it is at the expense of a high computational complexity $O(N \log N)$, where N is the number of stations in the ring.

Alharbi and Ansari proposed a distributed bandwidth allocation (DBA) fairness algorithm with a low computational complexity $O(1)$ [32, 33]. The DBA measures the arrival rate so as to calculate the effective number of ingress-aggregated (IA) flows, where IA flow represents the aggregate of all flows originating from a given ingress station, transiting over the local station. By a recursive method, DBA uses the effective number of IA flows and the remaining bandwidth to obtain the advertised fairRate. After some rounds of recursion, an advertised fairRate which satisfies RIAS fairness can be obtained. However, whenever the effect of propagation delay is severe, the DBA would not be a stable local fairRate algorithm. It is because the local fairRate generated by DBA is related only with the amount of the arriving transit FE traffic flows measured during a short frame time. This short-term amount is easily influenced by the effect of the propagation delay, which starts from a station sending its advertised fairRate and ends the corresponding transit traffic flows arriving the station. If the propagation delay is large, the short-term

arriving transit FE traffic flows would be largely varied and make the generation of local fairRate unstable (incorrect).

Moreover, Yilmaz and Ansari investigated weighted fairness in IEEE802.17 but found one unexpected phenomenon [35]. When a station with a larger weight becomes a head of congestion domain, it leads to an undesirable result of bandwidth allocation and oscillation. However, after modifying a little in original fairness algorithm of AM, it can work correctly under weighted fairness.

1.2.3 Traffic control in Bridged Resilient Packet Rings

Settawong and Tantertdid proposed an enhancement by using a topology discovery and spanning tree algorithm [27]. The algorithm can manage traffic between rings more efficiently and can remove the need for flooding. The shortest path route controller (SPRC) was widely considered for metro rings [38]-[40] as it can maximize the spatial reuse and thus the achievable packet throughput for uniform traffic. However, as traffic load increases, incoming call requests could pile up at a node before being processed, and these would result in a potential bottleneck in network performance [40]. Also, Heiden et. al. analyzed the capacity of bidirectional optical packet ring networks, such as RPR, which employs the SPRC for multicast hotspot traffic [41]. They found that when the multicast traffic originating at the hotspot exceeds a critical threshold, the SPRC leads to a significant capacity reduction.

Intuitively, the route selection would be closely related with the congestion degree of the ringlet so as to follow the load balancing principle. Generally, RPR uses a queue length threshold to detect the congestion and a nodes adding rate limitation to avoid the network congestion [17]. Therefore, an intuitive queue-length

threshold route controller (QTRC) would be better than the SPRC. However, the correlation function between the congestion degree and these variables is nonlinear and complicated.

1.3 Dissertation Organization

In this dissertation, we first discuss the prioritized burst scheduling in OBS network, then the effective local fairRate generating in RPR network, and finally, the inter-ring route control.

In Chapter 2, we propose a new scheduling scheme, named priority burst scheduling with FDL assignment (PBS-FA) [42]. The PBS-FA scheme considers the preemptions because the high priority burst is more important than the low-priority one and the shorter burst is more easily to be rescheduled into the void. Therefore, it allows high-priority bursts to preempt low-priority ones and longer high-priority bursts to replace shorter ones. Meanwhile, FDL assignment is used when scheduling these bursts.

In Chapter 3, we propose an effective local fairRate generator based on fuzzy logic theory [43, 44] and moving average technique [45]. The effective local fairRate generator, named fuzzy local fairRate generator (FLAG), can meet the RIAS fairness and reflect timely the congestion status of station. The FLAG is sophisticatedly configured into three functional blocks: adaptive fairRate calculator (AFC), fuzzy congestion detector (FCD), and fuzzy fairRate generator (FFG). It first reproduces a local fairRate to meet the RIAS fairness and diminish the effect of propagation delay by AFC. Also, the FLAG evaluates the congestion degree of a station, denoting the forwarding capacity of added FE traffic flows at the station and buffering

capacity of the STQ, by FCD. Finally, the FLAG generates a precise local fairRate by FFG. The FFG finely adjusts the pre-produced local fairRate from AFC according to the congestion degree of the station from FCD, using fuzzy logics based upon domain knowledge.

In Chapter 4, we propose intelligent inter-ring route control for bridged resilient packet rings in this paper [47]. Either CW or CCW ringlet at bridge will be properly chosen for an incoming new call request from one RPR to the other RPR. The selection is based on the load balancing principle which is in the sense that the selected ringlet would be with lower congestion degree and higher service rate [46]. An intelligent inter-ring route controller (IIRC) is designed to contain a fuzzy bridge-node congestion indicator (FBCI) to intelligently detect the congestion degree of bridge, and a pipeline recurrent neural networks (PRNN) downstream-node fairness predictor (PDFP) to effectively predict the mean received fairRate. Besides, the IIRC consists of a fuzzy router controller (FRC) to determine preference values of route of CW and CCW ringlets according to the congestion indication provided by FBCI, the predicted mean received fairRate provided by PDFP, the number of hops to destination, and the service rate of the bridge. A ringlet with a larger route preference value would be more proper to be selected.

Finally, some concluding remarks and future research topics are addressed in Chapter 5.

Chapter 2

Priority Burst Scheduling with FDL Assignment for Optical Burst Switching Networks

2.1 Introduction

Optical burst switching (OBS) is a new data transmission/switching method to realize IP over WDM. It strikes a balance between optical circuit switching and optical packet switching [13, 29]. In OBS networks, the ingress node assembles a number of IP packets, which go to the same egress node, into a data burst (DB). For each DB, there is a control burst (CB) associated with it. A signaling protocol, called just-enough-time (JET), was proposed [12], where the CB is first transmitted to the next node to reserve the bandwidth and then the DB is sent after an offset time. The duration of the offset time is dependent on the number of intermediate OBS nodes in the routing path, and the routing path is a shortest path which is determined by the ingress node. A prioritized signaling protocol, called prioritized JET (PJET) [15], was proposed to decrease the dropping probability of high-priority bursts. The PJET introduces a longer offset time for the high-priority burst to make the reservation earlier than the low-priority one.

There are several burst scheduling schemes proposed and applied with PJET, which are the Horizon [29], the latest available unused channel with void filling (LAUC-VF) [48], and the efficient burst scheduling algorithms using geometric techniques [31]. The Horizon chooses a wavelength whose latest available time is the most close to the arrival time of the new burst. The LAUC-VF first chooses, among all the proper voids between two scheduled bursts, the one whose latest available time is the most close to the arrival time of the new burst and whose length is larger than that of the new burst. If there is no available void, then the LAUC-VF works the same as the Horizon. The scheduling algorithms in [31] are similar to the Horizon and LAUC-VF except they use an efficient data structure to reduce the computing complexity. Another new prioritized signaling protocol, called preemptive prioritized JET (PPJET), was proposed in [16]. It serves different traffic classes on the basis of a strict priority order. A new scheduling scheme using PPJET, called preemptive latest-available unused channel with void filling (PLAUC-VF), can provide better service for high-priority traffic by dropping reservations belonging to lower-priority traffic [16]. It is similar to the LAUC-VF except it could let the high-priority burst preempt the low-priority burst.

This chapter proposes a new scheduling scheme, named priority burst scheduling with FDL assignment (PBS-FA). The PBS-FA scheme considers the preemptions because the high-priority burst is more important than the low-priority one and the shorter burst is more easily to be rescheduled into the void. Therefore, it allows high-priority bursts to preempt low-priority ones and longer high-priority bursts to replace shorter ones. Meanwhile, FDL assignment is used when scheduling these bursts.

The rest of this chapter is organized as follows. In Section 2.2, the architecture of intermediate OBS node is introduced. Section 2.2 presents the proposed scheduling PBS-FA and Section 2.4 presents simulation results. Finally, some concluding remarks are given in Section 2.5.

2.2 Architecture of Intermediate OBS Node

Suppose that the intermediate OBS node is an $N \times N$ router connecting N incoming and N outgoing fibers. Each fiber contains $W + 1$ wavelengths; one is the *control channel* transmitting CBs, and others are *data channels* transmitting DBs. The architecture of an intermediate OBS routing node, shown in Fig. 2.1, consists of receiver equipment (RX), transmitter equipment (TX), N input FDLs, N wavelength converters (WC), an $NW \times NW$ non-blocking optical switching matrix (OSM), a control buffer (CBF), and a central processor embedded with the PBS-FA scheduler [48].

The RX receives a DB from a data channel and forwards it to the input FDLs if the DB could be scheduled into an available channel; otherwise, the RX drops it. Also, the RX receives CBs from the control channel and forwards them to the central processor. The TX transmits DBs (CBs) into the data (control) channel. When a CB arrives the central processor, the PBS-FA scheduler properly determines a suitable scheduling result (for the associated DB) which will be sent to the RX, input FDLs, WC, and OSM. Also, the central processor generates a new CB containing the new scheduling result and sends it to CBF, which will inform the next router to cancel the last reservation and to make a new reservation. The input FDLs consists of a number of FDLs with different units of length. The WC converts the wavelength of

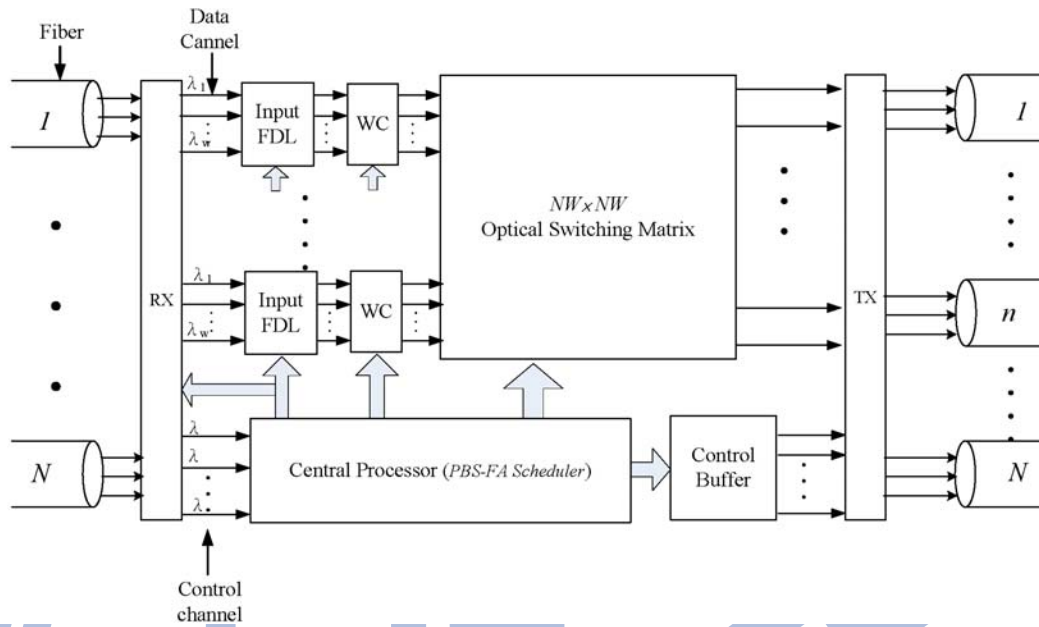


Figure 2.1: Architecture of the OBS node

a DB to the new one which is assigned for it.

The CBF is used to buffer CBs. If a DB is initially needed to be delayed by an FDL, the CBF will buffer its CB for a span whose length is the same as the FDL. If a DB is rescheduled and buffered by an FDL, the CBF buffers the new CB for a period of time. The duration of the time period is the difference between the residual time and the offset time if the residual time is larger than the offset time, where the residual time is defined as the time interval between the current time and the time the DB will be sent out in this reschedule. Otherwise, the duration is zero, denoting the new CB will be sent immediately. In this way, The offset time can be still kept the same as the original one and the router can easily know how many routers the burst still needs to pass through.

2.3 Priority Burst Scheduling with FDL Assignment (PBS-FA)

The priority burst scheduling with FDL assignment (PBS-FA) scheme is shown in the Fig. 2.2. When a CB arrives, the PBS-FA denotes its corresponding DB as \mathbf{b} , empties the set of replaced burst, denoted by \mathbf{B}_R , initializes the number of DB's reassignment, denoted by t , and checks \mathbf{b} 's priority. The \mathbf{B}_R is the set used to collect the preempted bursts. If \mathbf{b} is with low-priority, the PBS-FA finds whether a channel \mathbf{C}_A with a minimal free FDL is available (the free FDLs are used from the shortest to the longest). If \mathbf{C}_A exists, \mathbf{b} will be assigned into \mathbf{C}_A ; otherwise, \mathbf{b} will be dropped. If \mathbf{b} is with high-priority, the PBS-FA first finds whether a channel \mathbf{C}_A with a minimal free FDL is available. If \mathbf{C}_A does exist, \mathbf{b} is assigned into \mathbf{C}_A . Otherwise, the PBS-FA further finds whether a channel \mathbf{C}_L with a minimal free FDL is available for \mathbf{b} , where \mathbf{C}_L is the channel given to a set of the low priority bursts, denoted by \mathbf{B}_L , which block \mathbf{b} and have the shortest burst's sum length. If \mathbf{C}_L exists, the PBS-FA allocates \mathbf{C}_L to \mathbf{b} and removes the bursts in \mathbf{B}_L from \mathbf{C}_L into \mathbf{B}_R . Noticeably, the FDL which has been assigned to a burst will be released when the burst is replaced or dropped. If there is no \mathbf{C}_L , the PBS-FA looks into whether a channel \mathbf{C}_H with the minimal free FDL is available, where \mathbf{C}_H is the channel given to a set of the bursts, denoted by \mathbf{B}_H , which block \mathbf{b} and have the minimum sum length but the sum length of the high-priority bursts is smaller than \mathbf{b} . If \mathbf{C}_H exists, the PBS-FA removes the bursts in \mathbf{B}_H from \mathbf{C}_H into \mathbf{B}_R , and assigns \mathbf{b} into \mathbf{C}_H ; otherwise \mathbf{b} will be dropped. Next if \mathbf{B}_R is not empty, the PBS-FA will perform the rescheduling and check whether the number $t+1$ is smaller than T and free FDLs are available, where T is used to limit the reassignment times between

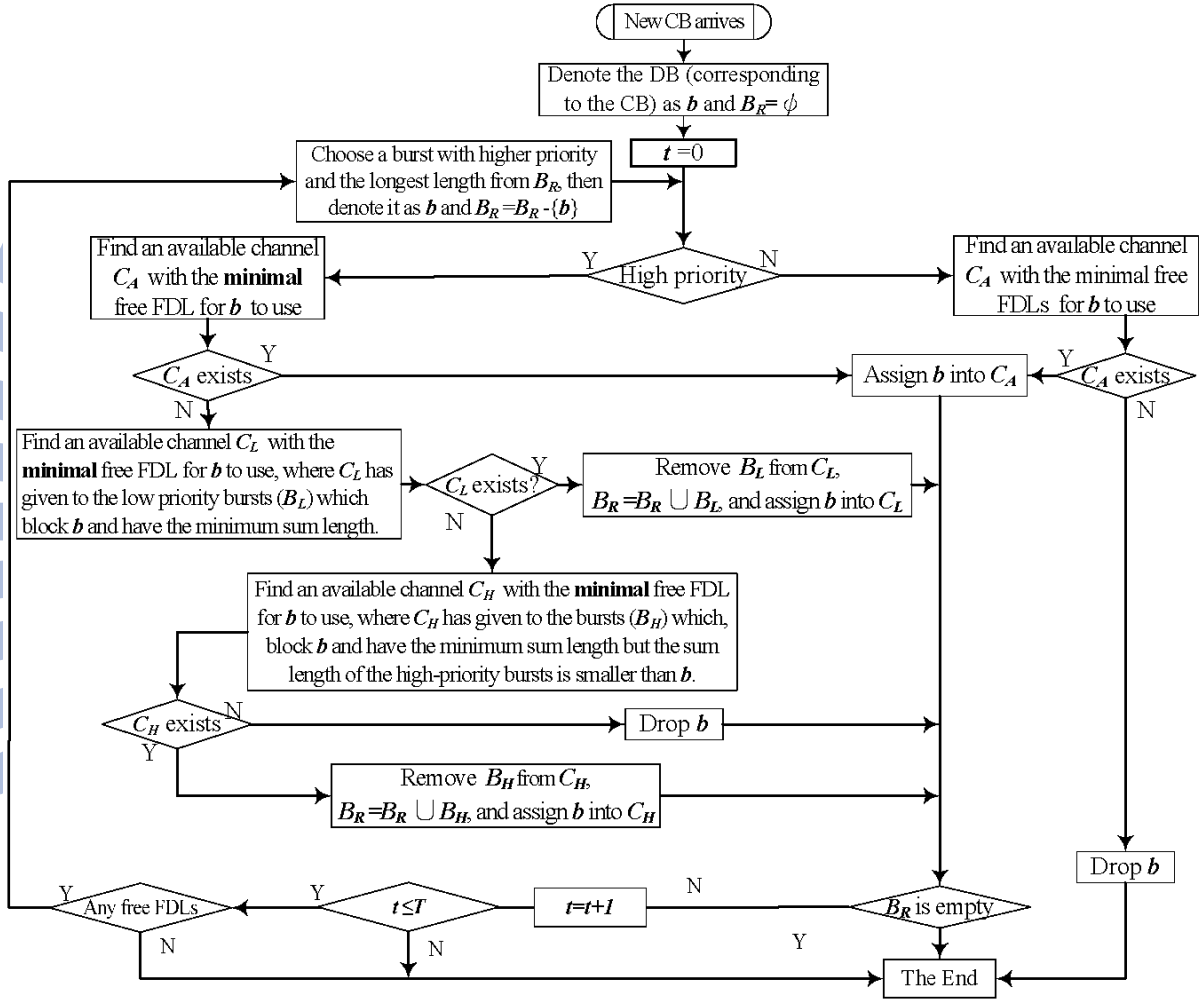


Figure 2.2: The flowchart of the PBS-FA scheme

two successive new CBs and its value is determined according to the processing time of the PBS-FA scheme. If they do exist, the PBS-FA reschedules the bursts in \mathbf{B}_R on the order of the burst priority and length; the higher-priority burst with the longest length will be rescheduled first. Otherwise, the scheme ends.

We illustrate an example in Fig. 2.3, where the class_0 (class_1) denotes the high-priority (low-priority). In Fig. 2.3 (a), assume that six bursts have been scheduled and the low-priority burst 7 with length is $|t_e - t_s|$ will arrive at the t_s . Since there is no available channel for the burst 7, the burst 7 will be dropped. If the burst 7 is with high-priority as in Fig. 2.3 (b), then the PBS-FA first searches a channel \mathbf{C}_L which has given to a set of the scheduled low-priority bursts \mathbf{B}_L which blocks the burst 7 and has the minimum sum burst length. It is found that \mathbf{C}_L is the channel \mathbf{C}_1 and \mathbf{B}_L consists of the burst 4 with low-priority. Then, the scheme schedules the burst 7 into \mathbf{C}_1 and reschedules the burst 4 with FDL, which is shown in Fig. 2.3 (c). If the scheduled bursts which block the burst 7 are with high-priority, as in Fig. 2.3 (d), the PBS-FA makes the burst 7 to preempt the burst 5 because the burst 5 has the least amount of length. The result is shown in Fig. 2.3 (e).

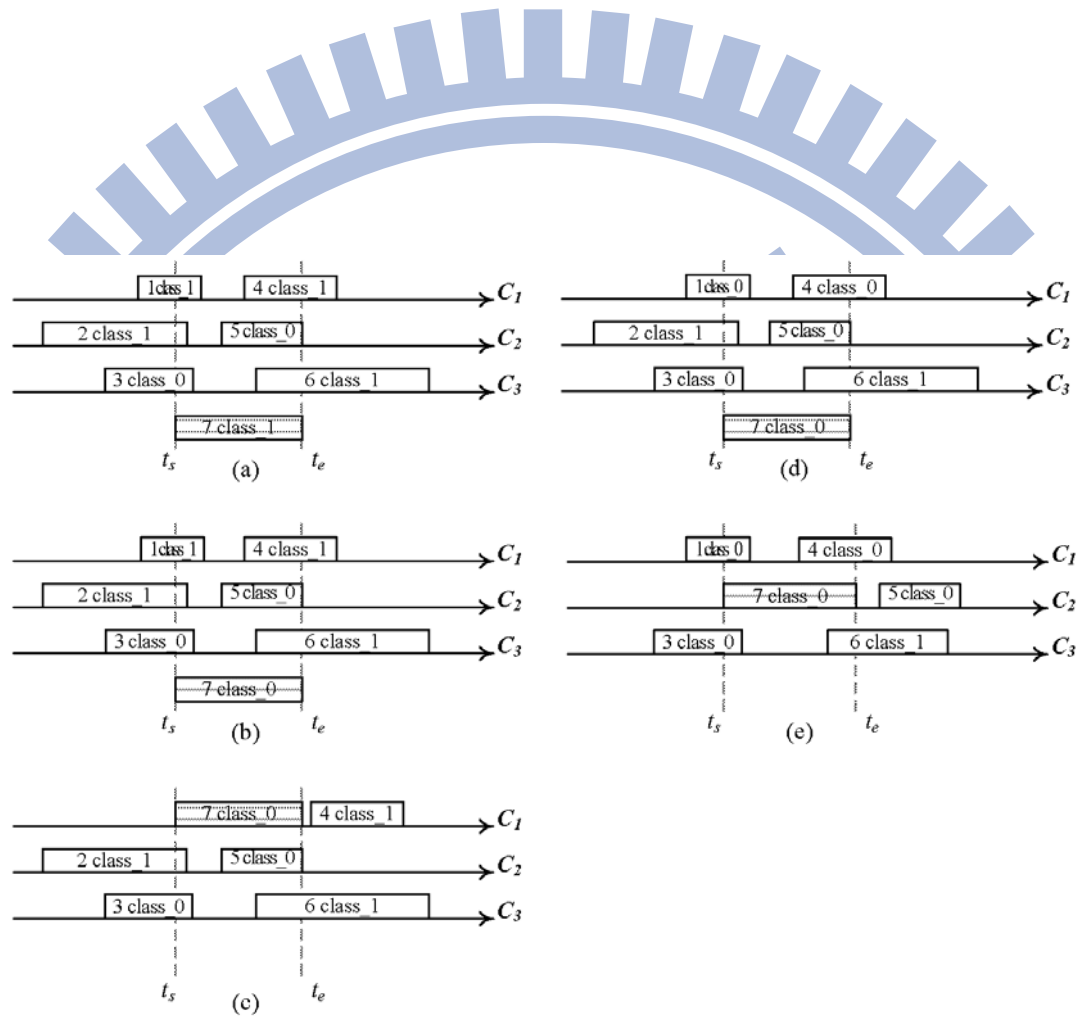


Figure 2.3: The flowchart of the PBS-FA scheme

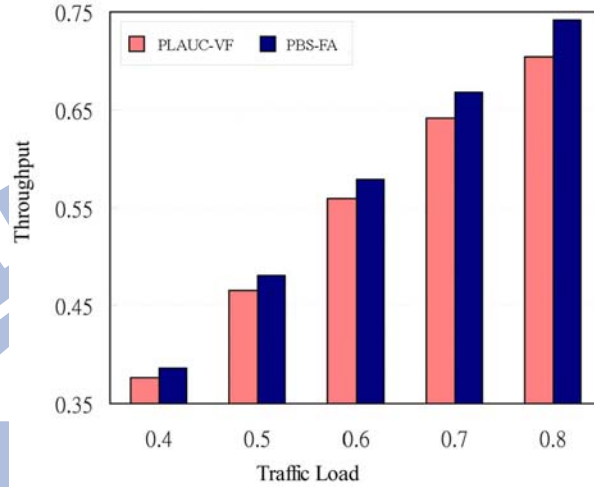


Figure 2.4: The system throughput

2.4 Simulation Results

We compare the PBS-FA scheme with the PLAUC-VF scheme in the performance measures of system throughput and average system dropping probability. Assume two classes of bursts, namely class_0 and class_1, in the simulations, where class_0(1) burst corresponds to the high (low)-priority. An 8×8 OBS routing node is considered and each burst coming from any input fiber goes to a considered output fiber with the $5/16$ probability. Suppose that a fiber contains 9 separate wavelengths, one (eight) for control (data) channel. The transmission speed per wavelength is 2.5 Gbit/s (OC-48). The burst arrival process is in Poisson distribution with a mean which is changed to show various traffic loads, and the burst length is in exponential distribution with a mean 16 KB. Class_0 and class_1 bursts share the total offered load in $9/16$ and $7/16$ percentages, respectively. The FDL length is measured in units of $10\mu s$, and the longest one is $200\mu s$.

Fig. 2.4 shows the system throughputs of the PBS-FA and the PLAUC-VF

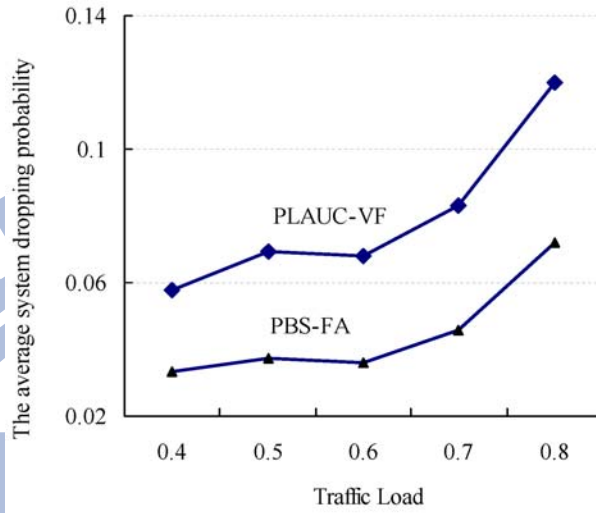


Figure 2.5: The average system dropping probability

schemes. The result reveals that the PBS-FA achieves the throughput higher than the PLAUC-VF by an amount of 3% - 10% at traffic load 0.5 - 0.8. It is because the PBS-FA makes the longer high-priority bursts preempt not only the low-priority bursts but also the shorter high-priority bursts. This results in that the longer high-priority bursts can be successfully scheduled with higher probability and thus the total sum length of the served bursts would be larger, in PBS-FA. Also, the PBS-FA reschedules the preempted bursts into an available channel, while the PLAUC-VF takes no action.

Fig. 2.5 shows the average system dropping probability. It can be seen that the PBS-FA attains a smaller dropping probability than the PLAUC-VF by about 30% to 45% at the traffic load 0.4 to 0.8. The reasons are that the PBS-FA preempts bursts with the shortest total sum length first whenever necessary and reschedules the preempted short bursts with FDL assignment. Noticeably, the short bursts are more easily to be rescheduled, not blocked, whenever the high-priority burst needs

to preempt the low-priority burst. The PLAUC-VF does not choose the bursts with the smallest total length to replace.

2.5 Concluding Remarks

In this chapter we propose a new channel-scheduling scheme called priority burst scheduling with FDL assignment (PBS-FA) with PPJET for OBS networks. The PBS-FA allows high-priority bursts to preempt low-priority ones and longer high-priority ones to replace shorter ones due to that the high-priority is more important than the low-priority and the shorter one is more easily to be scheduled into the void. Also, it reschedules those preempted bursts by using FDL assignment. Simulation results reveals that the PBS-FA improves the system throughput by 3% to 10% and reduces the average system dropping probability by about 30% to 45% at the traffic load 0.4 to 0.8 over the PLAUC-VF.

Chapter 3

FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Ring

3.1 Introduction

The resilient packet ring (RPR) is a ring based network for high-speed metropolitan area networks (MANs) [17]. It is a packet transport layer can provide guaranteed quality of service parameters and support service monitoring including performance management and fault management [17, 18]. Besides, the RPR has some noticeable properties such as *spatial reuse*, *fair bandwidth allocation*, and *fast network failure recovery* to get rid of deficiencies of conventional high-speed Ethernet and SONET [19, 20]. Therefore, the RPR can not only achieve high bandwidth utilization and fast network failure recovery but also satisfy the requirements of MANs, such as reliability, flexibility, scalability, and large capacity [19, 20, 21]. The RPR is a superior candidate for MANs.

The spatial reuse allows a frame to be removed from the ring at its destination so that the bandwidth on next links can be re-used at the same time. Also, the fair bandwidth allocation avoids stations at upstream transmitting too many low-

priority frames to cause stations at downstream system congestion. RPR needs congestion control to enhance the fair bandwidth division in the congestion domain which is defined in the IEEE 802.17 [19, 22]. The congestion control implemented in each station should periodically generate an *advertised fairRate* to advertise its upstream station for regulating the added fairness eligible (FE) traffic flow defined in IEEE 802.17 [19, 22]. The advertised fairRate should be determined referring to the local fairRate, the received fairRate, and the congestion degree of the station. The local fairRate is generated by a fairness algorithm, and the received fairRate is the advertised fairRate from the downstream station.

Two key factors affect performance of the fair bandwidth allocation: congestion detection and fairness algorithm. If the congestion detection is too rough, it would lower the network's throughput or raise frame loss. The fairness algorithm should consider the most important performance issues of FE traffic flows: stability, fairness, convergence time, and throughput loss caused by the FE traffic flow oscillation. The stability would avoid the oscillation of regulated FE traffic flows, which would cause the throughput loss. If a fairness algorithm referees a "ring ingress aggregated with spatial reuse (RIAS)" fairness, it has been proved that the algorithm will achieve high system utilization [25]. It is because the RIAS has two key properties. The first property is that an *ingress-aggregated* (IA) flow fairly shares the bandwidth on each link, relating to other IA flows on the same link, where an IA flow is the aggregate of all flows originating from a given ingress station. The second property is that the maximal spatial reuse subjecting to the first property. Thus, the bandwidth can be reclaimed by IA flows when it is unused. In summary, the RIAS is a max-min fairness with traffic granularity of IA flow. The convergence

time is the time interval between the instant of starting the congestion occurrence and the instant that the amount of arriving specified traffic flow approaches the ideal fairRate which meets the the RIAS fairness. Therefore, a fairness algorithm should achieve not only high stability based on the RIAS fairness but also low convergence time and flow oscillation.

The aggressive mode (AM) fairness algorithm has been proposed in IEEE 802.17. It would suffer from severe oscillations and bandwidth utilization degradation [19, 22, 24, 25, 32]. It is because AM issues a un-limited fairRate, called FullRate, as its advertised fairRate when the station is released from congestion. Several fairness algorithms were proposed to solve this problem and some of them were designed based on the RIAS fairness [24, 25, 32, 33, 35, 34, 36, 37]. Gambiroza et al. proposed a distributed virtual-time scheduling in rings (DVSR) [25]. Unfortunately, it is at the expense of a high computational complexity $O(N \log N)$, where N is the number of stations in the ring. Alharbi and Ansari proposed a distributed bandwidth allocation (DBA) fairness algorithm with a low computational complexity $O(1)$ [24, 32, 33]. However, whenever the effect of propagation delay is severe, the DBA would not be a stable local fairRate algorithm. It is because the local fairRate generated by DBA is related only with the amount of the arriving transit FE traffic flows measured during a short frame time. This short-term amount is easily influenced by the effect of the propagation delay, which starts from a station sending its advertised fairRate and ends the corresponding transit traffic flows arriving the station. If the propagation delay is large, the short-term arriving transit FE traffic flows would be largely varied and makes the generation of local fairRate unstable (incorrect).

Recently, fuzzy logics system, which is a kind of intelligent techniques, has been widely applied to control nonlinear, time-varying, and well-defined systems for that fuzzy logic control can provide effective solutions with small computational complexity. Fuzzy set theory appears to be able to support a robust mathematical framework for dealing with real-world imprecision, and exhibits a soft behavior, which means a greater ability to adapt itself to dynamic, imprecise, and bursty environments [43].

In this chapter, we propose an effective local fairRate generator based on fuzzy logic theory [43] and moving average technique [45]. The effective local fairRate generator, named fuzzy local fairRate generator (FLAG), can meet the RIAS fairness and reflect timely the congestion status of station. The FLAG is sophisticatedly configured into three functional blocks: adaptive fairRate calculator (AFC), fuzzy congestion detector (FCD), and fuzzy fairRate generator (FFG). It first pre-produces a local fairRate to meet the RIAS fairness and diminish the effect of propagation delay by AFC. Also, the FLAG evaluates the congestion degree of a station, denoting the forwarding capacity of added FE traffic flows at the station and buffering capacity of the STQ, by FCD. Finally, the FLAG generates a precise local fairRate by FFG. The FFG finely adjusts the pre-produced local fairRate from AFC according to the congestion degree of the station from FCD, using fuzzy logics based upon domain knowledge. Simulation results show that the FLAG has better performance than AM and DBA in various scenarios in the aspects of lower convergence time, more fairness, and higher throughput. Take a small parking lot scenario with short propagation delay as an instance. The FLAG improves by more than 7 times over AM and by 2 times over DBA, in the convergence time of traffic flows.

The remaining of this chapter is organized as follows. Section 3.2 introduces the RPR system model. The concept of fuzzy logic system (FLS) and the most basic and popular architectures of a fuzzy logic controller are stated in section 3.3. Section 3.4 describes the proposed FLAG. Section 3.5 shows simulation results and discussions. Finally, concluding remarks are given in Section 3.6.

3.2 RPR System Model

Assume that a resilient packet ring (RPR) with N stations, shown in Fig. 3.1, is constructed by two unidirectional, counter-rotating ringlets, named ringlet-0 and ringlet-1. Each station has two pairs of input and output ports to communicate with neighbor stations. Station X (Y) is said to be a upstream (downstream) node of station Y (X) on ringlet-0 or ringlet-1 if the station Y (X) traffic becomes the received traffic of station X (Y) on the referenced ringlet. There are three classes of service for RPR. The classA is used for real-time services and it has subclassA0 for reserved bandwidth and subclassA1 for reclaimable bandwidth. The classB is targeted for near real-time services, and it also has two subclasses: classB-CIR (committed information rate) which requires the bounded delay and guaranteed bandwidth, and classB-EIR (excess information rate) which does not guarantee bandwidth or delay bound. The classC is intended for best effort services and has the lowest priority. Each station only reserves bandwidth for subclassA0, and the remaining bandwidth is provided for other traffic classes according to the order of subclassA1, classB-CIR, classB-EIR, and classC. The latter two low priority traffics are called the *fairness eligible* (FE) traffic and are controlled by a fairness algorithm [17, 18, 19, 20, 21, 22].

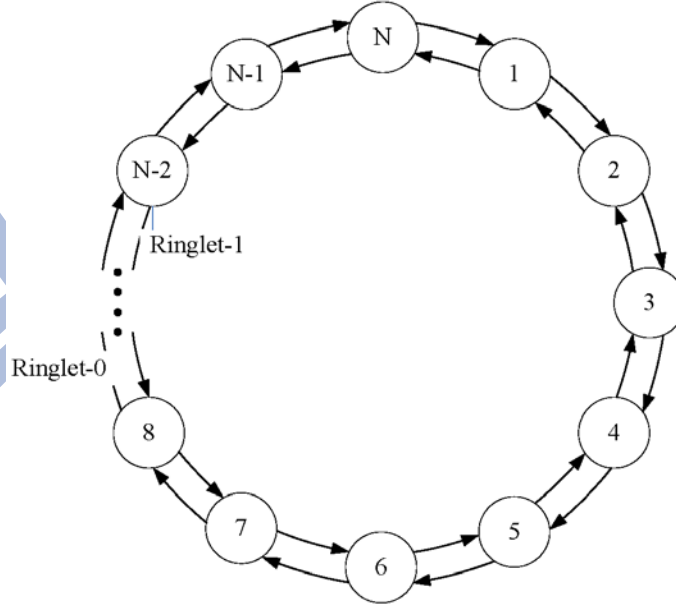


Figure 3.1: Resilient packet ring structure

Fig. 3.2 shows the station structure for ringlet-0 transmission, which contains an ingress queue with ClassA, ClassB, and ClassC queues, a transit queue with primary transit queue (PTQ) and secondary transit queue (STQ), a scheduler, the fuzzy local fairRate generator (FLAG), and a fairness control unit. The ClassX queue, $X = A, B, \text{ or } C$, stores the added classX traffic to the station. The PTQ (STQ) stores the transiting classA and classB-CIR (classB-EIR and classC) frames. The scheduler decides the transmitting order. If the STQ occupancy is less than the *stqHighthreshold* defined in the IEEE802.17 [17], the order is PTQ, ClassA, ClassB, ClassC, and STQ; otherwise, it is PTQ, ClassA, ClassB, STQ, and ClassC. The FLAG generates a local fairRate at every time nT , denoted by $f_l(n)$, where n is a positive integer and T is the duration of an agingInterval. Notice that f_l is also generated per agingInterval in DBA but is generated only when the station is in

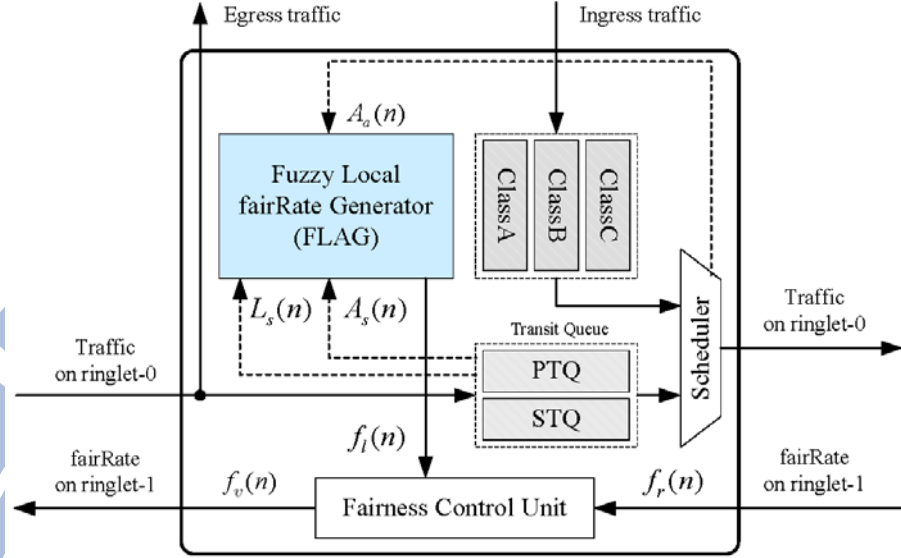


Figure 3.2: RPR station structure

congestion in AM. The fairness control unit usually refers to both $f_l(n)$ and the received fairRate, denoted by $f_r(n)$, to determine an advertised fairRate, denoted by $f_v(n)$, and then sends $f_v(n)$ to upstream stations to regulate traffic flows, at every agingInterval time nT .

The advertised fairRate generated by the fairness control unit are described as follows. The f_v would be set to be f_l if f_r is smaller than f_l and larger than the bandwidth rate of the transit FE traffic flows which will pass through the originally congested station. Otherwise, it is set to be $\min(f_l, f_r)$. Here we also describe the advertised fairRate generated by AM below. When the station is congestion free, the f_v is set to be the FullRate if the f_r is larger than the bandwidth rate of the transit FE traffic flows which will pass through the originally congested station; to be f_r , otherwise. The FullRate is a specially advertised fairRate to indicate that the station does not need to limit its added FE traffic flow. When the station is in

congestion, the f_v is set to be f_l if the f_r is FullRate; to be $\min(f_l, f_r)$, otherwise. Note that the congestion is occurred at a station for AM if the STQ occupancy of the station is larger than the *stqLowthreshold*, defined in IEEE802.17 [17]. Also, the originally congested station is known to the observation station since the message of the advertised fairRate contains a field to record it [17]; the f_l is the added FE traffic flow rate to the network.

3.3 Fuzzy Logic System

Fuzzy logic system is mimicked the behaviors of human brain: fuzzy logic operates on the way the brain deals with vague information [43]. Fuzzy logic system is numerical model-free estimators and dynamical system and, also, it has been shown to have the capability of modelling complex nonlinear processes to arbitrary degrees of accuracy. Fuzzy logic systems employ linguistic *if-then* fuzzy rules as a kind of expert knowledge to formalize insights about the structure of categories founding the real world. Fuzzy logic systems combine the mathematical theory of fuzzy sets with fuzzy rules to produce overall complex nonlinear behavior.

3.3.1 Fuzzy Inference System (FIS)

Fuzzy logic is based on the concepts of linguistic variables and fuzzy sets theory. A *fuzzy set* in a universe of discourse U is characterized by a membership function $\mu(\cdot)$ which takes values in the interval $[0, 1]$. A fuzzy set F is represented as a set of ordered pairs, each made up of a generic element $u \in U$ and its degree of membership $\mu(u)$. A *linguistic variable* x in a universe of discourse U is characterized by $T(x) = \{T_x^1, \dots, T_x^i, \dots, T_x^K\}$ and $M(x) = \{M_x^1(u), \dots, M_x^i(u), \dots, M_x^K(u)\}$, where

$T(x)$ is the fuzzy term set, i.e., the set of linguistic values' names T_x^i the linguistic variable x can take, and $M_x^i(u)$ is a membership function with respect to the term T_x^i . If, for instance, x indicates the temperature, $T(x)$ could be the set as $\{Low, Medium, High\}$, and each element in $T(x)$ is associated with a membership function.

The *fuzzy inference system* (FIS) is a popular computing framework based on the concept of fuzzy logic and fuzzy reasoning. As shown in Fig. 3.3, a fuzzy inference system consists of four fundamental blocks [51]: *fuzzifier*, *fuzzy rule base*, *inference engine*, and *defuzzifier*. The fuzzifier performs a mapping function from the observed value of each input linguistic variable x_i to a fuzzy term set $T(x_i)$ with associated set of membership degree $M(x_i)$, $i = 1, \dots, m$. The fuzzy rule base is a knowledge base characterized by a set of linguistic statements in a form of “*if-then*” rules that describe a fuzzy logic relationship between the m -dim input linguistic variables $\{x_i\}$ and the n -dim output linguistic variables $\{y_j\}$. The inference engine performs an implication function according to the pre-condition of the fuzzy rule with the input linguistic terms. It is a decision-making logic that acquires the input linguistic terms of $T(x_i)$ from the fuzzifier and uses an inference method to obtain the output linguistic terms of $T(y_j)$ [50]. The defuzzifier adopts a defuzzification function to convert $T(y_j)$ into a non-fuzzy (crisp) value that represents the decision y_j . Several implementation ways have been introduced to build a fuzzy inference system as a fuzzy logic controller, such as the *Mamdani fuzzy model*, *Tsukamoto fuzzy model*, and *Sugeno fuzzy model* [49]. Briefly speaking, these fuzzy models (or said implementation ways) differ on the high-level linguistic expression form of the fuzzy rule and the consequent reasoning way. Because the Mamdani fuzzy model is the most basic and popular one, some descriptions about the Mamdani fuzzy model

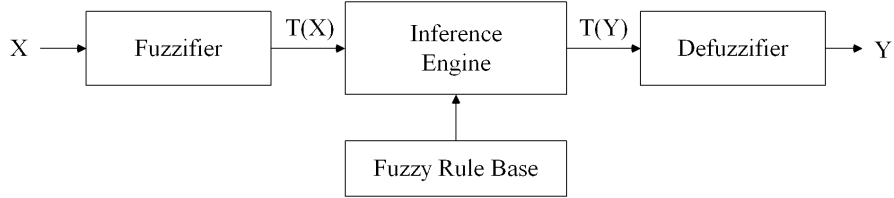


Figure 3.3: The basic structure of fuzzy inference system

are given in the following subsection.

3.3.2 Mamdani Fuzzy Model

The *Mamdani fuzzy model* is a way to implement a fuzzy inference system to serve as a controller. It was proposed as the first attempt to control a system by a set of linguistic control rules obtained from experienced human knowledge. Fig. 3.4 shows an example of Mamdani fuzzy model, where the overall output Z is derived from two linguistic variables X and Y . Here, the fuzzy rule is expressed by

if X is A_i and Y is B_i , then output Z is C_i with $\mu(C_i)$, $i=1$ and 2

where A_i , B_i and C_i are all fuzzy terms, and $\mu(C_i)$ is the membership value on C_i . In the Mamdani model, each input linguistic variable is firstly fuzzified by the membership function $\mu(\cdot)$. Then, the inferred value of the output of each fuzzy rule is determined by a pre-defined inference method. In this example, the *min-max* method is applied. That is, the inferred value of each fuzzy rule is obtained by *min* operator and the inferred value of the same fuzzy term is obtained by *max* operator. Finally, the overall crisp output is derived by a pre-defined defuzzification method. There are diverse defuzzification methods such as: centroid of area (COA), bisector of area (BOA), mean of maximum (MOM), smallest of maximum (SOM),

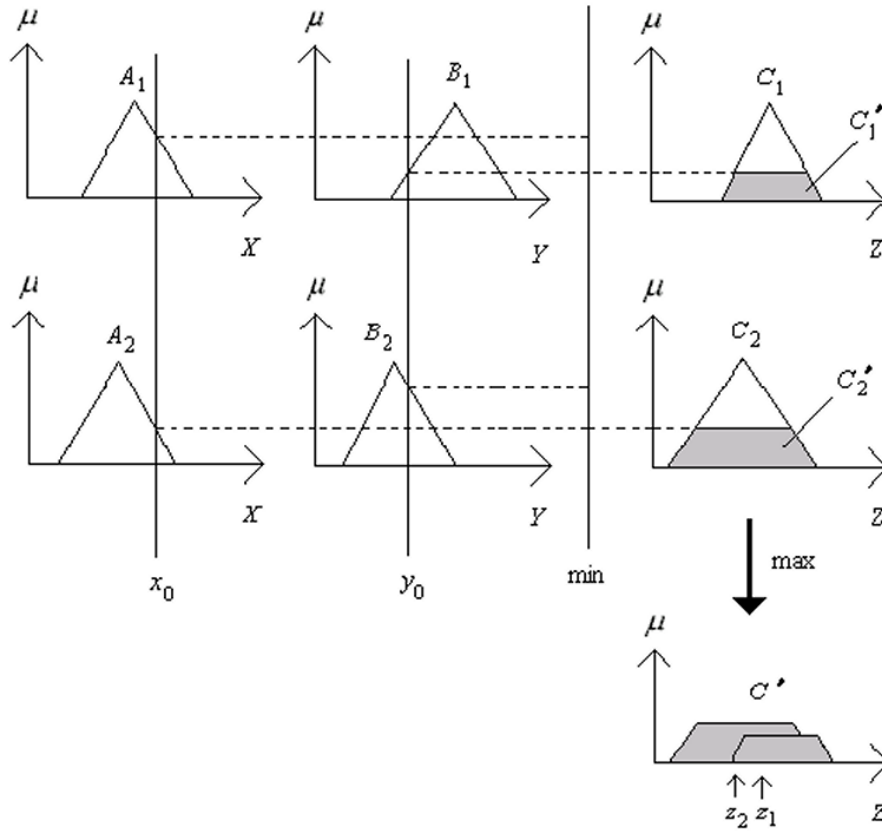


Figure 3.4: An example of Mamdani fuzzy model

and largest of maximum (LOM), among which COA is the most popular one.

Additionally, the membership functions for terms in the term set should be defined with the proper shape and position. In general, a triangular function $f(x; x_0, a_0, a_1)$ or a trapezoidal function $g(x; x_0, x_1, a_0, a_1)$ is chosen as the membership function because of the advantage of simple computational complexity. This feature makes these functions are suitable for real-time application [50]. As shown

in Fig. 3.5, $f(x; x_0, a_0, a_1)$ and $g(x; x_0, x_1, a_0, a_1)$ are given by

$$f(x; x_0, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1 & \text{for } x_0 - a_0 < x \leq x_0 \\ \frac{x_0-x}{a_1} + 1 & \text{for } x_0 < x \leq x_0 + a_1 \\ 0 & \text{otherwise,} \end{cases} \quad (3.1)$$

$$g(x; x_0, x_1, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1 & \text{for } x_0 - a_0 < x \leq x_0 \\ 1 & \text{for } x_0 < x \leq x_1 \\ \frac{x_1-x}{a_1} & \text{for } x_1 < x \leq x_1 + a_1 \\ 0 & \text{otherwise,} \end{cases} \quad (3.2)$$

where x_0 in $f(\cdot)$ is the center of the triangular function; x_0 (x_1) in $g(\cdot)$ is the left (right) edge of the trapezoidal function; and a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

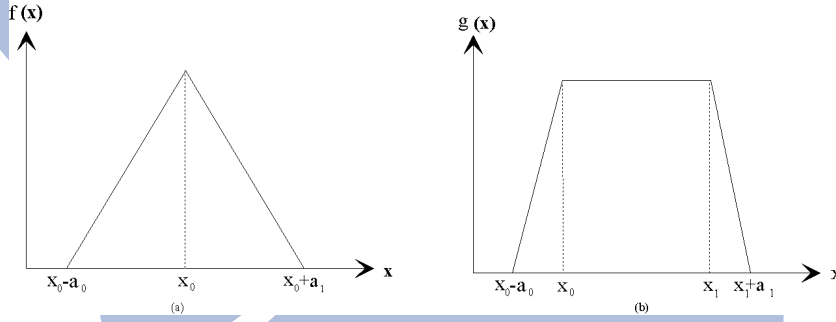


Figure 3.5: Definitions for functions $f(\cdot)$ and $g(\cdot)$

3.4 Fuzzy Local FairRate Generator

The proposed fuzzy local fairRate generator (FLAG), shown in Fig. 3.6, is composed of an adaptive fairRate calculator (AFC), a fuzzy congestion detection (FCD), and a fuzzy fairRate generator (FFG). During the n th agingInterval which is from time $(n - 1)T$ to time nT , the FLAG determines $f_l(n)$ by referring to the arriving FE traffic flows to STQ, denoted as $A_s(n)$, the added FE traffic flow to the network,

denoted as $A_a(n)$, and STQ occupancy, denoted as $L_s(n)$. The AFC pre-generates a local fairRate, called p -fairRate and denoted by $f_p(n)$, which satisfies the RIAS fairness. Its design imitates the DBA's generation of local fairRate, but it would overcome the unstable (incorrect) local fairRate generation by DBA when the propagation delay is significant. Instead of using the short-term arriving transit FE traffic flows, it calculates a proper average of the arriving transit FE traffic flows by *moving average technique* to mitigate the effect of the propagation delay. The FCD appraises the congestion status of station using fuzzy logics. Its design can softly detect the congestion degree of the station in each agingInterval n , denoted by $D_c(n)$, considering not only the STQ occupancy but also the amount of the arriving transit FE traffic flows at the queue. The latter term denotes the change rate of the STQ occupancy which would play an important role in the congestion detection. Finally, the FFG generates a precise local fairRate by fine-tuning the p -fairRate from AFC, referring to the congestion degree from FCD, and further using domain knowledge designed by fuzzy logics. The FLAG would avoid serious regulating FE traffic flows to decrease the throughput or excessive relaxing the traffic flows to increase the frame losses.

3.4.1 Adaptive fairRate Calculator (AFC)

The adaptive fairRate calculator (AFC) adopts the moving average technique [45] on the short-term arriving FE traffic flows, trying to mitigate the effect of propagation delay on the generation of local fairRate by the DBA [24, 32]. During the n -th agingInterval, the AFC first takes the *moving average* of arriving transit

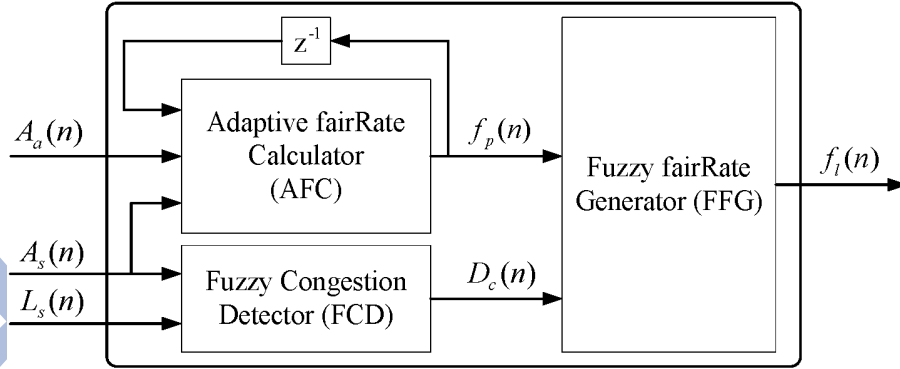


Figure 3.6: Functional blocks of FLAG

FE traffic flows to STQ, $A_s(n)$. Denote the average by $\tilde{A}_s(n)$ and give it by

$$\tilde{A}_s(n) = \sum_{i=n-k+1}^n A_s(i)/k, \quad (3.3)$$

where k is the size of observation window. The k is the sum of two kinds of the data frame trip time: one is the time from the furthest source to this observation station, and the other is the time from this station to originally congested station. It is because the FE traffic flow of a station in this interval would be regulated by an advertised fairRate which is sent out from one of the stations in the interval. The $\tilde{A}_s(n)$ will not vary too much and become more stable.

Then the AFC computes the effective number of IA flows during the n -th agingInterval, denoted by $M(n)$, which is obtained by

$$M(n) = \frac{\tilde{A}_s(n) + A_a(n)}{f_p(n-1)}. \quad (3.4)$$

The AFC fairly allocates the remaining bandwidth to these effective IA flows, which would be $\frac{1}{M(n)}(C - (A_s(n) + A_a(n)))$. Finally, the AFC calculates the $f_p(n)$ by adding up the previous p -fairRate, $f_p(n-1)$, and the fairly shared bandwidth. The

$f_p(n)$ is given by

$$f_p(n) = \min \left\{ C, f_p(n-1) + \frac{1}{M(n)} [C - (A_s(n) + A_a(n))] \right\}, \quad (3.5)$$

where C is the unreserved bandwidth for FE traffic flows per agingInterval used to denote the upper bound of the local fairRate.

3.4.2 Fuzzy Congestion Detector (FCD)

The FCD refers not only the occupancy of STQ, $L_s(n)$, as defined in the IEEE802.17 [17], but also the arriving FE traffic flows to STQ, $A_s(n)$, to determine the congestion degree, $D_c(n)$. The $A_s(n)$ can be viewed as the change rate of STQ, which is also an important variable in the detection of congestion degree. We define the term set for $L_s(n)$ as $T(L_s(n)) = \{\text{Short } (S), \text{Long } (L)\}$; for $A_s(n)$ as $T(A_s(n)) = \{\text{Low } (L), \text{Medium } (M), \text{High } (H)\}$; for $D_c(n)$ as $T(D_c(n)) = \{\text{Very Low } (VL), \text{Low } (L), \text{Medium } (M), \text{High } (H), \text{Very High } (VH)\}$.

The corresponding membership functions of S and L in $T(L_s(n))$ are denoted by

$$\mu_S(L_s(n)) = g(L_s(n); 0, 0.125Q, 0, 0.25Q), \quad (3.6)$$

$$\mu_L(L_s(n)) = g(L_s(n); 0.35Q, Q, 0.25Q, 0), \quad (3.7)$$

where Q is the size of STQ. As defined in IEEE 802.17 [17] standard, we take 0.125 of the STQ size as the *stqLowthreshold* to judge the light congestion degree, and 0.25 of the STQ size as the *stqHighthreshold* to judge the heavy congestion degree. The corresponding membership functions of L , M , and H in $T(A_s(n))$ are denoted by

$$\mu_L(A_s(n)) = g(L_s(n); 0, 0.125C, 0, 0.375C), \quad (3.8)$$

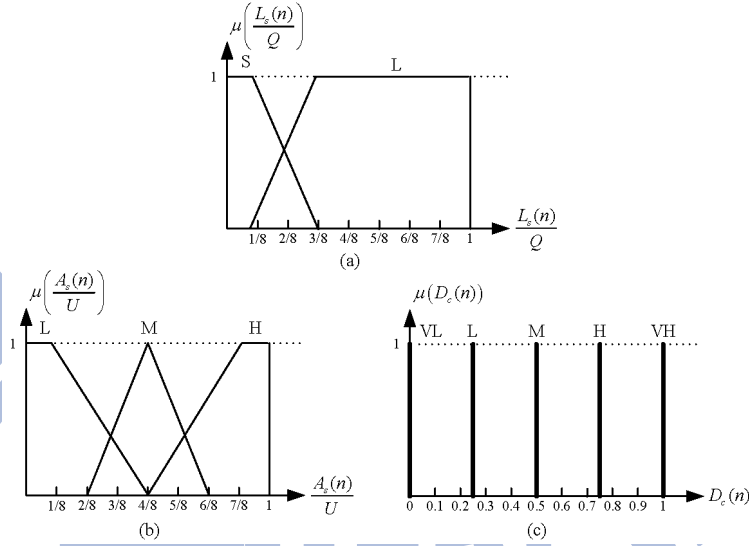


Figure 3.7: The membership functions of the term set (a) $T(L_s(n))$ (b) $T(A_s(n))$ (c) $T(D_c(n))$

$$\mu_M(A_s(n)) = f(A_s(n); 0.5C, 0.25C, 0.25C), \quad (3.9)$$

$$\mu_H(A_s(n)) = g(A_s(n); 0.875C, C, 0.375C, 0). \quad (3.10)$$

For the reason of simplicity in computation of defuzzification, the corresponding membership functions of VL , L , M , H , and VH in $T(D_c(n))$ are defined as

$$\mu_{VL}(D_c(n)) = f(D_c(n); 0, 0, 0), \quad (3.11)$$

$$\mu_L(D_c(n)) = f(D_c(n); 0.25, 0, 0), \quad (3.12)$$

$$\mu_M(D_c(n)) = f(D_c(n); 0.5, 0, 0), \quad (3.13)$$

$$\mu_H(D_c(n)) = f(D_c(n); 0.75, 0, 0), \quad (3.14)$$

$$\mu_{VH}(D_c(n)) = f(D_c(n); 1, 0, 0). \quad (3.15)$$

There are 6 fuzzy rules for FCD. As shown in Table 3.1, the order of significance of the input linguistic variables is $L_s(n)$ then $A_s(n)$. The station with high occupancy

Table 3.1: The rule base of FCD

Rule	$L_s(n)$	$A_s(n)$	$D_c(n)$	Rule	$L_s(n)$	$A_s(n)$	$D_c(n)$
1	S	L	VL	4	L	L	M
2	S	M	VL	5	L	M	H
3	S	H	L	6	L	H	VH

of STQ would be in high congestion degree, and it would be in higher (medium) congestion degree if the arriving FE traffic flows to STQ is also high (low).

The fuzzy congestion detector adopts the *max-min* inference method for inference engine because it is suitable for real-time operation. To explain max-min inference method, we take rule 1 and rule 2, which have the same control action " $D_c(n)$ is VL ", as an example. Applying the "min" operator, we obtain the membership function values of the control action " $D_c(n)$ is VL " of rule 1 and rule 2, denoted by $m_1(n)$ and $m_2(n)$, respectively, by

$$m_1(n) = \min\{\mu_S(L_s(n)), \mu_L(A_s(n))\}, \quad (3.16)$$

$$m_2(n) = \min\{\mu_S(L_s(n)), \mu_M(A_s(n))\}. \quad (3.17)$$

Subsequently, applying the "max" operator yields the overall membership function value of the control action " $D_c(n)$ is VL ", denoted by $w_{VL}(n)$, by

$$w_{VL}(n) = \max\{m_1(n), m_2(n)\}. \quad (3.18)$$

The fuzzy inference results of the output indication L , M , H , and VH , denoted by $w_L(n)$, $w_M(n)$, $w_H(n)$, and $w_{VH}(n)$, respectively, can be obtained by the same way. Finally, the fuzzy inference results are to be defuzzified to become usable values. The defuzzification method adopted is the center of area defuzzification method,

and a crisp value of the congestion degree $D_c(n)$, denoted by z_0 , can be obtained by

$$z_0 = \frac{0.0 \times w_{VL}(n) + 0.25 \times w_L(n) + 0.5 \times w_M(n) + 0.75 \times w_H(n) + 1.0 \times w_{VH}(n)}{w_{VL}(n) + w_L(n) + w_M(n) + w_H(n) + w_{VH}(n)}. \quad (3.19)$$

3.4.3 Fuzzy fairRate Generator (FFG)

The FFG refers to the p -fairRate, $f_p(n)$, and the congestion degree, $D_c(n)$, as the input variables to generate a proper and robust local fairRate, $f_l(n)$. The local fairRate $f_l(n)$ affects both the fairness performance and the bandwidth utilization. Define the term set with six terms for $f_p(n)$ as $T(f_p(n)) = \{\text{Extremely Low (EL), Pretty Low (PL), Slightly Low (SL), Slightly High (SH), Pretty High (PH), Extremely High (EH)}\}$; the term set with three terms for $D_c(n)$ as $T(D_c(n)) = \{\text{Low (L), Medium (M), High (H)}\}$; and the term set with eleven terms for $f_l(n)$ as $T(f_l(n)) = \{\text{Extremely Low (EL), Very Low (VL), Pretty Low (PL), Low (L), Slightly Low (SL), Medium (M), Slightly High (SH), High (H), Pretty High (PH), Very High (VH), Extremely High (EH)}\}$. Note that the number of the terms in $T(f_l(n))$ would be larger than that of $T(f_p(n))$ for better performance. The membership functions for terms EL, PL, SL, SH, PH , and EH in $T(f_p(n))$ are defined as

$$\mu_{EL}(f_p(n)) = f(f_p(n); 0, 0, 0.3C), \quad (3.20)$$

$$\mu_{PL}(f_p(n)) = f(f_p(n); 0.2C, 0.2C, 0.2C), \quad (3.21)$$

$$\mu_{SL}(f_p(n)) = f(f_p(n); 0.4C, 0.2C, 0.2C), \quad (3.22)$$

$$\mu_{SH}(f_p(n)) = f(f_p(n); 0.6C, 0.2C, 0.2C), \quad (3.23)$$

$$\mu_{PH}(f_p(n)) = f(f_p(n); 0.8C, 0.2C, 0.2C), \quad (3.24)$$

$$\mu_{EH}(f_p(n)) = f(f_p(n); C, 0.3C, 0). \quad (3.25)$$

The membership functions for terms $L, M,$ and H in $T(D_c(n))$ are defined as

$$\mu_L(D_c(n)) = g(D_c(n); 0, 0.125, 0, 0.375), \quad (3.26)$$

$$\mu_M(D_c(n)) = f(D_c(n); 0.5, 0.25, 0.25), \quad (3.27)$$

$$\mu_H D_c(n) = g(D_c(n); 0.875, 1, 0.375, 0). \quad (3.28)$$

The membership functions for terms in $T(f_l(n))$ are defined as fuzzy singletons, denoted by

$$\mu_T(f_l(n)) = f(f_l(n); x_T, 0, 0), \quad (3.29)$$

where $T = EL, VL, PL, L, SL, M, SH, H, PH, VH,$ or $EH,$ and $x_{EL} = 0, x_{VL} = 0.1C, x_{PL} = 0.2C, x_L = 0.3C, x_{SL} = 0.4C, x_M = 0.5C, x_{SH} = 0.6C, x_H = 0.7C, x_{PH} = 0.8C, x_{VH} = 0.9C, x_{EH} = C.$ Notice that the center value of the triangular membership function f of each term for $f_p(n)$ is the same as the center value of the singleton function f of the same term for $f_l(n),$ where these terms are $EL, PL, SL, SH, PH,$ and $EH.$ Fig. 3.8 illustrates all membership functions of FFG.

There are 18 fuzzy rules for FFG. As shown in Table 3.2, the order of significance of the input linguistic variables is $f_p(n)$ then $D_c(n).$ These fuzzy rules are set in such a way that the generation of $f_l(n)$ mainly refers to $f_p(n)$ but slightly adjusted by $D_c(n)$ so as to achieve lower convergence time and thus higher the throughput. When $f_p(n)$ is "EL" or "PL", $f_l(n)$ is designed to raise two levels more than $f_p(n)$ ($EL \rightarrow PL$ or $PL \rightarrow SL$) if $D_c(n)$ is "L" and $f_l(n)$ remains unchanged if $D_c(n)$ is "H". This intends to increase the throughput. When $f_p(n)$ is "SL", "SH", or "PH", $f_l(n)$ decreases one level less than $f_p(n)$ if $D_c(n)$ is "H" and $f_l(n)$ increases one level

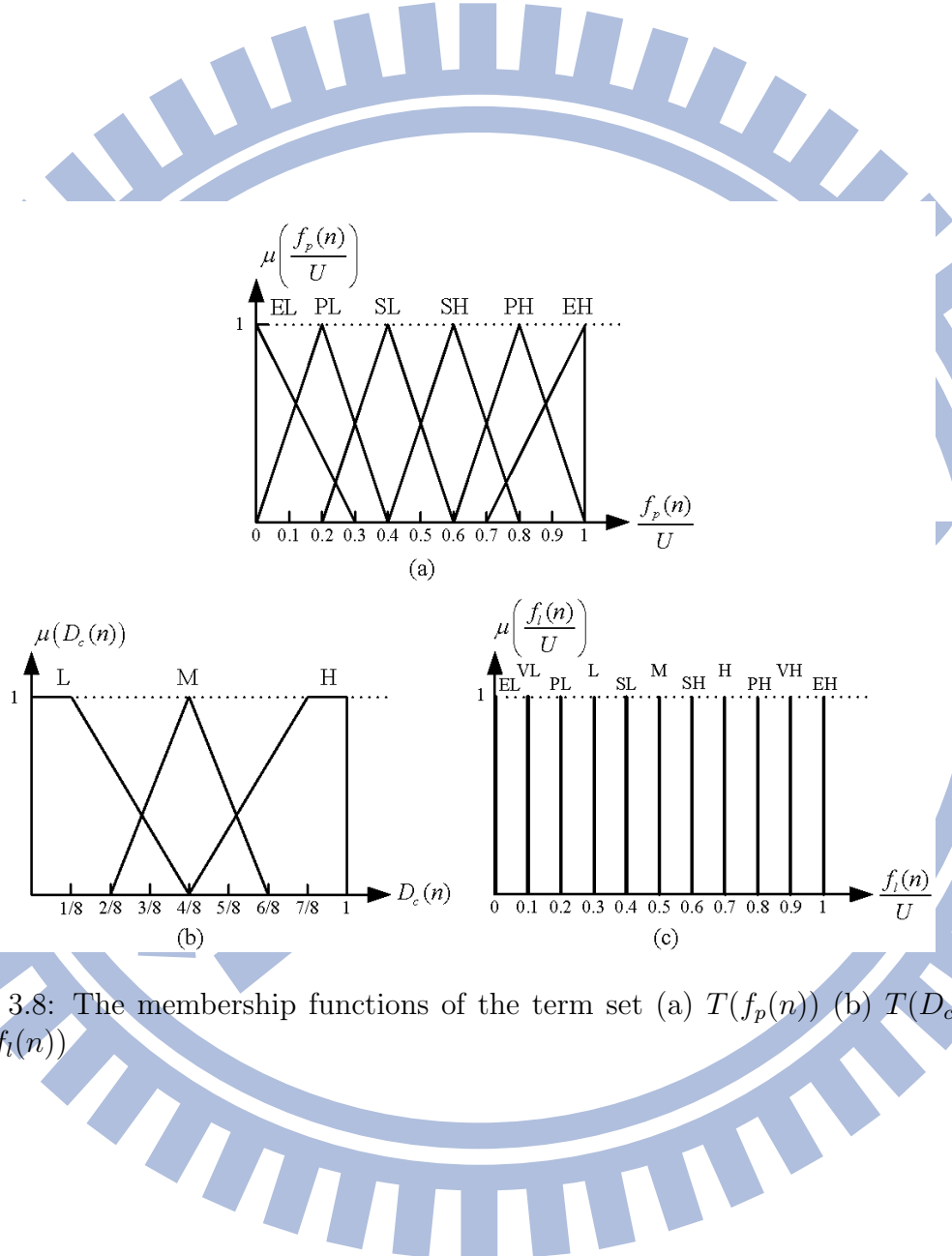


Figure 3.8: The membership functions of the term set (a) $T(f_p(n))$ (b) $T(D_c(n))$ (c) $T(f_l(n))$

Table 3.2: The rule base of FFG

Rule	$f_p(n)$	$D_c(n)$	$f_l(n)$	Rule	$f_p(n)$	$D_c(n)$	$f_l(n)$
1	<i>EL</i>	<i>L</i>	<i>PL</i>	10	<i>SH</i>	<i>L</i>	<i>H</i>
2	<i>EL</i>	<i>M</i>	<i>VL</i>	11	<i>SH</i>	<i>M</i>	<i>SH</i>
3	<i>EL</i>	<i>H</i>	<i>EL</i>	12	<i>SH</i>	<i>H</i>	<i>M</i>
4	<i>PL</i>	<i>L</i>	<i>SL</i>	13	<i>PH</i>	<i>L</i>	<i>VH</i>
5	<i>PL</i>	<i>M</i>	<i>L</i>	14	<i>PH</i>	<i>M</i>	<i>PH</i>
6	<i>PL</i>	<i>H</i>	<i>PL</i>	15	<i>PH</i>	<i>H</i>	<i>H</i>
7	<i>SL</i>	<i>L</i>	<i>M</i>	16	<i>EH</i>	<i>L</i>	<i>EH</i>
8	<i>SL</i>	<i>M</i>	<i>SL</i>	17	<i>EH</i>	<i>M</i>	<i>VH</i>
9	<i>SL</i>	<i>H</i>	<i>L</i>	18	<i>EH</i>	<i>H</i>	<i>PH</i>

larger than $f_p(n)$ if $D_c(n)$ is "L". When $f_p(n)$ is "EH", $f_l(n)$ should be decreased two levels less than $f_p(n)$ ($EH \rightarrow PH$) if $D_c(n)$ is "H" and $f_l(n)$ remains unchanged if $D_c(n)$ is "L". This intends to achieve RIAS fairness. Finally, the defuzzifier uses the min-max method mentioned in Section 3.4.2 to generate a crisp-valued local fairRate.

3.5 Simulation Results and Discussions

In the simulations, settings for the environment include 10 Gbps link capacity, 100 μs propagation delay between stations, 4 Mbytes STQ size, and 100 μs agingInterval. The value of the *stqHighthreshold* is 1 Mbytes and the value of the *stqLowthresold* is 0.5 Mbytes. Simulations for the proposed FLAG, DBA with moving average technique (DMA), DBA [24, 32, 33], and AM [17, 22] also conducted for performance comparison. Simulation results are recorded per agingInterval. Also, assume that the reserved bandwidth is zero, and only fairness eligible (FE) traffic flow is considered.

Fig. 3.9(a) shows a small parking lot scenario where there are 5 (0 ~ 4) greedy

stations, and Figs. 3.9(b), 3.9(c), 3.9(d) and 3.9(e) present the throughput of each flow by AM, DBA, DMA, and FLAG, respectively. This small parking lot scenario assumes that flows are generated from station 0, 1, 2, and 3 but terminated at station 4. The propagation delay is *small*. It can be seen that FE flows of AM, DBA, DMA, and FLAG take 49ms, 14ms, 13.5ms, and 7ms to stabilize, respectively. Thus FLAG improves by 7 times over AM and by 2 times over DBA, in the convergence time of traffic flows. The reasons are given as follows. The fuzzy logics provides a robust mathematical method to solve problems which are complicated to find a proper mathematical model for them. Especially, the FLAG contains sophisticated functional blocks, which combine advantages of AM and DBA. It fine-tunes the so-called p -fairRate generated by AFC, according to the congestion degree softly determined by the FCD using the fuzzy logic and the effective fuzzy rules designed in FFG by expert's domain knowledge. On the other hand, the DBA and DMA generate the local fairRate depending *only* on the short-term (average) arriving FE traffic flow, or equivalently the change rate of the STQ, without considering the STQ occupancy which usually used to determine the congestion degree of station given in [17]. This would incorrectly limit the amount of the passing transit FE traffic flow to the next station and cause DBA make error decision. For example, if the amount of the short-term arriving transit FE traffic flow is large but the STQ occupancy of a station is short, the station should not seriously regulate the FE traffic flow of its upstream stations. Also, AM generates a local fairRate which is equal to the added FE traffic flow rate of the station to regulate the flow when the station is in congestion. AM immediately sets the advertised fairRate as FullRate to allow the upstream stations to un-limitedly send traffic flow when the congestion is

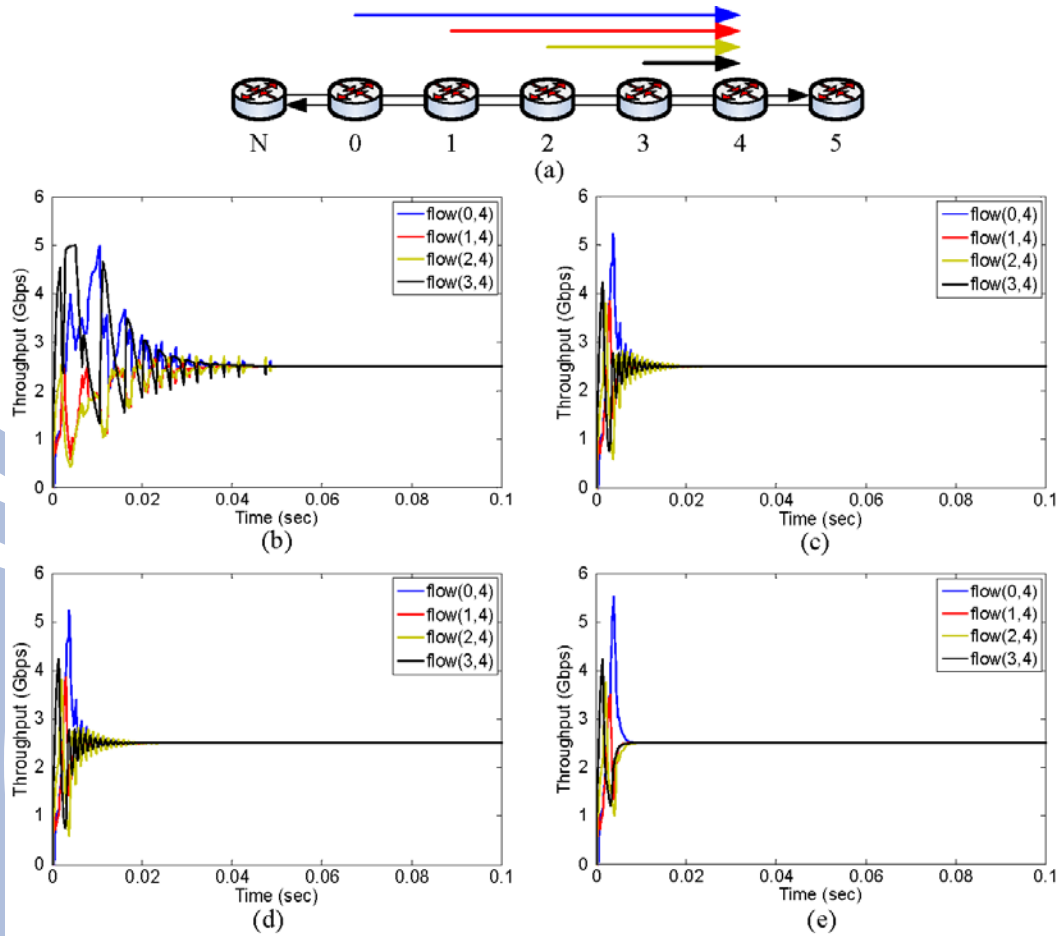


Figure 3.9: (a) Small parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DBA with moving average (DMA), and (e) FLAG.

released. This too-much variation of the advertised fairRate would cause the station congestion again and thus make the flow of AM damping the longest.

Fig. 3.10(a) shows a large parking lot scenario where there are containing 8 (0 ~ 7) greedy stations, and Figs. 3.10(b), 3.10(c), 3.10(d) and 3.10(e) present the throughput of flow(0, 7), flow(2, 7), flow(4, 7), and flow(6, 7) at station 7 by AM, DBA, DMA, and FLAG, respectively. This scenario differs from the previous

one of Fig. 3.9 in that the propagation delay would be *large*. It can be seen that the FLAG and the AM take 11ms and 27ms to stabilize the flows, respectively; unfortunately, DBA and DMA take quite a long time to stabilize the traffic flows. It is because that DBA computes the number of the effective IA flows referring to both the short aggregating traffic (per agingInterval) and the pervious local fairRate to generate the current local fairRate. However, due to the large propagation delay, the correlation between the short aggregating traffic and the pervious local fairRate becomes low. Therefore, DBA cannot generate a correct local fairRate to regulate flows. Thus the flows oscillate and converge slowly; the convergence time takes about 0.15s which is not shown here. The DMA uses the moving average technique to lessen the effect of propagation delay. The flow oscillation of the DMA is half smaller than the DBA but still exists. Since without considering the STQ occupancy for the congestion degree of station, the DMA incorrectly limits the amount of the passing transit FE traffic flow to the next station. On the other hand, the FLAG can correctly generate the p -fairRate to meet the RIAS fairness and diminish the effect of the propagation delay to some extent. Also, the FLAG finely adjusts the p -fairRate to a precise local fairRate according to both the congestion degree and the effective fuzzy rules well designed by domain knowledge. The main reason that AM in this scenario takes less time to stabilize all flows than AM in the previous scenario shown in Fig. 3.9(b) is given below. Since, here in Fig. 3.10(a), there are more stations with greedy traffic, more aggregated traffic per agingInterval will be caused. This more aggregated traffic and the larger propagation delay would make the station congestion always occur earlier. Afterwards, the station would not have the chance to set the advertised fairRate as FullRate. Thus the convergence time is

shorter.

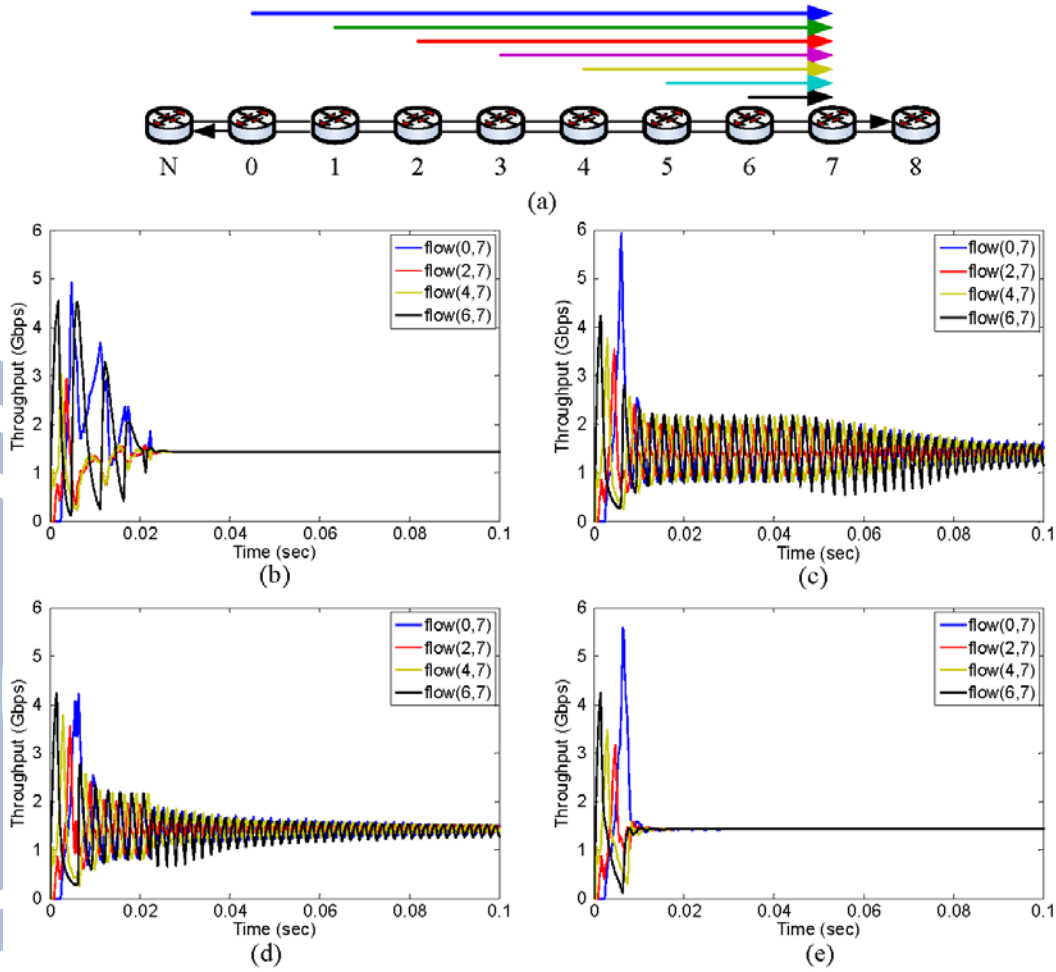


Figure 3.10: (a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.

Fig. 3.11(a) shows a large parking lot scenario where there are containing 8 (0 ~ 7), such as in Fig. 3.10(a) but with various *finite* traffic demands, greedy stations, and Figs. 3.11(b), 3.11(c), 3.11(d), and 3.11(e) present throughputs of flow(0, 7), flow(2, 7), flow(4, 7), and flow(6, 7) at station 7 by AM, DBA, DMA, and FLAG, respectively. Assume that flow(0, 7) and flow(1, 7) require 2.1 Gpbs, flow(4, 7) and flow(5, 7) require 1.5 Gpbs, and flow(2, 7), flow(3, 7) and flow(6,

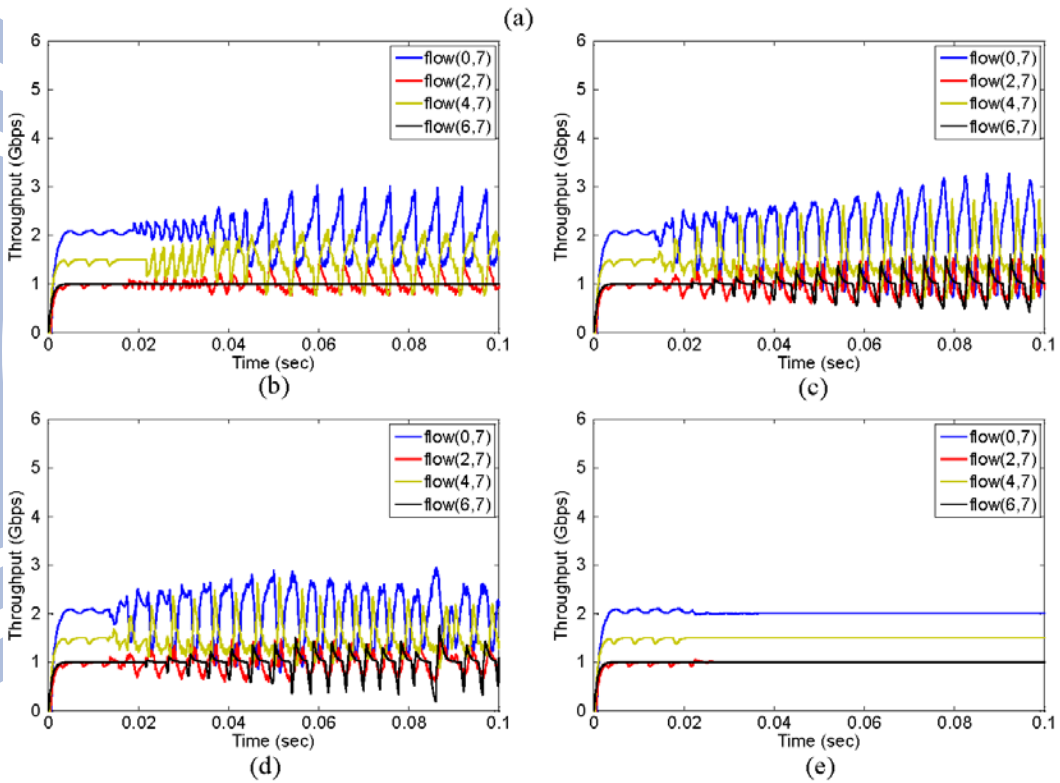
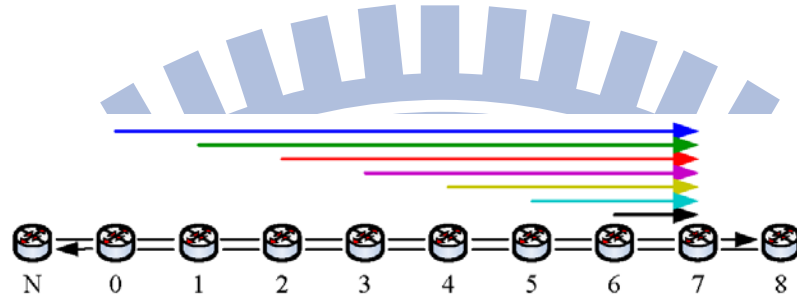


Figure 3.11: (a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG in a large parking lot scenario with various finite traffic flows.

7) require 1.0 Gbps. It would be facts that station 6 will be the *first* one to incur congestion, and the added FE traffic flow to network at each station cannot always match its received fairRate due to the finite traffic demand at each station. Also, flow(0,7) and flow(1,7) will have the highest throughput when station 6 is in free-congestion or the remaining bandwidth is large because of their largest required traffic demands. It can be seen that at the first beginning, all flows just oscillate slightly, and then AM, DBA, and DMA oscillate all the ways, while FLAG can make all flows converge but takes 30 ms. It is because that FLAG indeed diminishes the effect of the propagation delay and generates the correct local fairRate at each agingInterval. Also, since each traffic flow is with different finite traffic demand and is much less than that of the greedy case in Fig. 3.10(e), the damping amplitude is smaller than that in Fig. 3.10(e). Moreover, the FLAG stably realizes the RIAS fairness and has higher throughput by about 2.8%, 3.5%, and 2.4% than AM, DBA and DMA, respectively. On the other hand, the advertised fairRate by AM is often set as FullRate in this scenario because the bandwidth of the total demand traffic is 10.2 Gbps, slightly higher than the link capacity but much less than that of the greedy case in Fig. 3.10(b). In this situation, the aggregated traffic per agingInterval would be smaller, and the congestion, if any, could be solved by AM most of time. Thus, the flows by AM oscillate always and the flow(0,7) seriously oscillates due to its largest traffic demand. By DBA, its generation accuracy of local fairRate is susceptible to the propagation delay, as seen in Fig. 3.10. Also, in this scenario, station 0 and station 1 are the farthest ones to station 6 and flow(0,7) and flow(1,7) are with the largest traffic demand. These facts result in that flow(0,7) and flow(1,7) cannot be regulated by the station 6 quickly. This violent varying

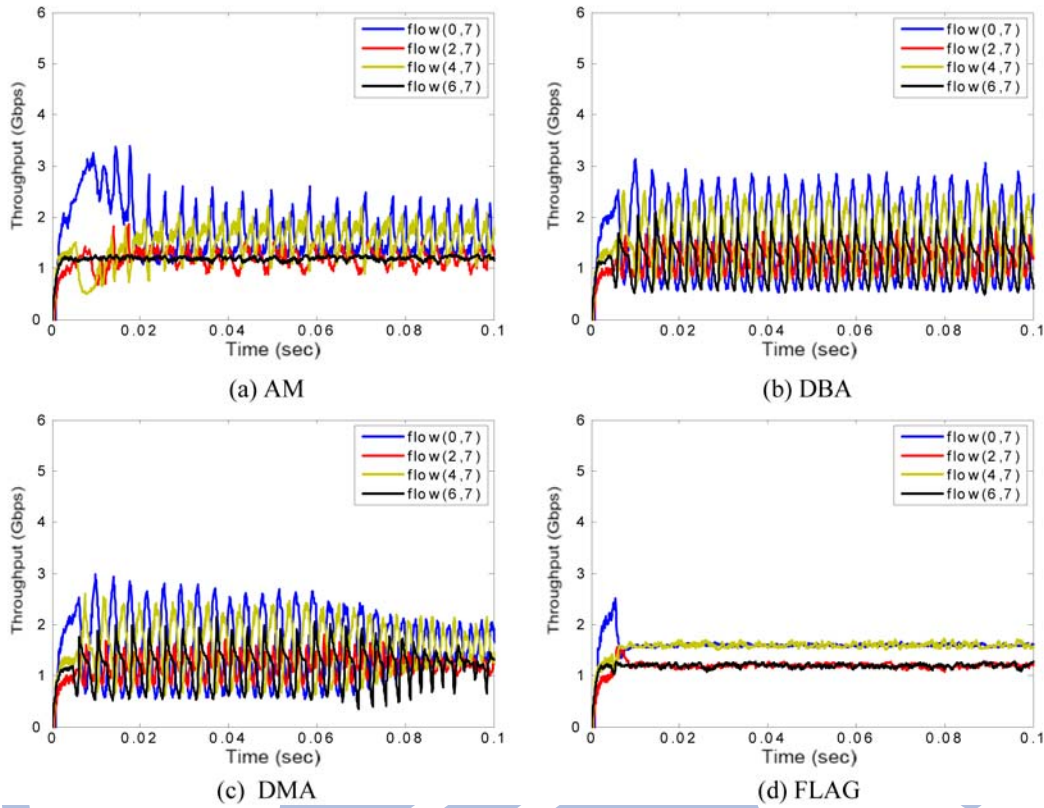


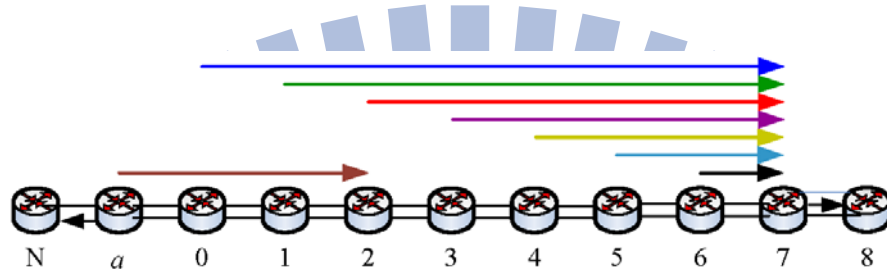
Figure 3.12: The throughputs of (a) AM, (b) DBA, (c) DMA, and (d) FLAG in a large parking lot scenario containing 8 stations, where each flow is with truncated Pareto traffic model.

aggregation traffic per agingInterval and the effect of the propagation delay thus result in DBA generating the local fairRate improperly. Notice that if flow(0,7) requires less traffic demand, the oscillation amplitude of flows will be smaller. The DMA has the same phenomenon but its performance is better than DBA by 1.5% due to using the moving average technique.

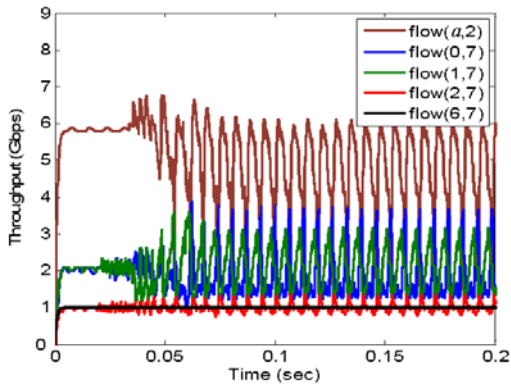
Figs. 3.12 (a), 3.12 (b), 3.12 (c), and 3.12 (d) present throughputs of flow(0, 7), flow(2, 7), flow(4, 7), and flow(6, 7) at station 7 by AM, DBA, DMA, and FLAG, respectively, in a large parking lot scenario containing 8 stations as in Fig. 3.10(a),

where each flow is with truncated Pareto traffic model [52]. Assume that flow(0, 7) and flow(1, 7) require 2.1 Gbps, flow(4, 7) and flow(5, 7) require 1.5 Gbps, and flow(2, 7), flow(3, 7) and flow(6, 7) require 1.0 Gbps. We can see that the phenomena of all flows are the same as those in Fig. 3.11, where all algorithms oscillate all the ways but FLAG makes all flows be with the smallest oscillation comparing with the other three algorithms. Thus we can claim that, due to the robustness and the sophisticate of the proposed FLAG for the fairness control, the FLAG can still perform better than the other schemes in the cases of realistic traffic models.

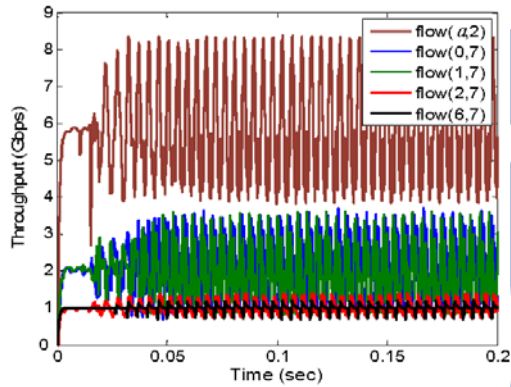
Fig. 3.13(a) shows an available bandwidth reclaiming scenario where there are 9 stations with *finite* traffic demand and a spatial reuse of flow($a,2$) occurs, and Figs. 3.13(b), 3.13(c), 3.13(d) and 3.13(e) present the throughput of flow($a,2$) at station and flow(0,7), flow(1,7), flow(2,7), and flow(6,7) at station 7 by AM, DBA, DMA, and FLAG, respectively. In this scenario, the flow($a, 2$) requires 5.9 Gpbs, and similar to Fig. 3.11, flow(0, 7) and flow(1, 7) require 2.1 Gpbs, flow(4, 7) and flow(5, 7) require 1.5 Gpbs, and flow(2, 7), flow(3, 7), and flow(6, 7) require 1.0 Gpbs. It can be seen that, just as in Fig. 3.11, at the beginning, all flows of all algorithms oscillate slightly, and finally FLAG makes all flows stabilize but takes 78 ms, while AM, DBA, and DMA oscillate all the ways. The reasons that all algorithms in this scenario behave worse than in the large parking lot scenario with various finite traffic flows, given in Fig. 3.11, are as follows. Since flow($a,2$) is sunk at station 2, station 1 would have more transient FE traffic flows than station 2, where station 1 has 10.1 Gpbs traffic flow maximum, while station 2 has 5.2 Gpbs traffic flow maximum. This phenomenon is conversed in Fig. 3.11, where station 1 has 4.2 Gpbs traffic flow maximum, while station 2 has 5.2 Gpbs maximum. Therefore, the



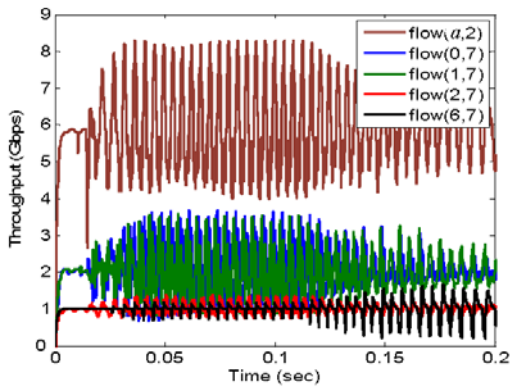
(a)



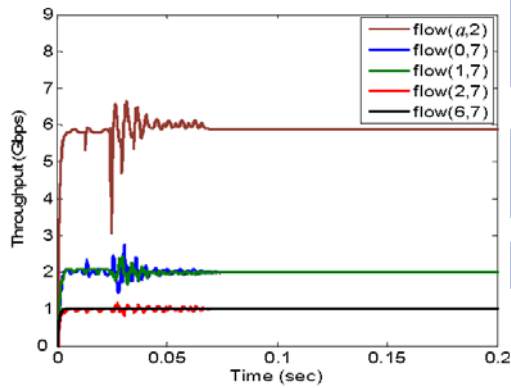
(b)



(c)



(d)

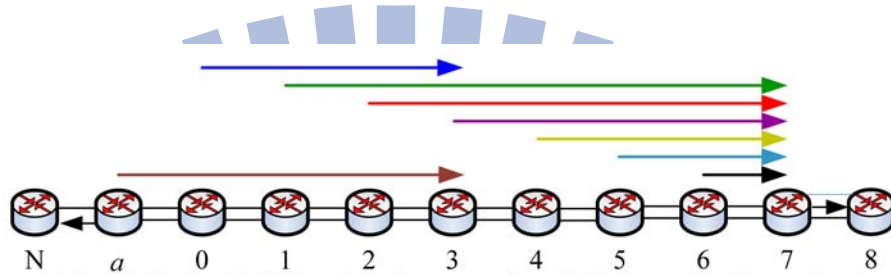


(e)

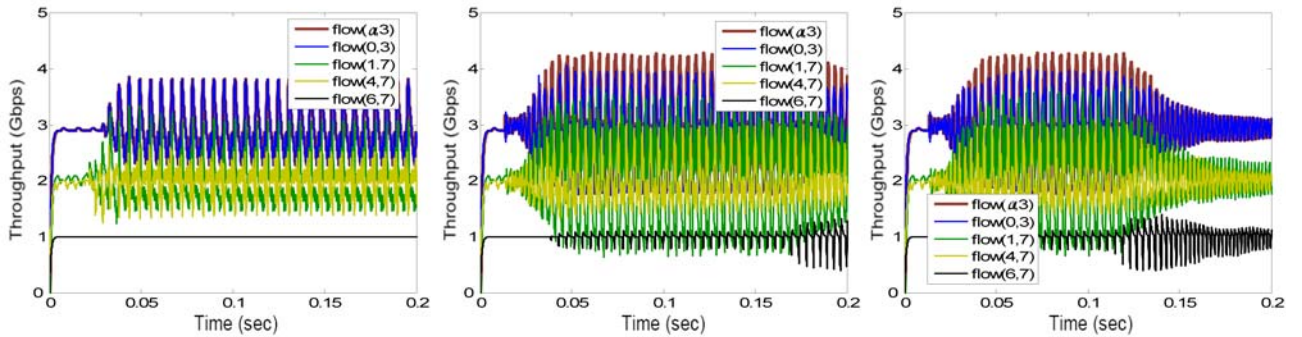
Figure 3.13: (a) Available bandwidth reclaiming scenario with finite traffic demand, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.

station 1 in Fig. 3.13 will more frequently and heavily regulate its station 0, which has 5.9 Gbps transient traffic flow and 2.1 Gbps local traffic flow, than the station 1 in Fig. 5 will regulate its station 0, which has only 2.1 Gbps local traffic flow. Thus it can be believed that all flows in Fig. 3.13 would oscillate worse than in Fig. 3.11 for all schemes. Moreover, according to our computation, the throughput at station 6 by FLAG is about 0.990, which is higher than AM's 0.825, DBA's 0.914, and DMA's 0.933. The reasons would be the same as those given before and are not mentioned again here.

Fig. 3.14(a) shows an available bandwidth reclaiming scenario with reuse traffic flows, where there are 9 stations with finite traffic demand and two spatial reuses of flow(a , 3) and flow(0, 3). Figs. 3.14(b), 3.14(c), 3.14(d), 3.14(e) and 3.14(f) present throughputs of flow(a , 3) and flow(0, 3) at station 3 and throughputs of flow(1, 7), flow(4, 7), and flow(6, 7) at station 7 by AM, DBA, AFC, FLAG, and M-FLAG, respectively, where the M-FLAG denotes the modified FLAG with DBA to replace AFC. In this scenario, flow(a , 3) and flow(0, 3) require 3.0 Gbps, flow(1, 7) and flow(2, 7) require 2.1 Gbps, flow(3, 7) and flow(6, 7) require 1.0 Gbps, and flow(4,7) and flow(5, 7) require 2.0 Gbps. It can be seen that, similar to the phenomena illustrated in Fig. 3.11, at the beginning, all flows of all algorithms oscillate slightly, and finally FLAG and M-FLAG make all flows stabilized and take 48 ms and 46 ms, respectively, but M-FLAG has oscillations much larger than FLAG during the transitional period, while AFC converges at about 1129 ms and AM and DBA oscillate all the ways. Also, all algorithms in this scenario behave worse than in the large parking lot scenario with various finite traffic flows given in Fig. 3.11. The reasons are as follows. Since flow(a , 3) and flow(0, 3) are sunk at station 3,



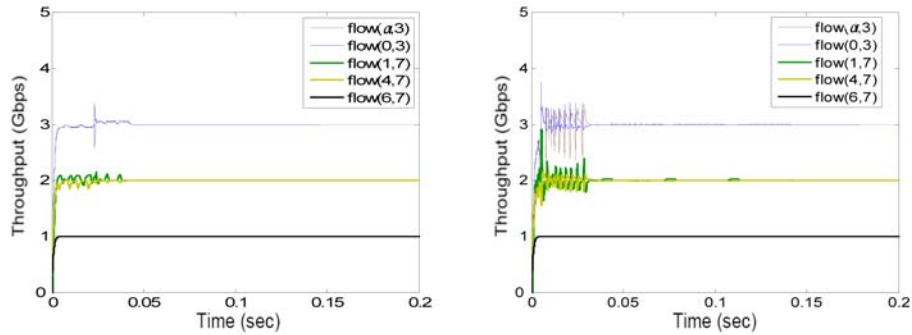
(a) Available bandwidth reclaiming scenario with finite traffic demand and two reuse traffic flows



(b) AM

(c) DBA

(d) DMA



(e) FLAG

(f) M-FLAG

Figure 3.14: (a) Available bandwidth reclaiming scenario with finite traffic demand and two reuse traffic flows, and the throughput of (b) AM, (c) DBA, (d) DMA, (e) FLAG and (f) M-FLAG.

station 2 would have more transient FE traffic flows than station 3, where station 2 has 10.2 Gbps traffic flow maximum, while station 3 has 5.2 Gbps traffic flow maximum. This phenomenon is conversed in Fig. 5, where station 2 has 5.2 Gbps traffic flow maximum, while station 3 has 6.2 Gbps maximum. Therefore, the station 2 in Fig. 3.14 will more frequently and heavily regulate its station 1, which has 6.0 Gbps transient traffic flow and 2.1 Gbps local traffic flow, than the station 2 in Fig. 3.11 will regulate its station 1, which has 2.1 Gbps transient traffic flow and 2.1 Gbps local traffic flow. Thus it can be believed that all flows in Fig. 6 would oscillate worse than those in Fig. 5 for all algorithms. Also, that the M-FLAG has oscillations larger than the FLAG during the transitional periods shows that the AFC can indeed diminish the effect of the propagation delay once occurred in DBA. Moreover, according to our computation, the throughput at station 6 by FLAG is about 0.993, which is higher than AM's 0.842, DBA's 0.921, AFC's 0.943, and M-FLAG's 0.988.

3.6 Concluding Remarks

In this chapter, an effective fuzzy local fairRate generator (FLAG) is proposed for resilient packet ring (RPR). The FLAG is sophisticatedly composed of three function blocks: an adaptive fairRate calculator (AFC), a fuzzy congestion detector (FCD), and a fuzzy fairRate generator (FFG). The AFC pre-generates a fairRate, which meets RIAS fairness and can diminish the effect of the propagation delay. The FCD softly detects the congestion degree of station, considering STQ queue length and its change rate which is the arriving transit FE traffic flows to STQ. Subsequently, the FFG generates a suitable local fairRate by intelligently fine-tuning the

pre-generated fairRate, using fuzzy logics, based on the congestion degree of the station. The FLAG can make traffic flows satisfy RIAS fairness criterion and converge to an ideal fairRate in an efficient way. Simulation results show that each flow by FLAG is indeed close to the designated rate with the smallest damping amplitude and the least convergence time in the parking lot scenarios and the available bandwidth reclaiming scenario, compared to conventional AM, DBA, and DBA fairness algorithms. These prove that the configuration of FLAG is indeed sophisticated, where AFC pre-generates the local fairRate using the moving average technique; FCD determines the congestion degree of station using fuzzy logics, considering not only the STQ length but also change rate of STQ length; and finally the FFG adopts the fuzzy logics and the expert's domain knowledge to precisely generate the local fairRate by fine-tuning the pre-generated local fairRate by AFC according to the congestion degree by FCD. Also, the performance superiority of DMA over DBA proves that the moving average technique is indeed effective to diminish the effect of propagation delay on the stability of traffic flows.

Chapter 4

Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings

4.1 Introduction

The resilient packet ring (RPR) is a dual-ring-based network protocol and has been recently approved as the IEEE 802.17 Standard [17]. A RPR network consists of a clockwise (CW) and a counter-clockwise (CCW) ringlets, giving each station on the ring a full duplex connection to its neighbors. It can be used for implementing local area networks (LAN) and metropolitan area networks (MAN) at rates scalable to many gigabits per second. More than one RPR can be interconnected by a *bridge* which forwards packets from one RPR to another. A spatially aware sublayer (SAS), which is a part of the MAC layer, in the bridge is used to decide which ringlet interface the packet should be routed to [17, 26]. Current research on SAS, including the IEEE 802.17b Working Group, is mainly focusing on how to modify this sublayer in order to avoid flooding the entire bridged network when transmitting inter-ring packets [17, 26, 27, 28].

Settawong and Tanterdtid proposed an enhancement by using a topology dis-

covery and spanning tree algorithm [27]. The algorithm can manage traffic between rings more efficiently and can remove the need for flooding. The shortest path route controller (SPRC) was widely considered for metro rings [38, 39, 40] as it can maximize the spatial reuse and thus the achievable packet throughput for uniform traffic. However, as traffic load increases, incoming call requests could pile up at a node before being processed, and these would result in a potential bottleneck in network performance [40]. Also, Heiden et. al. analyzed the capacity of bidirectional optical packet ring networks, such as RPR, which employs the SPRC for multicast hotspot traffic [41]. They found that when the multicast traffic originating at the hotspot exceeds a critical threshold, the SPRC leads to a significant capacity reduction.

Intuitively, the route selection would be closely related with the congestion degree of the ringlet so as to follow the load balancing principle. Generally, RPR uses a queue length threshold to detect the congestion and a node's adding rate limitation to avoid the network congestion [17]. Therefore, an intuitive queue-length threshold route controller (QTRC) would be better than the SPRC. However, the correlation function between the congestion degree and these variables is nonlinear and complicated.

Recently, intelligent techniques such as fuzzy logics and neural networks have been widely applied to control nonlinear, time-varying, and well-defined systems for that fuzzy logic and neural network control can provide *effective* solutions with *small* computational complexity. Fuzzy set theory appears to be able to support a robust mathematical framework for dealing with real-world imprecision, and exhibits a soft behavior, which means a greater ability to adapt itself to dynamic, imprecise, and bursty environments [53]. Fuzzy and neural fuzzy implementations

of the two-threshold congestion control method and the equivalent capacity admission control method were once studied in the literature [53, 54]. Results have shown that the proposed fuzzy logic and neural network approaches significantly improve system performance, compared to conventional approaches. Moreover, fuzzy logic and neural network systems are easily implemented in a chip. This will greatly reduce the computational time and make fuzzy logic and neural network control feasible for real applications.

Therefore, we propose intelligent inter-ring route control for bridged resilient packet rings in this paper. Either CW or CCW ringlet at bridge will be properly chosen for an incoming new call request from one RPR to the other. The selection is based on the load balancing principle which is in the sense that the selected ringlet would be with lower congestion degree and higher service rate [46]. An *intelligent inter-ring route controller* (IIRC) is designed to contain a *fuzzy bridge-node congestion indicator* (FBCI) to intelligently detect the congestion degree of bridge, and a *pipeline recurrent neural networks* (PRNN) *downstream-node fairness predictor* (PDFP) to effectively predict the mean received fairRate. Besides, the IIRC consists of a *fuzzy router controller* (FRC) to determine preference values of route of CW and CCW ringlets according to the congestion indication provided by FBCI, the predicted mean received fairRate provided by PDFP, the *number of hops to destination*, and the *service rate of the bridge*. A ringlet with a larger route preference value would be more proper to be selected. Simulation results show that the IIRC can effectively attain the load balancing property and improve the packet dropping probability (average packet delay, throughput) by 10% and 220% (13% and 18%, 6% and 19%) over QTRC and SPRC [38], respectively, in a scenario.

This is due to the fact that the IIRC sophisticatedly detects the system congestion degree and correctly predicts the mean received fairRate using fuzzy system and neural network. Also, IIRC achieves higher throughput by 7% and 6.7% than IIRC itself but without considering the prediction of the received fairRate and without considering the amount of the reserved bandwidth as well as the equivalent capacity for a new call request, respectively.

The rest of this chapter is organized as follows. In section 4.2, the system model is described. The neural networks and learning mechanism are presented in section 4.3, along with two popular architectures for implementing a neural network controller. The intelligent inter-ring route controller (IIRC) and its functional blocks: FBCI, PDFP, and FRC, are designed in Section 4.4. Section 4.5 presents simulation results and compares the proposed IIRC scheme to the QTRC and SPRC schemes. Finally, concluding remarks are given in Section 4.6.

4.2 System Model

4.2.1 Architecture of Bridge Node

Fig. 4.1 shows a bridge node connecting R_0 and R_1 RPR rings, where each ring contains a clockwise (CW) ringlet and a counter-clockwise (CCW) ringlet and there are M nodes on the ring. Assume that the fiber link capacity of the ringlet is C Gbps and the distance between every two consecutive nodes in the ringlet is the same. The proposed intelligent inter-ring route controller (IIRC) is installed in a spatially aware sublayer (SAS). As a new call request coming from one ring to the other, the IIRC will determine an appropriate ringlet for the inter-ring new call request. Also, the SAS forwards packets of existing calls to their interface in the

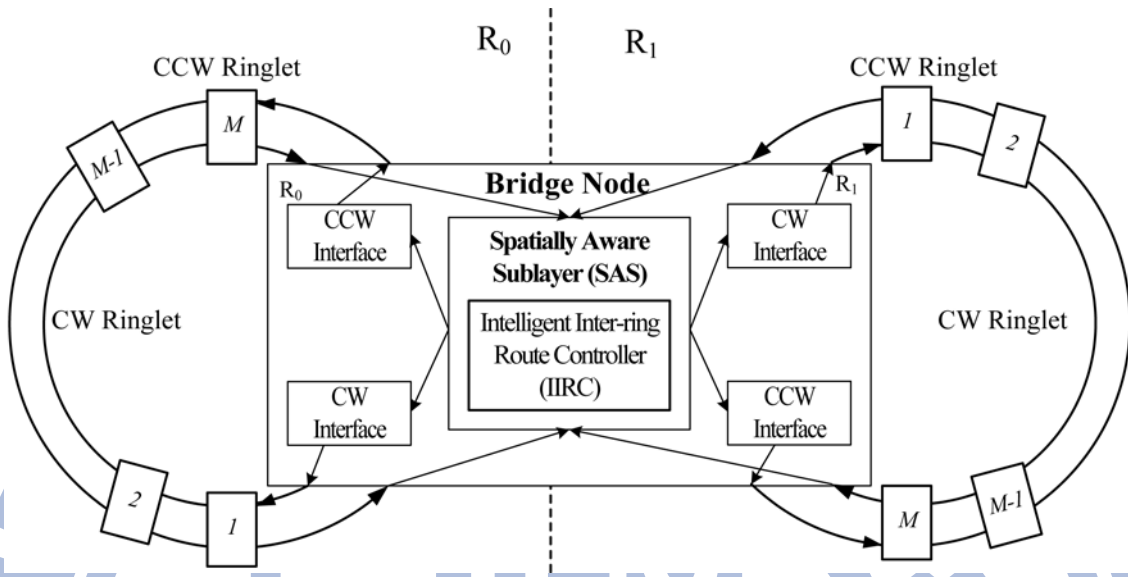


Figure 4.1: Architecture of the bridge node

bridge node based on the determined route. The bridge node has one interface associated each ringlet, and as shown in Fig. 4.2, each interface has two transit buffers: the ringlet and ingress buffers. The packets to the same ring are stored in the ringlet buffer, and those to the other ring are buffered in the ingress buffer. Each buffer contains a primary transit queue (PTQ) and a secondary transit queue (STQ). The high- (low-) priority packets, such as *Class A* and *Class B-CIR* (*Class B-EIR* and *Class C*), are stored in the PTQ (STQ). Voice packets, video packets of I-frame, video packets of B- or P-frames, and data packets are classified as Class A, Class B-CIR, Class B-EIR, and Class C, respectively. The bridge node always reserves bandwidth for the high-priority traffic. The scheduler in the bridge first serves the PTQs exhaustively with the round robin policy, and then serves the two STQs with the proportional round robin policy associated with their queue lengths [55].

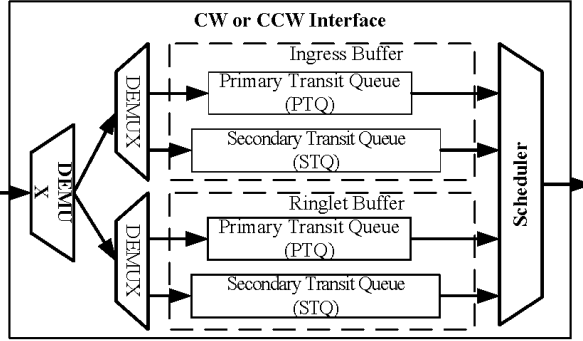


Figure 4.2: Architecture of the interface

4.2.2 Fairness Algorithm

There are two fairness algorithms, called aggressive mode (AM) and conservative mode (CM), proposed in IEEE 802.17 standard [17], and another fairness algorithm, called distributed bandwidth allocation (DBA), proposed by Alharbi and Ansari [24, 32, 33]. For simplicity, we adopt the AM fairness algorithm in each node for simulations. The AM fairness algorithm is described as follows. As specified in [17], if a node finds that its STQ queue length is longer than a threshold, it regards that congestion occurs and will initiate the AM fairness algorithm to limit its upper node's add rate of the low-priority traffic to relieve congestion. The AM generates a limited value, called fairRate whose value is the available add rate of the low-priority traffic of node, each frame time period $100 \mu\text{s}$. If a node finds that its upper node's forward rate is less than its received fairRate, it will release the upper node's add rate limitation by sending a fairRate with a special value, called FullRate, and the service rate of the node is the total link capacity C . If the received fairRate is not a FullRate, the node will limit its adding rate, which is bounded by the received fairRate, into the ring, and the service rate of the node is the summation of the

arrival rate to the STQ, its received rate, and the reserved rate for high-priority traffic.

4.3 Neural Network Controller

4.3.1 Neural Networks and its Learning Capability

Neural networks are inspired by modeling networks of real (biological) neurons in the brain. It has a large number of highly interconnected processing elements which correspond to biological neurons and thus also be called *artificial neurons*, or simply *neurons*. The nodes are configured in regular architectures and can usually operate in parallel to make the whole network as a parallel distributed information processing structure. The collective behavior of an neural network, like a human brain, demonstrates the ability to learn, recall, and generalize from training pattern or data. The building blocks of neural network consists of three basic entities: *neurons model*, *connectionist structures* (among neurons) and *learning rules* [43]. Neurons are the basic information processing elements and can be viewed as consisting of two parts in the mathematical model: input part and output part. Associated with the input of a neuron is an *integration function* f which serves to combine information, activation, or evidence from an external source or other neurons into a *net input* to the neuron. The integration function f is usually a linear function of the input. A second action of each neuron is to output an *activation value* as a function of its net input through an *activation function* or *transfer function* $a(f)$. The step function, unipolar sigmoid function and bipolar sigmoid function are commonly used examples of the activation function. The connectionist structures are then applied to link neurons to mimic how the human brain works while the learning rules are applied

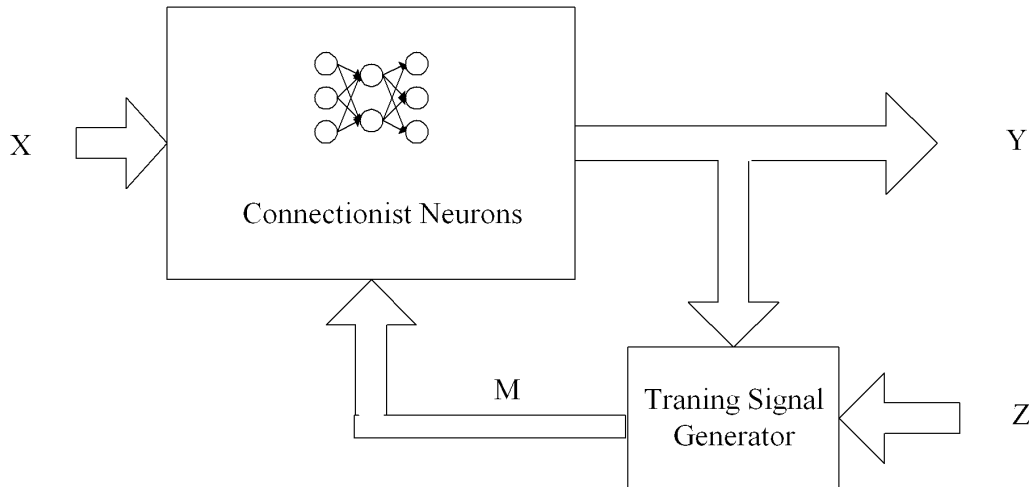


Figure 4.3: The basic structure of neural network

to adaptively modify the behavior of the neural networks through past experience. Fig. 4.3 shows the basic concept of neural network. In the figure, X is the input signal, Y is the actual output, Z is the reference signal, and M is the training signal. The *connectionist neurons* block computes the output signal Y for input signal X and then the *training signal generator* block will generate a training signal according to a specified learning rules. The training signal is used to update the weighting of the nodes in the neural networks.

Generally speaking, the learning rules can be classified into three kinds of categories: *supervised learning*, *reinforcement learning*, and *unsupervised learning*. For different learning rules, there are different sets of Z and M . In the following, the main concepts of three learning rules are briefly described.

- **Supervised Learning**

In supervised learning, each input signal X has its own *desired* output D . Here, the reference signal Z is equal to desired output D . When the actual output Y is different from reference signal Z , an error occurs. Then, the training signal will be generated to adjust the weighting of the nodes in the neural network so that the actual output will approach the reference signal. Therefore, the supervised learning can be considered as a input/output mapping machine or a function approximation tool.

- **Reinforcement Learning**

In the reinforcement learning, there is *no desired* output, only a reinforcement signal R . The reinforcement signal is an evaluation value of the actual output Y . For example, in the control problems, the reinforcement signal may be “good” or “bad”. Here, the reference signal Z is equal to reinforcement signal R . Using the reinforcement signal R , a training signal is generated to update the weighting so that the actual output will achieve a better evaluation value in the future. Therefore, the reinforcement learning is *learning with a teacher*. Using the reinforcement learning, the neural network acts as a controller to make the system work better according to a pre-defined evaluation function.

- **Unsupervised Learning**

Unlike the previous two learning rules, there is no feedback information from the environment in the unsupervised learning. Neither the desired output or reinforcement signal are available. Instead, the training signal is generated from actual output Y and the internal weighting of the neural network. The training signal here

is used to increase the weightings of the nodes that connect to the actual output. That is, the correlation between the chosen input nodes and output data will be enhanced. In the unsupervised learning, the neural network discovers its patterns and the correlation through experiments, which is called *self-organizing*. Therefore, the unsupervised learning are usually applied to deal with the classification or clustering problems.

Like the condition in fuzzy logic controller, there are also diverse implementation ways to build a neural network as a controller.

4.3.2 Multilayer Feedforward Neural Networks

Multilayer feedforward neural networks, as shown in Fig. 4.4, is a typical model for implementing a neural network controller. The neural network controller possesses an ability to perfectly approximate a generic function from input/output data pairs $\{X, Y\}$. Consider a multilayer feedforward neural network $\text{NN}(X, W)$, with input vector X and a set of (link) weight vector W which will be updated by some learning rules; denote a continuous function by $Z = f(X) : D \subseteq R^{n_i} \rightarrow R^{n_o}$, where D is a compact metric space on R , and n_i (n_o) is the input (output) space dimension. The Stone-Weierstrass theorem [56] showed that $\text{NN}(X, W)$ (actual output) can be trained to asymptotically approach any continuous desired output function $f(X)$ as close as possible. That is, an $\text{NN}(X, W)$ with appropriate weight W can be found so that $\|\text{NN}(X, W) - f(X)\|_X < \varepsilon$ for an arbitrary $\varepsilon > 0$, where $\|e\|_X = \sum_{X \in D} \|e(X)\|^2$ and $\|\cdot\|$ is a vector norm. The neural network is a non-structured network, which cannot incorporate knowledge about system.

A back-propagation learning algorithm [57], which is a kind of supervised learn-

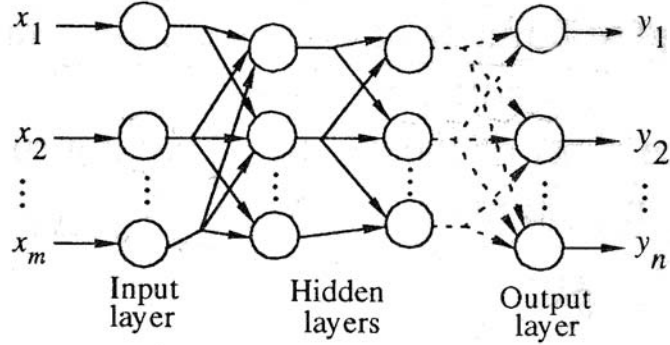


Figure 4.4: The structure of multilayer feedforward neural network

ing, is usually employed to train the neural network controller. Let $X(i)$ denote the vector randomly sampled from D and used as an input to the neural network controller at time instant t_i , let $\mathbf{NN}(X(i), W) = \hat{z}(i)$ denote the corresponding decision of the neural network controller, and let $f(X(i)) = z(i)$ denote the desired decision. The objective of the back-propagation learning algorithm is to minimize decision error E by recursively adjusting its weight in each layer, where E is defined as

$$\begin{aligned}
 E &= \frac{1}{2} \|\mathbf{NN}(X(i), W) - f(X(i))\|^2 \\
 &= \frac{1}{2} (\hat{z}(i) - z(i))^2.
 \end{aligned} \tag{4.1}$$

Consider an M -layer feedforward neural network. Each layer has a number of processing elements (neurons) which are fully interconnected with the neurons in neighboring layers via adaptive weights. Neurons in the input layer (layer $k = 1$) do not process the input data; they simply store input data values. Neurons in the hidden layers ($2 \leq \text{layer } k \leq M - 1$) and output layer (layer $k = M$) perform two operations. The j^{th} neuron in the k^{th} layer, for example, first calculates a weighted

sum, denoted by $S_j^{(k)}$, of all outputs $o_i^{(k-1)}$ of the $(k-1)^{th}$ layer. $S_j^{(k)}$ is given by

$$S_j^{(k)} = \begin{cases} x_j & \text{if } k = 1, \\ \sum_{i=1}^{n_{k-1}} w_{ji}^{(k)} o_i^{(k-1)} & \text{if } 2 \leq k \leq M, \end{cases} \quad (4.2)$$

where x_j is the input variable of the j^{th} neuron in the input layer, n_{k-1} is the number of neurons in layer $(k-1)$, and $w_{ji}^{(k)}$ is the weight of the link connected from the i^{th} neuron in layer $(k-1)$ to the j^{th} neuron in layer k . After that, the neuron further transforms $S_j^{(k)}$ into output $o_j^{(k)}$ via an activation function $G(\cdot)$. $o_j^{(k)}$ is expressed as

$$o_j^{(k)} = \begin{cases} S_j^{(k)} & \text{if } k = 1, \\ G(S_j^{(k)}) & \text{if } 2 \leq k \leq M. \end{cases} \quad (4.3)$$

The adjustment of weights is based on a steepest-descent algorithm [57]. It can be expressed as

$$w_{ji}^{(k),new} = w_{ji}^{(k),old} - \eta \left. \frac{\partial E}{\partial w_{ji}^{(k)}} \right|_{w_{ji}^{(k)} = w_{ji}^{(k),old}}, \quad (4.4)$$

where η is a gain term that determines the learning rate of the link weight. η is usually set equal to a positive constant less than unity. In order to obtain the partial derivative for the quadratic error E , an error produced by the j^{th} neuron in layer k , denoted by $\delta_j^{(k)}$, is obtained from

$$\delta_j^{(k)} = -\frac{\partial E}{\partial S_j^{(k)}}, \quad 1 \leq k \leq M, 1 \leq j \leq n_k. \quad (4.5)$$

It was shown in [57] that the error signals $\delta_j^{(k)}$'s can be computed according to a recursive procedure of the generalized delta learning rule [57] described as follows,

$$\delta_j^{(k)} = \begin{cases} G'(S_j^{(k)}) \sum_l \delta_l^{(k+1)} w_{lj}^{(k+1)} & \text{for } 2 \leq k \leq M-1, \\ (z - \hat{z})G'(S_j^{(k)}) & \text{for } k = M. \end{cases} \quad (4.6)$$

Once these error signal terms have been determined, the partial derivative for the quadratic error can be computed directly by

$$\frac{\partial E}{\partial w_{ji}^{(k)}} = \frac{\partial E}{\partial S_j^{(k)}} \frac{\partial S_j^{(k)}}{\partial w_{ji}^{(k)}} = -\delta_j^{(k)} o_i^{(k-1)}. \quad (4.7)$$

And the update rule for the back-propagation algorithm is then given by

$$w_{ji}^{(k),new} = w_{ji}^{(k),old} - \eta \frac{\partial E}{\partial w_{ji}^{(k)}} = w_{ji}^{(k),old} + \eta \delta_j^{(k)} o_i^{(k-1)}. \quad (4.8)$$

4.3.3 Radial Basis Function Neural Networks

The radial basis function neural networks (RBFN), which was suggested by Moody and Darken in [58], is another implementation of the neural network to serve as a controller. It has the architecture of the instar-outstar neural network model and uses the hybrid unsupervised and supervised learning scheme. It offers a viable alternative to the two-layer neural network in many applications of signal processing, pattern recognition, control, and function approximation. The structure of RBFN is showed in Fig. 4.5. Unlike the instar-outstar neural network model in which the hidden nodes are linear winner-take-all nodes, the hidden nodes in the RBFN have normalized Gaussian activation function

$$z_q = g_q(\mathbf{x}) = \frac{R_q(\mathbf{x})}{\sum_k R_k(\mathbf{x})} = \frac{\exp[-\frac{|\mathbf{x}-\mathbf{m}_q|^2}{2\sigma_q^2}]}{\sum_k \exp[-\frac{|\mathbf{x}-\mathbf{m}_k|^2}{2\sigma_k^2}]}, \quad (4.9)$$

where \mathbf{x} is the input vector. Thus, hidden node q gives a maximum response to input vectors close to \mathbf{m}_q . Each hidden node q is said to have its own receptive field $R_q(\mathbf{x})$ in the input space, which is a region centered on \mathbf{m}_q with size proportional to σ_q , where \mathbf{m}_q and σ_q are the mean (an m -dimensional vector) and variance of the q th Gaussian function. The Gaussian function is a particular example of radial

basis functions. The output of the RBFN, denoted by y , is simply the weighted sum of the hidden node output, which is given by

$$y = a\left(\sum_{q=1}^l w_q z_q + \theta\right), \quad (4.10)$$

where $a(\cdot)$ is the output activation function and θ is the threshold value. Generally, $a(\cdot)$ is an identity function (i.e., the output node is a linear unit) and $\theta = 0$.

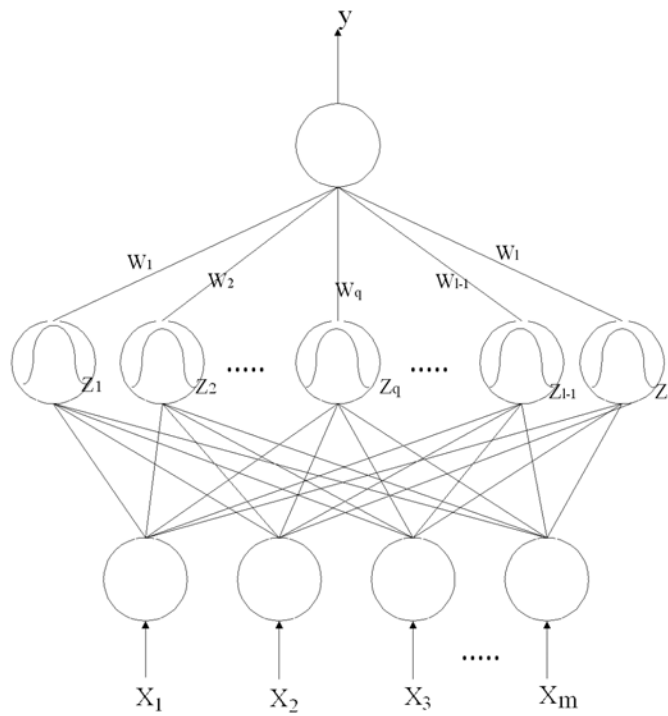


Figure 4.5: The structure of RBFN controller

The purpose of the RBFN is to pave the input space with overlapping receptive fields. For an input vector \mathbf{x} lying somewhere in the input space, the receptive fields with centers close to it will be appreciably activated. The output of the RBFN is then the weighted sum of the activation of these receptive fields.

The training rule of RBFN is hybrid. It includes unsupervised learning in the input layer and supervised learning in the output layer. The unsupervised part of the learning involves the determination of the receptive field centers \mathbf{m}_q and widths σ_q , $q = 1, 2, \dots, l$. The proper centers \mathbf{m}_q can be found by unsupervised learning rules such as the vector quantization approach, competitive learning rules, or simply the Kohonen learning rule; that is

$$\Delta \mathbf{m}_{\text{closest}} = \eta(\mathbf{x} - \mathbf{m}_{\text{closest}}), \quad (4.11)$$

where $\mathbf{m}_{\text{closest}}$ is the center of the receptive field closest to the input vector x and the other centers are kept unchanged. In simple case, the widths σ_q are determined by

$$\sigma_q = \frac{|\mathbf{m}_q - \mathbf{m}_{\text{closest}}|}{\gamma}, \quad (4.12)$$

where $\mathbf{m}_{\text{closest}}$ is the closest vector to \mathbf{m}_q and γ is an overlap parameter.

According to the delta learning rule, the weights in the output layer can be updated by

$$\Delta w_q = \eta(d - y)z_q. \quad (4.13)$$

When averaged over the p training pairs, the objective is to minimize the following mean squared error cost function:

$$E(w_q) = \frac{1}{2} \sum_k [d^k - y^k]^2 \quad (4.14)$$

$$= \frac{1}{2} \sum_k [d^k - \sum_{q=1}^l w_q z_q^k]^2 \quad (4.15)$$

$$= \frac{1}{2} \sum_k [d^k - \sum_{q=1}^l w_q g_q(\mathbf{x}^k)]^2. \quad (4.16)$$

Although RBFN generally cannot quite achieve the same accuracy as the multilayer feedforward neural network, it can be trained several orders of magnitude faster than the the multilayer feedforward neural network with back-propagation learning. This is due to the advantage of hybrid-learning networks which have only one layer of connections trained by supervised learning. It is suitable for the application where the neural network controller is necessary to be on-line trained to adaptively capture the dynamic features of a system.

4.4 Intelligent Inter-Ring Route Controller

The intelligent inter-ring route controller (IIRC) is to determine a proper ringlet (CW or CCW) for an incoming inter-ring new call request at bridge. The determination of ringlet is based on the load balancing principle, in which the CW or CCW ringlet with lower congestion degree and higher service rate will be chosen. The congestion may come from the bridge node or the CW (CCW) downstream node. The former is related with the two STQ lengths of the associated interface in the bridge node given in Figs. 4.1 and 4.2. Thus as shown in Fig. 4.6, the IIRC designs a *fuzzy bridge-node congestion indicator* (FBCI) to intelligently detect this congestion. The latter is related with the received fairRate from the downstream node of the associated ringlet. Therefore, the IIRC designs a *PRNN (pipeline recurrent neural networks) downstream-node fairness predictor* (PDFP) to predict the CW or CCW downstream-node congestion degree. Finally, the IIRC designs a *fuzzy route controller* (FRC) to determine a proper ringlet for the incoming inter-ring new call request. It receives the congestion indication from FBCI, denoted by C_I , and the predicted mean received fairRate from PDFP, denoted by $\widehat{\mathfrak{R}}_f$, as input lin-

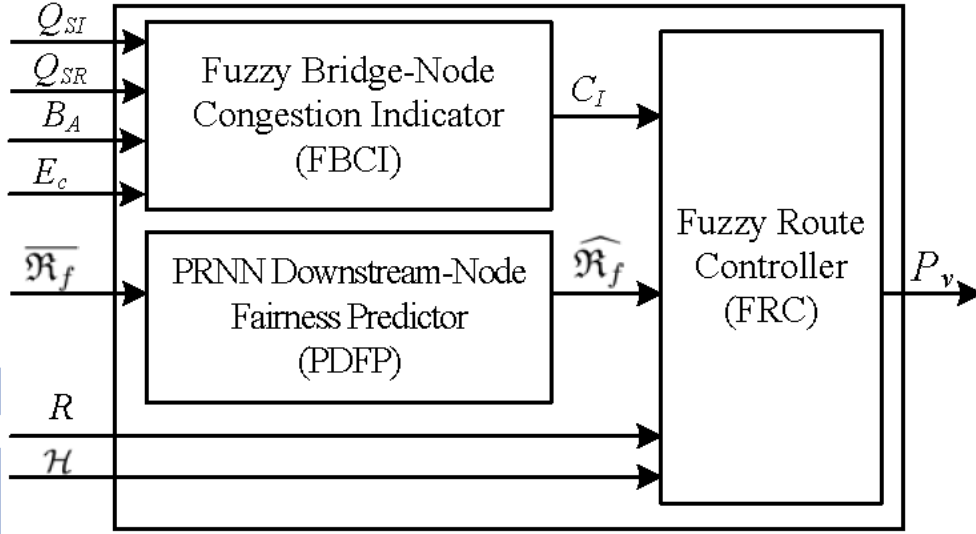


Figure 4.6: Intelligent inter-ring route controller (IIRC)

guistic variables. Also, it considers the service rate of the CW or CCW ringlet at the bridge node, denoted by R , and the number of hops between the bridge and the destination, denoted by \mathcal{H} , as input linguistic variables. Notice that the ringlet service rate at the bridge node, defined in Section 4.2.2, is related with the received fairRate and more hops consume more system bandwidth. The FRC calculates the preference value of route, denoted by P_v , for CW and CCW interfaces and selects the ringlet with larger P_v as the proper ringlet route for the incoming inter-ring new call request.

4.4.1 Fuzzy Bridge-Node Congestion Indicator (FBCI)

The fuzzy bridge-node congestion indicator (FBCI) considers four measures as the input linguistic variables to determine the congestion degree of the bridge node at the CW or CCW interface. They are STQ lengths in the ingress buffer and the ringlet buffer, denoted by Q_{SI} and Q_{SR} , respectively, the amount of the reserved

bandwidth for high class traffic (which are stored in PTQ), denoted by B_A , and the equivalent capacity of the incoming inter-ring new call [59], denoted by E_c . Note that the equivalent capacity for a new call can be estimated from its traffic description parameters: the peak rate, mean rate, and peak rate duration of packets [53, 59]. Among the four measures, the two STQ lengths are the more essential measures to indicate the degree of the congestion in the RPR bridge node. The B_A occupancy is highly correlated with the STQ due to the fact that the system bandwidth is allocated to high priority traffic first. Also, the amount of E_c can cause the increment of the STQ length. The output linguistic variable of the FBCI is the congestion degree of the CW or CCW interface of the bridge, denoted by C_I . Term sets for the four input linguistic variables and the output linguistic variable are defined as $T(Q_{SI}(Q_{SR})) = \{\text{Short } (S), \text{Medium } (M), \text{Long } (L)\}$; $T(B_A) = \{\text{Few } (F_w), \text{Many } (M_a)\}$; $T(E_c) = \{\text{Small } (S), \text{Large } (L)\}$, and $T(C_I) = \{\text{Very Low } (VL), \text{Low } (L), \text{Medium } (M), \text{High } (H), \text{Very High } (VH)\}$.

Membership functions for each term T in the term set of input/output linguistic variable X , denoted by $\mu_T(X)$, should be defined with a proper shape and position. The determination for the membership function is subjective in nature; however, it cannot be selected arbitrarily [43]. Usually, a triangular function $f(x; x_0, a_0, a_1)$ or a trapezoidal function $g(x; x_0, x_1, a_0, a_1)$ is chosen as membership function because they are *simple* and thus *suitable* for real-time operation [53, 54]. The two functions are given by

$$f(x; x_0, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1, & \text{for } x_0 - a_0 < x \leq x_0, \\ \frac{x_0-x}{a_1} + 1, & \text{for } x_0 < x < x_0 + a_1, \\ 0, & \text{otherwise,} \end{cases} \quad (4.17)$$

and

$$g(x; x_0, x_1, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1, & \text{for } x_0 - a_0 < x \leq x_0, \\ 1, & \text{for } x_0 < x \leq x_1, \\ \frac{x_1-x}{a_1} + 1, & \text{for } x_1 < x < x_1 + a_1, \\ 0, & \text{otherwise,} \end{cases} \quad (4.18)$$

where x_0 in $f(\cdot)$ is the center of the triangular function; x_0 (x_1) in $g(\cdot)$ is the left (right) edge of the trapezoidal function; and a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function. The center, edge, or width of the triangular or trapezoidal membership function is set intuitively but based on the characteristics of the linguistic variables.

Membership functions for S , M and L in $T(Q_{SI})$ are expressed as

$$\mu_S(Q_{SI}) = g\left(\frac{Q_{SI}}{L_s}; 0, 0.25l_{th}, 0, 0.25l_{th}\right), \quad (4.19)$$

$$\mu_M(Q_{SI}) = f\left(\frac{Q_{SI}}{L_s}; 0.5l_{th}, 0.5l_{th}, 0.5l_{th}\right), \quad (4.20)$$

$$\mu_L(Q_{SI}) = g\left(\frac{Q_{SI}}{L_s}; l_{th}, 1, 0.5l_{th}, 0\right), \quad (4.21)$$

where L_s is the STQ queue size, and l_{th} is the threshold in percentage. Note that if the STQ length is larger than $l_{th} \cdot L_s$, the bridge is in congestion and the fairness algorithm will be enabled. Membership functions for S , M and L in $T(Q_{SR})$ are similar to S , M and L in $T(Q_{SI})$, respectively. Membership functions for F_w and M_a in $T(B_A)$ are defined as

$$\mu_{F_w}(B_A) = g(B_A; 0, C/40, 0, C/40), \quad (4.22)$$

$$\mu_{M_a}(B_A) = g(B_A; C/10, C, 3C/40, 0), \quad (4.23)$$

where C is the link capacity. Membership functions of terms in $T(E_c)$, are defined as

$$\mu_S(E_c) = g(E_c; 0, R_{voice}, 0, R_{video}), \quad (4.24)$$

Table 4.1: The Fuzzy Rule base of FBCI

Rule	Q_{SI}	Q_{SR}	B_A	E_c	C_I	Rule	Q_{SI}	Q_{SR}	B_A	E_c	C_I
1	S	S	X	X	VL	13	M	M	M_a	S	M
2	S	M	F_w	X	VL	14	M	M	M_a	L	H
3	S	M	M_a	S	VL	15	M	L	F_w	X	H
4	S	M	M_a	L	L	16	M	L	M_a	S	H
5	S	L	F_w	S	L	17	M	L	M_a	L	VH
6	S	L	F_w	L	M	18	L	S	F_w	X	M
7	S	L	M_a	S	M	19	L	S	M_a	S	M
8	S	L	M_a	L	H	20	L	S	M_a	L	H
9	M	S	F_w	X	VL	21	L	M	F_w	S	H
10	M	S	M_a	X	L	22	L	M	F_w	L	VH
11	M	M	F_w	S	L	23	L	M	M_a	X	VH
12	M	M	F_w	L	L	24	L	L	X	X	VH

$$\mu_L(E_c) = g(E_c; R_h, C, R_{video}, 0), \quad (4.25)$$

where R_{voice} and R_{video} are the minimum demand of the mean rates of the voice and video traffics, respectively, and R_h is the maximum demand of the mean rate of the video traffic to provide the high quality video. Membership functions for terms of output linguistic variable C_I are defined as

$$\mu_{VL}(C_I) = f(C_I; 0.1, 0, 0), \quad (4.26)$$

$$\mu_L(C_I) = f(C_I; 0.3, 0, 0), \quad (4.27)$$

$$\mu_M(C_I) = f(C_I; 0.5, 0, 0), \quad (4.28)$$

$$\mu_H(C_I) = f(C_I; 0.75, 0, 0), \quad (4.29)$$

$$\mu_{VH}(C_I) = f(C_I; 1, 0, 0). \quad (4.30)$$

As shown in Table 4.1, there are 24 fuzzy rules for FBCI, where the notation "X" in this table represents "don't care" of the linguistic variable. The order of significance of the input linguistic variables for the FBCI would be Q_{SI} , Q_{SR} , B_A ,

and E_c in sequence. The bridge will be in high degree of congestion if its two STQ queue lengths are close to or longer than the threshold (the corresponding terms of Q_{SI} and Q_{SR} are Medium or Long).

FBCI adopts the *max-min* method for fuzzy inference. In Tabel 4.1, there are the 4th, 5th, 10th, 11th, and 12th rules which have the same output congestion indication $C_I = L$. The provisional results after applying the *min* operation on rule 4 for inference is

$$w_4 = \min(\mu_S(Q_{SI}), \mu_M(Q_{SR}), \mu_{M_a}(B_A), \mu_L(E_c)). \quad (4.31)$$

$$w_5 = \min(\mu_S(Q_{SI}), \mu_L(Q_{SR}), \mu_{F_w}(B_A), \mu_S(E_c)). \quad (4.32)$$

$$w_{10} = \min(\mu_M(Q_{SI}), \mu_S(Q_{SR}), \mu_{M_a}(B_A), X), \quad (4.33)$$

$$w_{11} = \min(\mu_M(Q_{SI}), \mu_M(Q_{SR}), \mu_{F_w}(B_A), \mu_S(E_c)), \quad (4.34)$$

$$w_{12} = \min(\mu_M(Q_{SI}), \mu_M(Q_{SR}), \mu_{F_w}(B_A), \mu_L(E_c)), \quad (4.35)$$

Similarly, provisional results of w_5 , w_{10} , w_{11} , and w_{12} can be obtained. According to these provisional results, the *max* operator is applied to yield fuzzy inference result of the output indication $C_I = L$, denoted by w_L , and is given by

$$w_L = \max(w_4, w_5, w_{10}, w_{11}, w_{12}). \quad (4.36)$$

The fuzzy inference results of the output indication VL , M , H , and VH , denoted by w_{VL} , w_M , w_H and w_{VH} , respectively, can be obtained by the same way. Finally, the fuzzy inference results of FBCI are defuzzified to become a crisp value. The defuzzification method adopted is the *center of area defuzzification method*. Thus the crisp value of the congestion degree C_I is obtained by

$$C_I = \frac{0.1 \cdot w_{VL} + 0.3 \cdot w_L + 0.5 \cdot w_M + 0.75 \cdot w_H + 1 \cdot w_{VH}}{w_{VL} + w_L + w_M + w_H + w_{VH}}, \quad (4.37)$$

where the coefficient for w_T denotes the center value of the term T 's triangular membership function, and the term T is VL , L , M , H , or VH ; $0 \leq C_I \leq 1$.

4.4.2 PRNN Downstream-Node Fairness Predictor (PDFP)

The bridge uses the received fairRate from associated ringlet of downstream node to discern the congestion degree of the downstream node. If the received fairRate is high, it means that the downstream nodes's STQ can accept more flows and the bridge can raise its service rate. Otherwise, it means that the downstream nodes's STQ is going to be full or has overflowed and the bridge should decrease its service rate. However, by the AM fairness algorithm [17] considered in this paper, the received fairRate would vary. This high variation of the received fairRate would make the bridge not easily detect if its downstream node is in congestion or not. Therefore, we originally choose an average received fairRate over the past m periods from the current n th period, denoted by $\overline{\mathfrak{R}}_f(n)$, as the input variable, where m is the size of the observation window, $m \geq 1$. The $\overline{\mathfrak{R}}_f(n)$ could be appropriate to detect the congestion situation of the downstream nodes during a period and it is expressed by

$$\overline{\mathfrak{R}}_f(n) = \frac{\mathfrak{R}_f(n) + \mathfrak{R}_f(n-1) + \cdots + \mathfrak{R}_f(n-m+1)}{m}, \quad (4.38)$$

where $\mathfrak{R}_f(n)$ is the received fairRate at time n . Also, since the bridge node routes the traffic flows call by call, the next-step mean received fairRate could be more appropriate to determine the route for an accepted new call. Here, a pipeline recurrent neural networks (PRNN) [60] is adopted to design the PRNN downstream-node fairness predictor (PDFP). The fairRate with one-step prediction as a function of p received fairRates and q previously predicted fairRate, denoted by $\widehat{\mathfrak{R}}_f(n+1)$ or $\widehat{\mathfrak{R}}_f$

for convenience, is given by

$$\widehat{\mathfrak{R}}_f(n+1)=H\left(\overline{\mathfrak{R}}_f(n),\dots,\overline{\mathfrak{R}}_f(n-p+1);\widehat{\mathfrak{R}}_f(n),\dots,\widehat{\mathfrak{R}}_f(n-q+1)\right), \quad (4.39)$$

where $\widehat{\mathfrak{R}}_f(i)$ is the previously predicted mean fairRate at i th period, $n-q+1 \leq i \leq n$, and $H(\cdot)$ is an unknown nonlinear function to be determined. The pipeline recurrent neural network (PRNN) prediction is a fast, low-complexity, and non-linear one that can approximate the function $H(\cdot)$ [60, 61, 62].

The architecture of PRNN can be referred to [61, 62]. The incremental change of synaptic weights is according to the steepest decent method. Also, the training of PRNN consists of two stages. During the off-line training phase, the PRNN, fed with the received fairRates, adjusts the synaptic weights recursively until the root mean square error (RMSE) of the desired prediction output is lower than the criteria. During the on-line training phase, the PRNN fairness predictor obtains the fairRate predictions at $(n+1)$ th period, $\widehat{\mathfrak{R}}_f(n+1)$, from the output of the first neuron of the first module, and receives the new fairRate $\overline{\mathfrak{R}}_f(n+1)$; then it adjusts the synaptic weights using the real time recurrent learning (RTRL) algorithm. Due to the on-line learning capability, PDFP can adapt its wights to the current load conditions other than those set in the off-line training phase [61]. If a PRNN contains q modules and M neurons per module, the computational complexity would be $O(qM^4)$. However, when the system is in operation and the PRNN has determined each parameter by learning, the computational complexity is reduced to $O(1)$ [61, 62].

4.4.3 Fuzzy Route Controller (FRC)

The fuzzy route controller (FRC) is to determine the route preference values, P_v s, for both of CW and CCW ringlets. The determination is based on four input

linguistic variables of ringlet: the congestion indication of the bridge node, C_I , the predicted mean received fairRate, $\widehat{\mathfrak{R}}_f$, the current service rate of the ringlet, R , and the number of hops to destination, \mathcal{H} . The higher value P_v of a ringlet means that the ringlet is more suitable to accept the incoming new call request. Term sets for the input and output linguistic variables are defined as $T(C_I) = \{\text{Low } (L_o), \text{Medium } (M_e), \text{High } (H_i)\}$, $T(\widehat{\mathfrak{R}}_f) = \{\text{Small } (S_m), \text{Medium } (M_e), \text{Large } (L_a)\}$, $T(R) = \{\text{Low } (L_o), \text{High } (H_i)\}$, $T(\mathcal{H}) = \{\text{Few } (F_w), \text{Many } (M_a)\}$, and $T(P_v) = \{\text{Unsuitable } (U), \text{Weakly Unsuitable } (WU), \text{Weakly Suitable } (WS), \text{Suitable } (S)\}$.

Membership functions for terms of L_o , M_e , and H_i in $T(C_I)$ are defined as

$$\mu_{L_o}(C_I) = g(C_I; 0, \frac{1}{4}, 0, \frac{1}{4}), \quad (4.40)$$

$$\mu_{M_e}(C_I) = f(C_I; \frac{1}{2}, \frac{1}{4}, \frac{1}{4}), \quad (4.41)$$

$$\mu_{H_i}(C_I) = g(C_I; \frac{3}{4}, 1, \frac{1}{4}, 0). \quad (4.42)$$

Membership functions for terms of S_m , M_e , and L_a in $T(\widehat{\mathfrak{R}}_f)$ are expressed as

$$\mu_{S_m}(\widehat{\mathfrak{R}}_f) = g(\widehat{\mathfrak{R}}_f; 0, \frac{v}{5}, 0, \frac{v}{4}), \quad (4.43)$$

$$\mu_{M_e}(\widehat{\mathfrak{R}}_f) = f(\widehat{\mathfrak{R}}_f; \frac{v}{2}, \frac{v}{4}, \frac{v}{4}), \quad (4.44)$$

$$\mu_{L_a}(\widehat{\mathfrak{R}}_f) = g(\widehat{\mathfrak{R}}_f; \frac{3v}{5}, v, \frac{v}{10}, 0), \quad (4.45)$$

where v denotes the unreserved bandwidth for the low priority traffic at the bridge node and $v = C - B_A$. Here, we set $\frac{v}{5}$, $\frac{v}{2}$, and $\frac{3v}{5}$ as the right edge, center, and left edge of membership functions for terms S_m , M_e , and L_a of $\widehat{\mathfrak{R}}_f$, respectively. Similarly, membership functions for $T(R)$ are defined as

$$\mu_{L_o}(R) = g(R; 0, \frac{1}{4}C, 0, \frac{1}{4}C), \quad (4.46)$$

$$\mu_{H_i}(R) = g(R; \frac{3}{5}C, C, \frac{C}{5}, 0), \quad (4.47)$$

where C is the total capacity of the fiber link. Here, we set $\frac{C}{4}$ and $\frac{3C}{5}$ as the right edge and left edge of membership functions for terms F_w and M_a of R . Membership functions for terms of F_w and M_a in $T(\mathcal{H})$ are defined as

$$\mu_{F_w}(\mathcal{H}) = g(\mathcal{H}; 0, \frac{1}{3}N, 0, \frac{1}{3}N), \quad (4.48)$$

$$\mu_{M_a}(\mathcal{H}) = g(\mathcal{H}; \frac{2}{3}N, N, \frac{1}{3}N, 0), \quad (4.49)$$

where N is the total number of nodes in a RPR network. Finally, membership functions for terms in $T(P_v)$ are defined as

$$\mu_U(P_v) = f(P_v; 0.1, 0, 0), \quad (4.50)$$

$$\mu_{WU}(P_v) = f(P_v; 0.4, 0, 0), \quad (4.51)$$

$$\mu_{US}(P_v) = f(P_v; 0.7, 0, 0), \quad (4.52)$$

$$\mu_S(P_v) = f(P_v; 1, 0, 0). \quad (4.53)$$

As shown in Table 4.2, there are 21 fuzzy rules. The notation "X" in Table 4.2 represents "don't care" of the linguistic variable. The rules are designed according to the load balancing principle for FRC, and the order of significance of the input linguistic variables for the FRC is C_I , $\widehat{\mathfrak{R}}_f$, R , and \mathcal{H} . The low congestion degree of ringlet interface ($C_I = L_o$) and the large or medium predicted mean received fairRate ($\widehat{\mathfrak{R}}_f = L_a$ or M_e) would make the inter-ring new call have more chance to enter the interface. However, the low congestion degree of ringlet interface ($C_I = L_o$), but the small predicted mean received fairRate ($\widehat{\mathfrak{R}}_f = S_m$) which means that the downstream nodes may incur congestion, and the high ringlet service rate ($R = H_i$)

Table 4.2: The Fuzzy Rule Base of FRC

Rule	C_I	$\widehat{\mathfrak{R}}_f$	R	\mathcal{H}	P_v	Rule	C_I	$\widehat{\mathfrak{R}}_f$	R	\mathcal{H}	P_v
1	L_o	L_a	X	X	S	12	M_e	M_e	L_o	M_a	WU
2	L_o	M_e	H_i	X	S	13	M_e	S_m	H_i	X	WU
3	L_o	M_e	L_o	X	S	14	M_e	S_m	L_o	X	U
4	L_o	S_m	H_i	F_w	WS	15	H_i	L_a	H_i	F_w	WS
5	L_o	S_m	H_i	M_a	WU	16	H_i	L_a	H_i	M_a	WU
6	L_o	S_m	L_o	X	WU	17	H_i	L_a	L_o	X	WU
7	M_e	L_a	H_i	X	S	18	H_i	M_e	H_i	F_w	WU
8	M_e	L_a	L_o	F_w	S	19	H_i	M_e	H_i	M_a	U
9	M_e	L_a	L_o	M_a	WS	20	H_i	M_e	L_o	X	U
10	M_e	M_e	H_i	X	WS	21	H_i	S_m	X	X	U
11	M_e	M_e	L_o	F_w	WS						

would make the variable of the number of hops to destination \mathcal{H} significant. If \mathcal{H} is Few, the new call will be weakly suitable for the ringlet, while if \mathcal{H} is Many, the new call will be weakly unsuitable for the ringlet. On the other hand, the high congestion degree of ringlet interface ($C_I = H_i$) and the small predicted mean received fairRate ($\widehat{\mathfrak{R}}_f = S_m$) would make the inter-ring new call have less chance to enter the interface. However, the high congestion degree of ringlet interface ($C_I = H_i$), but the large predicted mean received fairRate ($\widehat{\mathfrak{R}}_f = L_a$) which means that the downstream nodes are free of congestion, and the high ringlet service rate ($R = H_i$) would similarly make the variable of the number of hops to destination \mathcal{H} significant. The fuzzy inference algorithm also adopts the *max-min* inference method, and the defuzzification method, the *center of area* defuzzification method.

4.5 Simulation Results

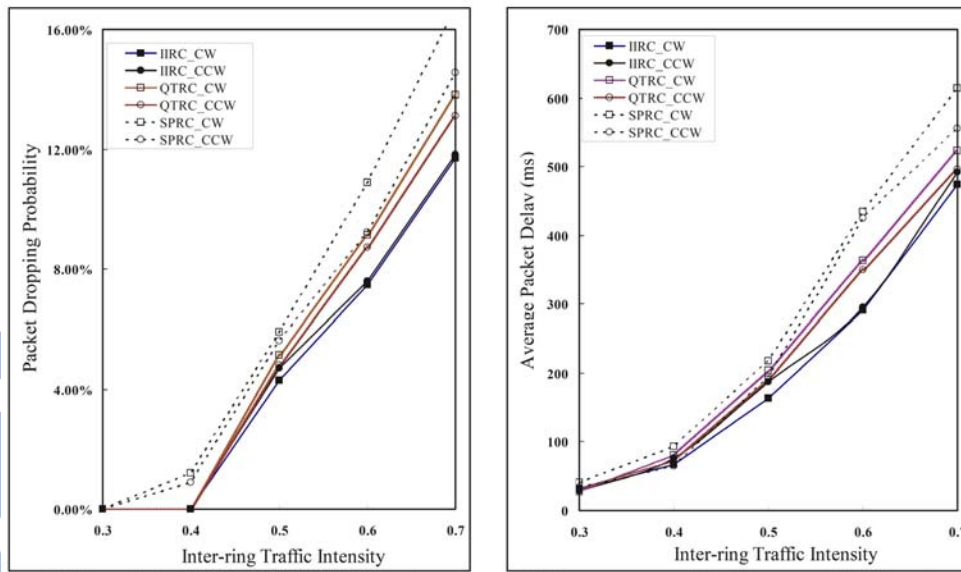
Simulations are here conducted to compare the performances of proposed IIRC, and SPRC [38]. Also, an intuitive queue-length threshold route controller (QTRC)

is included, which determines a proper ringlet depending on the shorter STQ length of ingress buffer. Traffic flows from R_1 to R_0 at the bridge node are considered. Referring to Fig. 4.1, assume that there are $M = 16$ non-bridge nodes on R_0 , the link capacity is $C = 10.0$ Gbps, and sizes of the two PTQs and the two STQs are 40 Mbyte with threshold $l_{th} = 1/4$. Three kinds of calls are considered in the system: voice, video, and data. The two-state Markov chain is used to model packet traffic flow of calls with two different arrival rates and two state transition rates. Then the peak rate R_p , the mean rate R_m , and the mean burst period T_p with the four previous rates can be obtained [59].

For voice packet generation process, during the ON (talkspurt) state, voice packets are generated with rate 21×10^{-4} ; during the OFF (silence) state, no packets are generated. A voice source has two transition rates of 4×10^{-5} and 8×10^{-5} in the ON and OFF states, respectively. The packet size is fixed at 70 bytes, and thus the generation rate is constant bit rate (CBR) during ON state. The arrival process of a voice source was assumed that $R_p = 21 \times 10^{-4}$, $R_m = 7 \times 10^{-4}$, and $T_p = 1.3$ s. Two kinds of video packet generation processes are assumed: the intraframe and interframe generation processes. The intraframe (I-frame) generation process is similar to the voice packet generation process with generating rate 5×10^{-2} , and two transition rates of 4×10^{-5} and 8×10^{-5} in the ON and OFF states, respectively. The arrival process of the I-frame of video packet source was assumed that $R_p = 5 \times 10^{-2}$, $R_m = 1 \times 10^{-2}$, and $T_p = 0.1$ s. The interframe (B- and P-frames) generation process includes B-frame-bit-rate and P-frame-bit-rate video services, and their generation was characterized by Bernoulli processes with rates θ_B and θ_P , respectively. For B-frame-bit-rate of the B-frame of video packet source,

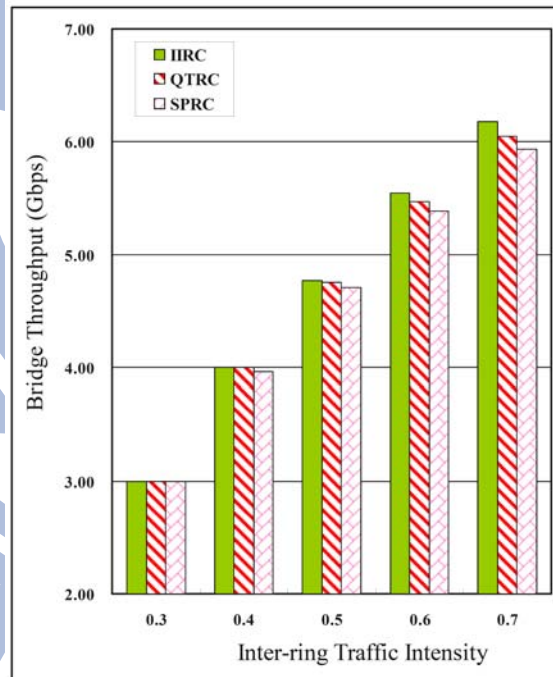
it was assumed that $R_p = 2 \times 10^{-2}$, $R_m = 2 \times 10^{-3}$, and $T_p = 0.01$ s, which is given $\theta_B = 0.1$; for P-frame-bit-rate of the P-frame of video packet source, it was assumed that $R_p = 1 \times 10^{-2}$, $R_m = 2 \times 10^{-4}$, and $T_p = 0.01$ s, which given $\theta_P = 0.02$. The I-frame packet size is fixed at 1000 bytes, and the generation rate is CBR; the B-frame, and P-frame packet sizes are uniformly distributed over 100 and 1518 bytes and the generation rates are with generation of variable bit rate (VBR). The data packet generation process includes high-bit-rate and low-bit-rate data services, and the generation of high-bit-rate data packets and low-bit-rate data packets are characterized by Bernoulli processes with rates θ_1 and θ_2 , respectively. For high-bit-rate of data source, it was assumed that $R_p = 7 \times 10^{-2}$, $R_m = 7 \times 10^{-3}$, and $T_p = 0.03$ s, which is given $\theta_1 = 0.1$; for low-bit-rate of data source, it was assumed that $R_p = 3.5 \times 10^{-2}$, $R_m = 7 \times 10^{-4}$, and $T_p = 0.03$ s, which is given $\theta_2 = 0.02$. The data packet sizes are uniformly distributed over 100 and 1518 bytes and the generation rates are with generation of variable bit rate (VBR). The parameters, R_{voice} , R_{video} , and R_h , are set to 64kbps, 640kbps, and 6.4Mbps, respectively.

Fig. 4.7(a), (b), and (c) show the average packet dropping probability, the average packet delay, and the throughput, respectively, for the proposed IIRC, QTRC, and SPRC, versus the traffic intensity from the R_1 to R_0 at the bridge in a balanced scenario. The traffic intensity at the bridge is here defined as the total arrival packet rate over the capacity of the fiber link. In this balanced scenario, in R_0 , both the local CW ringlet traffic intensity from node 16 to bridge and the local CCW ringlet traffic intensity from node 1 to bridge are fixed at 0.6, and the varying inter-ring traffic intensity is from 0.3 to 0.7; the add traffic intensity of node 1 in CW ringlet and the add traffic intensity of node 16 in CCW ringlet are both fixed at 0.2; the



(a) The packet dropping probability versus the inter-ring traffic intensity

(b) The average packet delay versus the inter-ring traffic intensity

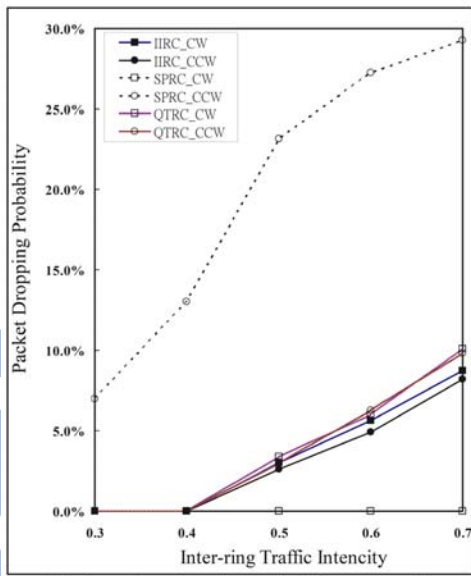


(c) The throughput of the routing traffic versus the inter-ring traffic intensity

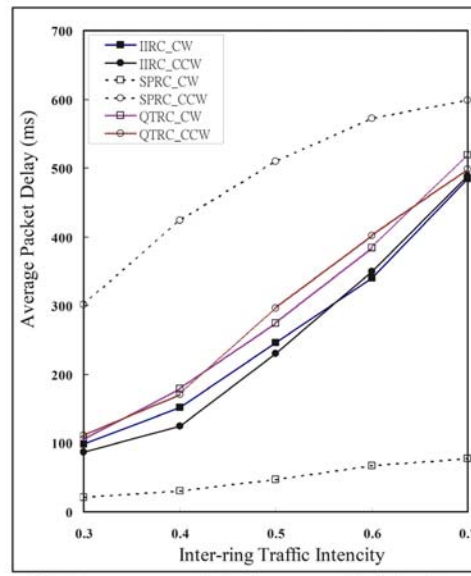
Figure 4.7: The performance comparison for IIRC, SPRC, and QTRC in the balanced scenario

probability of the destination of the incoming new calls is *uniformly distributed* over nodes on R_0 . It is found that the packet dropping probability and the average packet delay of both CW ringlet and CCW ringlet are almost the same for IIRC, QTRC, and SPRC. The results show that IIRC, QTRC and SPRC can achieve the load balancing in this balanced scenario. It is because the probability of the destination for the new call request is uniformly distributed over nodes; the routing policy of QTRC is simply according to a shorter STQ length of ingress buffers and the routing policy of SPRC is based on the shortest path. Also, this justifies that the IIRC, which chooses a suitable ringlet with lower congestion degree and higher service rate, is well designed. Furthermore, IIRC has the lower packet dropping probability by about 16% and 29%, the smaller average packet delay by about 9% and 21%, and the higher throughput by 5.1% and 7% in heavy bridge traffic intensity than QTRC and SPRC, respectively. It is because QTRC does not consider the number of hops to destination, and thus QTRC would route calls to paths with more nodes and then consume more bandwidth. Also, in the situation that many incoming new calls just happen to have the same destinations, SPRC's routing policy would make the STQ overflow. However, IIRC decides a suitable route for each call independently based on congestion degree and service rate.

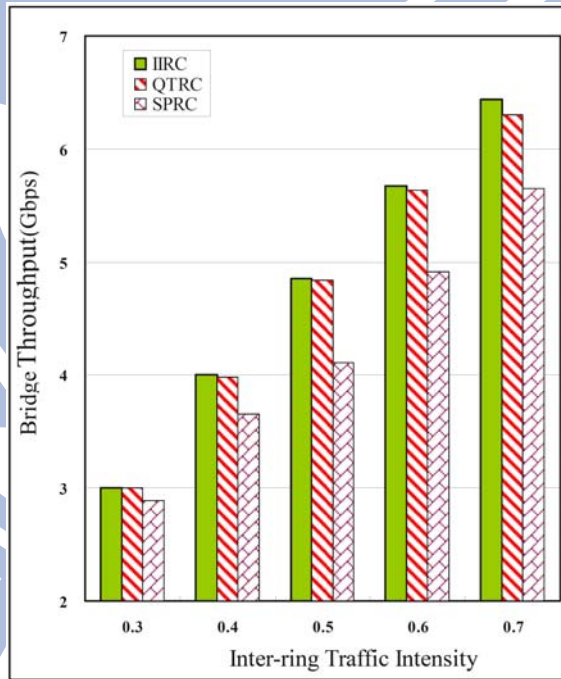
Fig. 4.8(a), (b), and (c) show the average packet dropping probability, the average packet delay, and the throughput, respectively, versus the bridge traffic intensity in an unbalanced scenario. Here, the probability of destination of nodes for new calls is *non-uniformly distributed*, where node 1 (9) to node 8 (16) are with the same probability $1/40$ ($1/10$). It can be found that the packet dropping probabilities and the average packet delays of CW and CCW ringlet by IIRC and



(a) The packet dropping probability versus the inter-ring traffic intensity



(b) The average packet delay versus the inter-ring traffic intensity



(c) The throughput of the routing traffic versus the inter-ring traffic intensity

Figure 4.8: The performance comparison for IIRC, SPRC, and QTRC in the unbalanced scenario

QTRC are still almost the same, while these by SPRC are quite different. We can deduce that the IIRC can indeed perceive the congestion degree of CW and CCW ringlets and sophisticatedly achieve the load balancing by overall considering the congestion degree, the received fairRate, the ringlet service rate, and the number of hops to destination. QTRC could avoid enlarging a longer STQ length of the ingress buffer due to its routing policy. Moreover, IIRC improves by about 10% and 220% in packet dropping probability, and by about 13% and 18% in average packet delay, by about 6% and 19% in throughput in heavy traffic intensity over QTRC and SPRC, respectively. It is because the SPRC scheme would route most calls via the CCW ringlet for most destinations of incoming new calls that are on the up side of the bridge. This will make the STQ occupancy of CCW interface in R_0 exceed a threshold and thus SPRC gets a worse throughput.

Fig. 4.9 shows the bridge throughputs of IIRC and IIRC without prediction in a balanced scenario as given in Fig. 4.7, where different m sizes for the average received fairRate $\overline{\mathfrak{R}}_f$ given in (6) are considered. The IIRC without prediction means that the FRC uses the received fairRat instead of the predicted received fairRate. It can be found that the IIRC achieves higher system throughput than IIRC without prediction by an amount of 7%, generally speaking, and the larger m -size for the average received fairRate would be better. It justifies that the information of the congestion degree of downstream nodes for the determination of proper route for the incoming new call should be timely. The obsolete information of congestion degree is not suitable. A proper number for the average could be more effective. It can be seen that a proper value of m , such as $m = 10$, is good enough. As shown in the Fig. 4.9, the size of $m = 20$ performs better than that of $m = 10$ by only an amount

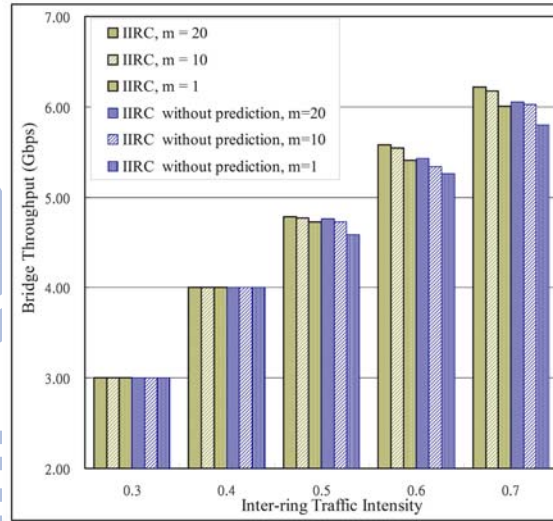


Figure 4.9: The bridge throughput versus the inter-ring traffic intensity under different m size of observing window

of 1.1%.

Fig. 4.10 shows the bridge throughputs under IIRC, IIRC without considering E_C and/or B_A in a balanced scenario as given in Fig. 4.7. It is found that the IIRC has the largest throughput; it improves by about 1.5%, 3.6%, and 6.7% over IIRC without considering E_C , IIRC without considering B_A , and IIRC without considering B_A and E_C , respectively. These can justify that, as the statements made in Section III. A, the input linguistic variables B_A and E_C are essential, and the B_A input linguistic variable is more important than E_C .

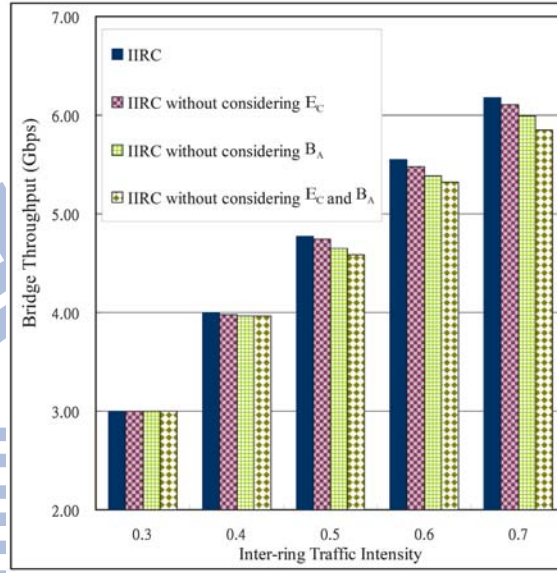


Figure 4.10: The comparison of bridge throughputs under different schemes versus the inter-ring traffic intensity

4.6 Concluding Remarks

This chapter proposes an intelligent inter-ring route controller (IIRC) for bridged resilient packet rings (RPRs). The IIRC uses not only the two STQ lengths but also the reserved bandwidth for highest priority traffic and the equivalent bandwidth of an incoming new call to indicate the congestion degree of the interface of the bridge node. It specially predicts the mean received fairRate to detect the congestion degree of downstream-node. Moreover, IIRC further considers the number of hops to destination and the service rate of the bridge, besides the indication of the congestion degree of bridge-node by FBCI and the prediction of the mean received fairRate by PDFP, to decide a route preference value of the interface by FRC. The rule structure of FRC is based on the load balancing principle. Finally, the IIRC chooses a ringlet with higher preference value of route to forward the call

to the destination. Simulation results show that the IIRC effectively follows the load balancing principle and achieves the better performance than the queue length threshold route controller (QTRC) and the shortest path route controller (SPRC). If the probability of destination nodes is non-uniformly distributed over all node in a ring, IIRC improves by about 10% and 220% in the packet dropping probability, by about 13% and 18% in the average packet delay, and by about 6% and 19% in the throughput over QTRC and SPRC. Also, IIRC achieves more gain in throughput by about 8% and 6.7% than IIRC itself without considering the prediction of the average received fairRate and without considering the amount of the reserved bandwidth as well as the equivalent capacity for a new call request, respectively. These justify that the IIRC is sophisticatedly configured and well designed in choosing the input linguistic variables, defining membership functions, and designing rule base to determine a proper ringlet for an incoming new call. The design philosophy of IIRC can be applied to any kind of bridged optical packet rings.

Moreover, the IIRC is feasible for real applications for that the computational complexity and the cost of IIRC are simple and effective, respectively. As mentioned previously, the IIRC is designed using fuzzy logic and neural networks and can be implemented in a chip.

Chapter 5

Conclusions and Future Works

The thesis provides a channel scheduling scheme, a bandwidth allocation, and a route control to improve the performance of the optical burst switching network of all the optical backbone (core) network, an optical packet ring network, which is called the resilient packet ring (RPR) and specified in IEEE 802.17 [17] and a bridged optical packet ring network based on the resilient packet ring, which is called bridged resilient packet rings (BRPR), respectively.

In chapter 2, we propose a new channel-scheduling scheme, called priority burst scheduling with FDL assignment (PBS-FA), for the OBS networks when the signal protocol is preemptive prioritized JET (PPJET) [16]. It is due to the fact that the high-priority burst is more important than the low-priority one and the shorter burst is more easily to be rescheduled into the void. Therefore, the PBS-FA scheme allows high-priority bursts to preempt low-priority ones and longer high-priority ones to replace shorter ones. Also, it reschedules those preempted bursts by using FDL assignment. Simulation results reveal that the PBS-FA improves the system throughput by about 3% to 10% 2.4 and reduces the average system dropping probability by about 30% to 45% 2.5 at the traffic load 0.4 to 0.8 over the

PLAUC-VF.

In chapter 3, an effective fuzzy local fairRate generator (FLAG) is proposed for RPR. The FLAG is sophisticatedly composed of three function blocks: an adaptive fairRate calculator (AFC), a fuzzy congestion detector (FCD), and a fuzzy fairRate generator (FFG). The AFC pre-generates a fairRate, which meets RIAS fairness and can diminish the effect of the propagation delay. The FCD softly detects the congestion degree of station, considering STQ queue length and its change rate which is the arriving transit FE traffic flows to STQ. Subsequently, the FFG generates a suitable local fairRate by intelligently fine-tuning the pre-generated fairRate, using fuzzy logics, based on the congestion degree of the station. The FLAG can make traffic flows satisfy RIAS fairness criterion and converge to an ideal fairRate in an efficient way. Simulation results show that each flow by FLAG is indeed close to the designated rate with the smallest damping amplitude and the least convergence time in the parking lot scenarios and the available bandwidth reclaiming scenario, compared to conventional AM, DBA, and DBA fairness algorithms, shown in 3.5. These prove that the configuration of FLAG is indeed sophisticated, where AFC pre-generates the local fairRate using the moving average technique; FCD determines the congestion degree of station using fuzzy logics, considering not only the STQ length but also change rate of STQ length; and finally the FFG adopts the fuzzy logics and the expert's domain knowledge to precisely generate the local fairRate by fine-tuning the pre-generated local fairRate by AFC according to the congestion degree by FCD. Also, the performance superiority of DMA over DBA proves that the moving average technique is indeed effective to diminish the effect of propagation delay on the stability of traffic flows.

In chapter 4, we propose an intelligent inter-ring route controller (IIRC) for BRPR. The IIRC uses not only the two STQ lengths but also the reserved bandwidth for highest priority traffic and the equivalent bandwidth of an incoming new call to indicate the congestion degree of the interface of the bridge node. It specially predicts the mean received fairRate to detect the congestion degree of downstream-node. Moreover, IIRC further considers the number of hops to destination and the service rate of the bridge, besides the indication of the congestion degree of bridge-node by FBCI and the prediction of the mean received fairRate by PDFP, to decide a route preference value of the interface by FRC. The rule structure of FRC is based on the load balancing principle. Finally, the IIRC chooses a ringlet with higher preference value of route to forward the call to the destination. Simulation results show that the IIRC effectively follows the load balancing principle and achieves the better performance than the queue length threshold route controller (QTRC) and the shortest path route controller (SPRC). If the probability of destination nodes is non-uniformly distributed over all nodes in a ring, IIRC improves by about 10% and 220% in the packet dropping probability, by about 13% and about 18% in the average packet delay, and by about 6% and 19% in the throughput over QTRC and SPRC 4.8. Also, IIRC achieves more gain in throughput by about 8% and 6.7% than IIRC itself but without considering the prediction of the average received fairRate and the amount of the reserved bandwidth as well as the equivalent capacity for a new call request, respectively. These justify that the IIRC is sophisticatedly configured and well-designed in choosing the input linguistic variables, defining membership functions, and designing rule base to determine a proper ringlet for an incoming new call. The design philosophy of IIRC can be applied to any kind of bridged

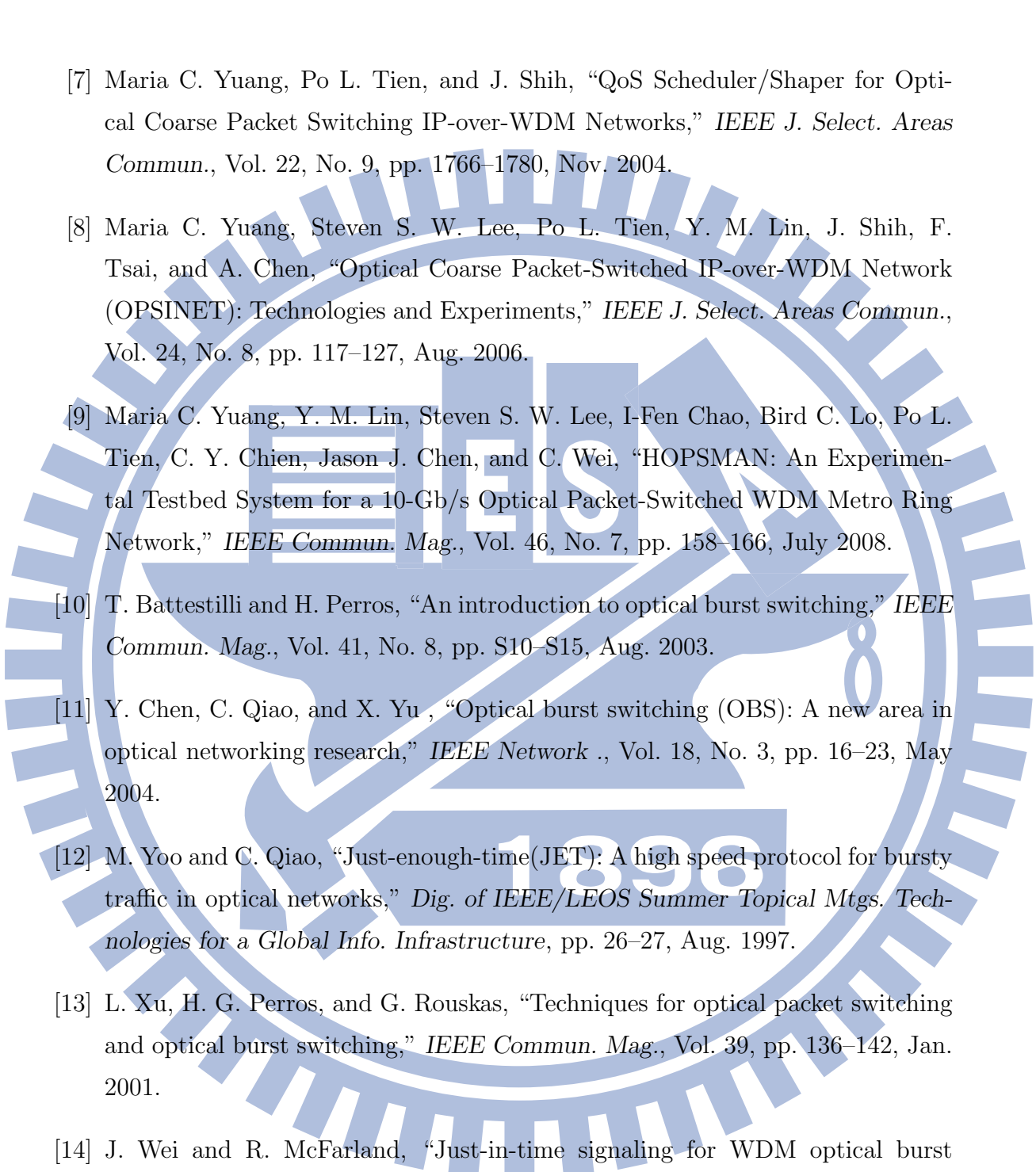
optical packet rings.

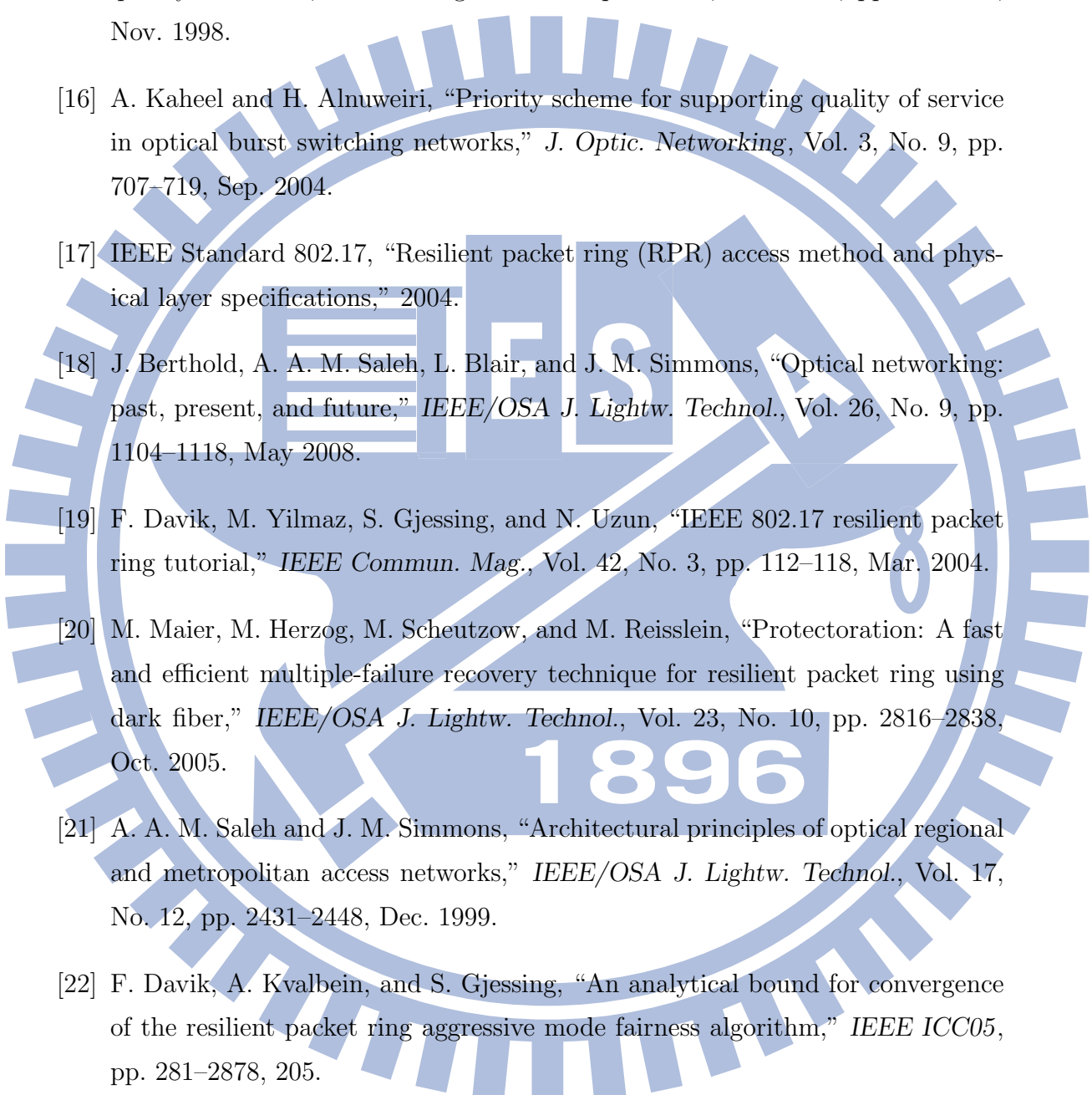
Moreover, the IIRC is feasible for real applications for that the computational complexity and the cost of IIRC are simple and effective, respectively. As mentioned previously, the IIRC is designed using fuzzy logic and neural networks and can be implemented in a chip.

Finally, in the bridged RPR, there are two critically accompanied issues. First, congestion easily happens at bridge for inter-ring traffic whose source and destination nodes are on different rings. Second, there is no mechanism which can guarantee global fairness for inter-ring traffic while obeying local fairness. Consequently, it is possible to have packet loss at bridge and unfair bandwidth allocation for inter-ring traffic. In the future work, we could design a global fairness algorithm for the inter-ring traffic in the BRPR based on the ideal of the proposed fuzzy local fairRate generator (FLAG) under the assumption that the inter-ring traffic is routed by the intelligent inter-ring route controller (IIRC).

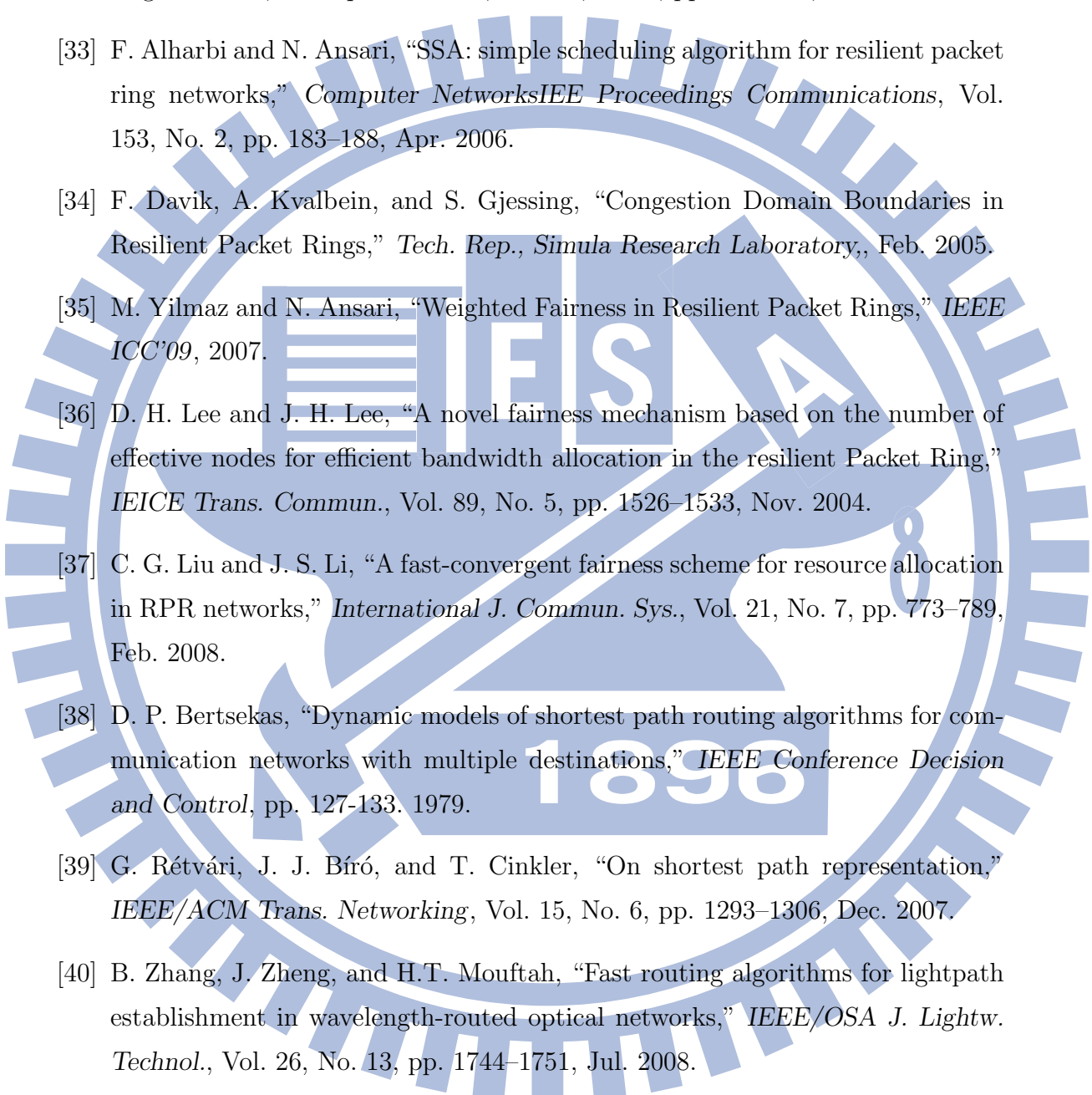
Bibliography

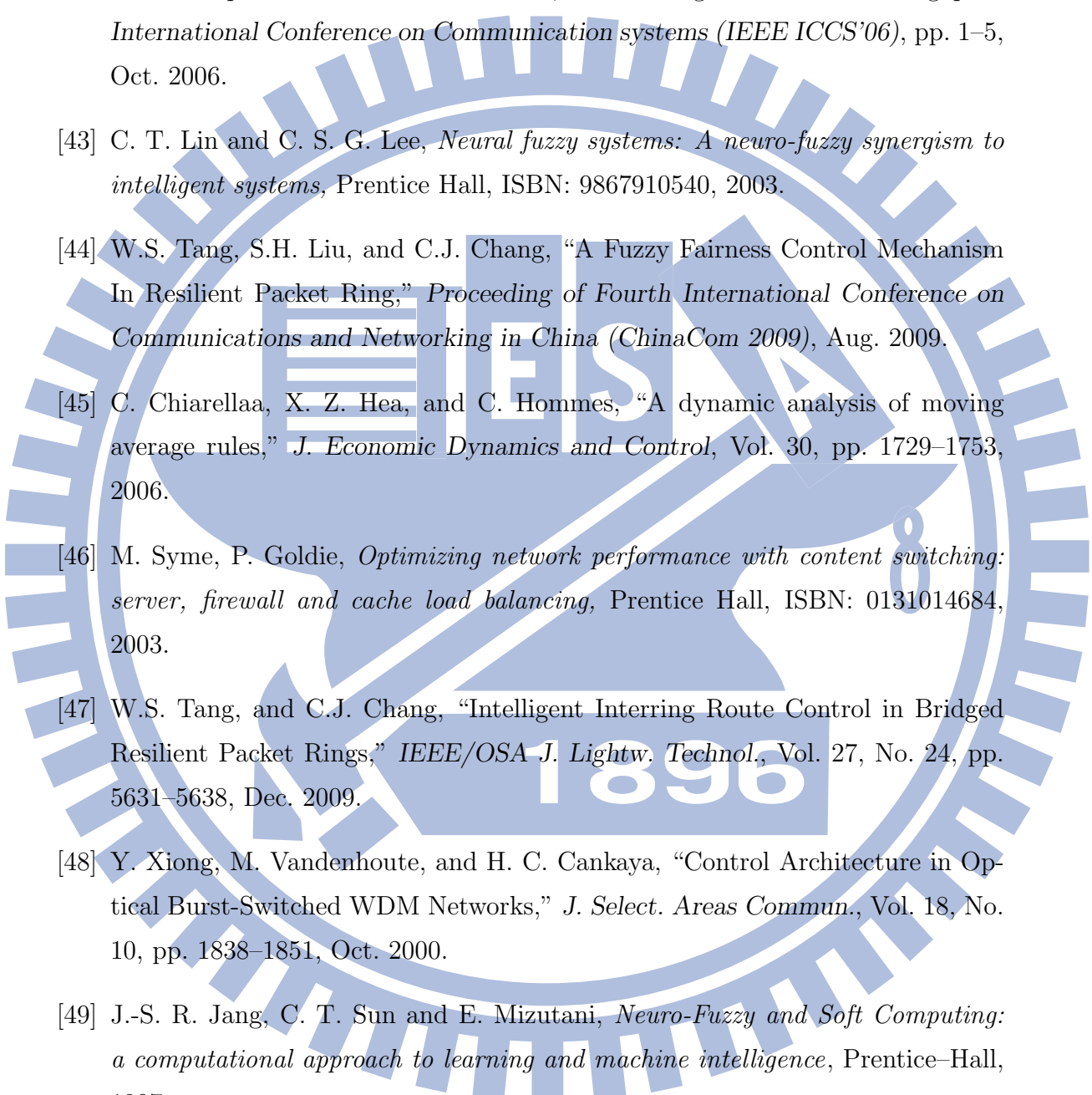
- [1] B. Mukherjee, “WDM optical communication networks: progress and challenges,” *IEEE J. Select. Areas Commun.*, Vol. 18, No. 10, pp. 1810–1824, Oct. 2000.
- [2] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, “Optical switching: Switch fabrics, techniques, and architectures,” *IEEE/OSA J. Lightw. Technol.*, Vol. 21, No. 2, pp. 384–405, Feb. 2003.
- [3] C. Qiao and M. Yoo, “Optical burst switching (OBS) X A new paradigm for an optical internet,” *J. High Speed Networks.*, Vol. 8, No. 1, pp. 69–84, Jan. 1999.
- [4] C. Qiao and M. Yoo, “Labeled optical burst switching for IP-over-WDM integration,” *IEEE Commun. Mag.*, Vol. 38, No. 9, pp. 104–114, Sept. 2000.
- [5] T. El-Bawab and J.-D. Shin, “Optical packet switching in core networks: between vision and reality,” *IEEE Commun. Mag.*, Vol. 40, No. 9, pp. 60–65, Sept. 2002.
- [6] W.S. Tang, C.Y. Huang, C.J. Chang, and F.C. Ren, “QoS-promoted dynamic bandwidth allocation for Ethernet passive optical networks,” *Proceeding of Third International Conference on Communications and Networking in China (ChinaCom 2008)*, pp. 202–256, Aug. 2008.

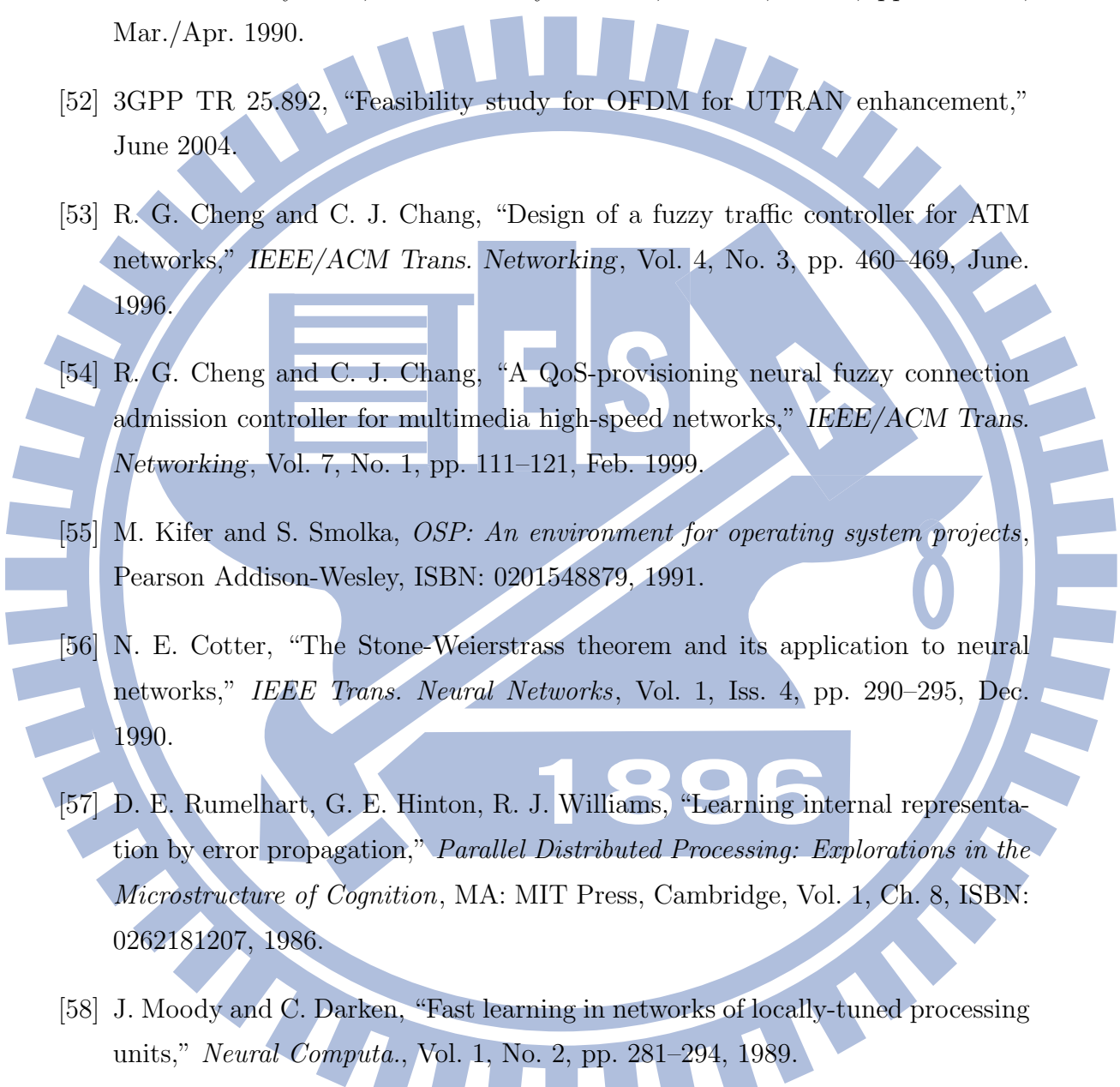
- 
- [7] Maria C. Yuang, Po L. Tien, and J. Shih, “QoS Scheduler/Shaper for Optical Coarse Packet Switching IP-over-WDM Networks,” *IEEE J. Select. Areas Commun.*, Vol. 22, No. 9, pp. 1766–1780, Nov. 2004.
- [8] Maria C. Yuang, Steven S. W. Lee, Po L. Tien, Y. M. Lin, J. Shih, F. Tsai, and A. Chen, “Optical Coarse Packet-Switched IP-over-WDM Network (OPSINET): Technologies and Experiments,” *IEEE J. Select. Areas Commun.*, Vol. 24, No. 8, pp. 117–127, Aug. 2006.
- [9] Maria C. Yuang, Y. M. Lin, Steven S. W. Lee, I-Fen Chao, Bird C. Lo, Po L. Tien, C. Y. Chien, Jason J. Chen, and C. Wei, “HOPSMAN: An Experimental Testbed System for a 10-Gb/s Optical Packet-Switched WDM Metro Ring Network,” *IEEE Commun. Mag.*, Vol. 46, No. 7, pp. 158–166, July 2008.
- [10] T. Battestilli and H. Perros, “An introduction to optical burst switching,” *IEEE Commun. Mag.*, Vol. 41, No. 8, pp. S10–S15, Aug. 2003.
- [11] Y. Chen, C. Qiao, and X. Yu, “Optical burst switching (OBS): A new area in optical networking research,” *IEEE Network*, Vol. 18, No. 3, pp. 16–23, May 2004.
- [12] M. Yoo and C. Qiao, “Just-enough-time(JET): A high speed protocol for bursty traffic in optical networks,” *Dig. of IEEE/LEOS Summer Topical Mtgs. Technologies for a Global Info. Infrastructure*, pp. 26–27, Aug. 1997.
- [13] L. Xu, H. G. Perros, and G. Rouskas, “Techniques for optical packet switching and optical burst switching,” *IEEE Commun. Mag.*, Vol. 39, pp. 136–142, Jan. 2001.
- [14] J. Wei and R. McFarland, “Just-in-time signaling for WDM optical burst switching networks,” *J. Lightwave Technol.*, Vol. 18, No. 12, pp. 2019–2037, Dec. 2000.

- 
- [15] M. Yoo and C. Qiao, "A new optical burst switching protocol for supporting quality of service," *Proceeding of SPIE Opticomm.*, Vol. 3531, pp. 396–405, Nov. 1998.
- [16] A. Kaheel and H. Alnuweiri, "Priority scheme for supporting quality of service in optical burst switching networks," *J. Optic. Networking*, Vol. 3, No. 9, pp. 707–719, Sep. 2004.
- [17] IEEE Standard 802.17, "Resilient packet ring (RPR) access method and physical layer specifications," 2004.
- [18] J. Berthold, A. A. M. Saleh, L. Blair, and J. M. Simmons, "Optical networking: past, present, and future," *IEEE/OSA J. Lightw. Technol.*, Vol. 26, No. 9, pp. 1104–1118, May 2008.
- [19] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun, "IEEE 802.17 resilient packet ring tutorial," *IEEE Commun. Mag.*, Vol. 42, No. 3, pp. 112–118, Mar. 2004.
- [20] M. Maier, M. Herzog, M. Scheutzow, and M. Reisslein, "Protection: A fast and efficient multiple-failure recovery technique for resilient packet ring using dark fiber," *IEEE/OSA J. Lightw. Technol.*, Vol. 23, No. 10, pp. 2816–2838, Oct. 2005.
- [21] A. A. M. Saleh and J. M. Simmons, "Architectural principles of optical regional and metropolitan access networks," *IEEE/OSA J. Lightw. Technol.*, Vol. 17, No. 12, pp. 2431–2448, Dec. 1999.
- [22] F. Davik, A. Kvalbein, and S. Gjessing, "An analytical bound for convergence of the resilient packet ring aggressive mode fairness algorithm," *IEEE ICC05*, pp. 281–2878, 2005.
- [23] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, "Design, analysis, and implementation of DVSR: a fair high-performance protocol

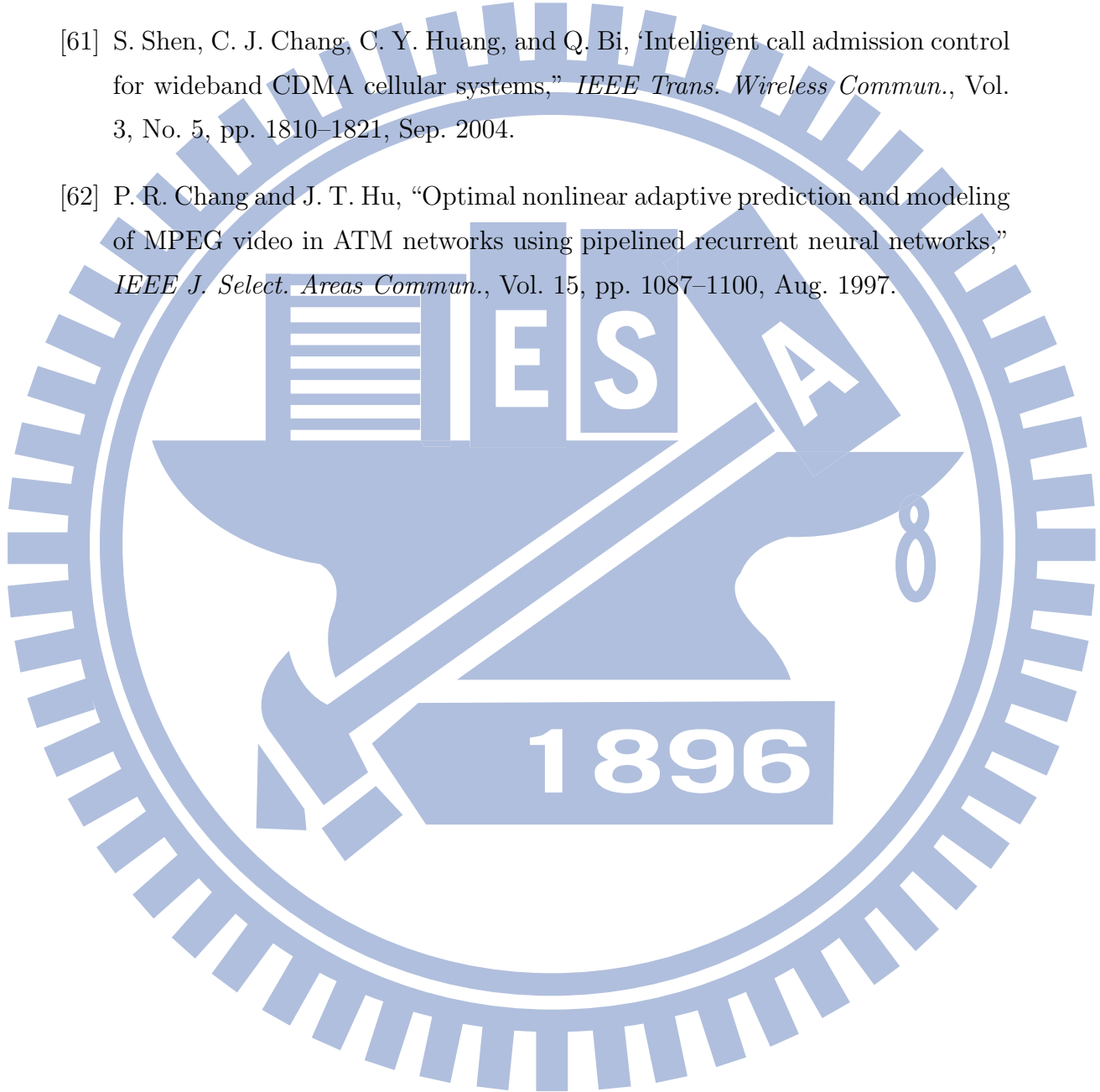
- for packet rings,” *IEEE/ACM Trans. Networking*, Vol. 12, No. 1, pp. 85–102, Feb. 2004.
- [24] F. Alharbi and N. Ansari, “Low complexity distributed bandwidth allocation for resilient packet ring networks,” *HPSR2004*, pp. 277–281, 2004.
- [25] A. Shokrani, J. Talim, and I. Lambadaris, “Modeling and analysis of fair rate calculation in resilient packet ring conservative mode,” *IEEE ICC’06*, pp. 203–210, 2006.
- [26] M. Yilmaz, N. Ansari, J. H. Kao, and P. Yilmaz, “Active queue management for MAC client implementation of resilient packet rings,” *IEEE ICC’09*, 2009.
- [27] P. Setthawong and S. Tanterdtid, “Efficient traffic management for bridged resilient packet rings using topology discovery and spanning tree algorithm,” *IASTED International Conference on Networks and Communication Systems*, pp. 392–397, 2005.
- [28] Z. Zhang, F. Cheng, J. Luo¹, Q. Zeng, and M. Jiang, etc., “Interconnected resilient packet rings (IRPRs): design and implementation,” *Phot. Network Commun.*, Vol. 12, No. 2, pp. 181–193, Sep. 2006.
- [29] J. S. Turner, “Terabit burst switching,” *J. High Speed Networks*, Vol. 8, pp. 3–16, 1999.
- [30] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, “Control architecture in optical burst-switched WDM networks,” *J. Select. Areas Commun.*, Vol. 18, No. 10, pp. 1833–1851, Oct. 2000.
- [31] J. Xu, C. Qiao, J. Li, and G. Xu, “Efficient burst scheduling algorithms in optical burst-switched networks using geometric techniques,” *J. Select. Areas Commun.*, Vol. 22, No. 9, pp. 1796–1811, Nov. 2004.

- 
- [32] F. Alharbi and N. Ansari, "Distributed bandwidth allocation for resilient packet ring networks," *Comp. Networks*, Vol. 49, No. 2, pp. 161–171, Oct. 2005.
- [33] F. Alharbi and N. Ansari, "SSA: simple scheduling algorithm for resilient packet ring networks," *Computer Networks IEE Proceedings Communications*, Vol. 153, No. 2, pp. 183–188, Apr. 2006.
- [34] F. Davik, A. Kvalbein, and S. Gjessing, "Congestion Domain Boundaries in Resilient Packet Rings," *Tech. Rep., Simula Research Laboratory*, Feb. 2005.
- [35] M. Yilmaz and N. Ansari, "Weighted Fairness in Resilient Packet Rings," *IEEE ICC'09*, 2007.
- [36] D. H. Lee and J. H. Lee, "A novel fairness mechanism based on the number of effective nodes for efficient bandwidth allocation in the resilient Packet Ring," *IEICE Trans. Commun.*, Vol. 89, No. 5, pp. 1526–1533, Nov. 2004.
- [37] C. G. Liu and J. S. Li, "A fast-convergent fairness scheme for resource allocation in RPR networks," *International J. Commun. Sys.*, Vol. 21, No. 7, pp. 773–789, Feb. 2008.
- [38] D. P. Bertsekas, "Dynamic models of shortest path routing algorithms for communication networks with multiple destinations," *IEEE Conference Decision and Control*, pp. 127–133. 1979.
- [39] G. Rétvári, J. J. Bíró, and T. Cinkler, "On shortest path representation," *IEEE/ACM Trans. Networking*, Vol. 15, No. 6, pp. 1293–1306, Dec. 2007.
- [40] B. Zhang, J. Zheng, and H.T. Mouftah, "Fast routing algorithms for lightpath establishment in wavelength-routed optical networks," *IEEE/OSA J. Lightw. Technol.*, Vol. 26, No. 13, pp. 1744–1751, Jul. 2008.
- [41] M. Heiden, M. Sortais, and M. Scheutzow, etc., "Multicast capacity of optical packet ring for hotspot traffic," *IEEE/OSA J. Lightw. Technol.*, Vol. 25, No. 9, pp. 2638–2652, Sep. 2007.

- 
- [42] W.S. Tang, and C.J. Chang, “Burst Priority Scheduling with FDL Reassignment in Optical Burst Switch Network,” *Proceeding of Tenth IEEE Singapore International Conference on Communication systems (IEEE ICCS’06)*, pp. 1–5, Oct. 2006.
- [43] C. T. Lin and C. S. G. Lee, *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*, Prentice Hall, ISBN: 9867910540, 2003.
- [44] W.S. Tang, S.H. Liu, and C.J. Chang, “A Fuzzy Fairness Control Mechanism In Resilient Packet Ring,” *Proceeding of Fourth International Conference on Communications and Networking in China (ChinaCom 2009)*, Aug. 2009.
- [45] C. Chiarellaa, X. Z. Hea, and C. Hommes, “A dynamic analysis of moving average rules,” *J. Economic Dynamics and Control*, Vol. 30, pp. 1729–1753, 2006.
- [46] M. Syme, P. Goldie, *Optimizing network performance with content switching: server, firewall and cache load balancing*, Prentice Hall, ISBN: 0131014684, 2003.
- [47] W.S. Tang, and C.J. Chang, “Intelligent Interring Route Control in Bridged Resilient Packet Rings,” *IEEE/OSA J. Lightw. Technol.*, Vol. 27, No. 24, pp. 5631–5638, Dec. 2009.
- [48] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, “Control Architecture in Optical Burst-Switched WDM Networks,” *J. Select. Areas Commun.*, Vol. 18, No. 10, pp. 1838–1851, Oct. 2000.
- [49] J.-S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*, Prentice-Hall, 1997.
- [50] H.-J. Zimmermann, *Fuzzy set theory and its applications*, Kluwer Academic Publishers, ISBN: 0792396243, 1996.

- 
- [51] C. C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller – Part I,” *IEEE Tran. Systems, Man And Cybernetics*, Vol. 20, No. 2, pp. 419–435, Mar./Apr. 1990.
- [52] 3GPP TR 25.892, “Feasibility study for OFDM for UTRAN enhancement,” June 2004.
- [53] R. G. Cheng and C. J. Chang, “Design of a fuzzy traffic controller for ATM networks,” *IEEE/ACM Trans. Networking*, Vol. 4, No. 3, pp. 460–469, June 1996.
- [54] R. G. Cheng and C. J. Chang, “A QoS-provisioning neural fuzzy connection admission controller for multimedia high-speed networks,” *IEEE/ACM Trans. Networking*, Vol. 7, No. 1, pp. 111–121, Feb. 1999.
- [55] M. Kifer and S. Smolka, *OSP: An environment for operating system projects*, Pearson Addison-Wesley, ISBN: 0201548879, 1991.
- [56] N. E. Cotter, “The Stone-Weierstrass theorem and its application to neural networks,” *IEEE Trans. Neural Networks*, Vol. 1, Iss. 4, pp. 290–295, Dec. 1990.
- [57] D. E. Rumelhart, G. E. Hinton, R. J. Williams, “Learning internal representation by error propagation,” *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MA: MIT Press, Cambridge, Vol. 1, Ch. 8, ISBN: 0262181207, 1986.
- [58] J. Moody and C. Darken, “Fast learning in networks of locally-tuned processing units,” *Neural Computa.*, Vol. 1, No. 2, pp. 281–294, 1989.
- [59] R. Gúerin, H. Ahmadi, and M. Naghshineh, “Equivalent capacity and its application to bandwidth allocation in high-speed networks,” *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 968–981, Sep. 1991.

- [60] S. Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Trans. Signal Process.*, Vol. 43, No. 2, pp. 526–535, Feb. 1995.
- [61] S. Shen, C. J. Chang, C. Y. Huang, and Q. Bi, "Intelligent call admission control for wideband CDMA cellular systems," *IEEE Trans. Wireless Commun.*, Vol. 3, No. 5, pp. 1810–1821, Sep. 2004.
- [62] P. R. Chang and J. T. Hu, "Optimal nonlinear adaptive prediction and modeling of MPEG video in ATM networks using pipelined recurrent neural networks," *IEEE J. Select. Areas Commun.*, Vol. 15, pp. 1087–1100, Aug. 1997.



Vita

Wen-Shiang Tang was born in Tainan, Taiwan, R.O.C., on September 12, 1978. He received B.E. degree in mathematics, National Kaohsiung Normal University, Taiwan, in 2001, and M.S. degree in applied mathematics, National Chiao Tung University, Taiwan, in 2004. He is currently working towards the Ph.D. degree in the Graduate Institute of Communication Engineering, National Chiao Tung University, Taiwan. His research interests include optical Internet and Ethernet, protocol design, performance analysis, and traffic control over multimedia high-speed networks, and intelligent techniques involving fuzzy logic, neural networks and neural fuzzy systems. The principal considerations of his Ph.D dissertation are the analysis, simulation and optimal design of traffic control schemes in next-generation optical networks.