行政院國家科學委員會補助專題研究計畫 ■成果報告 □期中進度報告

具 QoS 保證機制都會光纖網路資源控制策略設計之研究

(The Resource Control Strategy for QoS Guaranteed Optical Metro Mesh/Ring Networks)

計畫類別：□個別型計畫　　■ 整合型計畫

計畫編號：98-2221-E-009-058-MY3

執行期間：98 年 8 月 1 日至 101 年 7 月 31 日

計畫主持人：張仲儒 教授

共同主持人：

計畫參與人員：陳振宇、葉長青、蔣佩珊、穆瑞、王華立、劉韋成、楊巧翎、楊明翰、林冠宇、高瑋峻

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份

□赴大陸地區出差或研習心得報告一份

□出席國際學術會議心得報告及發表之論文各一份

□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學電信工程研究所

中　華　民　國　101　年　07　月　05　日

■ 中、英文摘要及關鍵詞(keywords)：（此部分的頁碼編寫請用羅馬字Ⅰ、Ⅱ、Ⅲ…標在每頁下方中央）

■ 目錄：（此部分的頁碼編寫請用羅馬字Ⅰ、Ⅱ、Ⅲ…標在每頁下方中央）

■ 報告內容：(「報告內容」至「附錄」部分請以阿拉伯數字 1.2.3.…順序標在每頁下方中央）

●計畫緣由及目的

●研究方法

●結果與討論：含結論與建議

●參考文獻

※若該計畫已有論文發表者，可以 A4 紙影印，作為成果報告內容或附錄，並請註明發表刊物名稱、卷期及出版日期。

※若有與執行本計畫相關之著作、專利、技術報告、或學生畢業論文等，請在參考文獻內註明之，俾可供進一步查考。

■ 計畫成果自評：請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

■ 可供推廣之研發成果資料表：凡研究性質屬應用研究及技術發展之計畫，請依國科會提供之表格，每項研發成果填寫一份。

■ 附錄（或附件）

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：■已發表 □未發表之文稿 □撰寫中 □無

專利：□已獲得 □申請中 □無

技轉：□已技轉 □洽談中 □無

其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

本計畫分別解決了彈性封包環與橋接式彈性封包環以及光群聚交換環網路三種都會網路的訊務公平性、路由和訊務排程等問題，而研究成果也受到肯定(兩篇國際期刊論文，四篇國際會議論文)。目前各國也開始陸續在鋪設光纖網路以及提高網路服務速度，如台灣中華電信的光世代，該計畫的成果可供中華電信等 ISP 業者應用或借鏡來解決光纖都會網路的服務品質等問題。

計畫中文摘要

高速都會區域網路中的彈性分封環(Resilient Packet Ring)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乏晰公平流速產生器(fuzzy local fairRate generator, FLAG)，藉著乏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)、乏晰壅塞偵測器(fuzzy congestion detector, FCD)、與乏晰公平流速計算器(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乏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。乏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乏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乏晰壅塞偵測器的使用，乏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。

基於頻寬的需求，以及服務更廣大的區域，多個彈性分封環可橋接成一個跨環式彈性分封環網路 BRPR(Bridged RPR)。但是該橋接式網路第一個需要考慮的就是路由控制問題。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擷點雍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。

此外，在光叢集交換(Optical Burst Switching)環中，頻寬利用率、公平性、和穩定性都是很重要的議題。我們提出了一個波長分配與訊務控制(WATC)的方案來達成上述的考量。系統採用了預約的機制來避免時槽式系統中資料的碰撞，而且為了減少光電之間的轉換，每個節點中都沒有過境緩衝器。首先，排程器會依據入口緩衝器中佇列的權重來分配頻寬，然後由波長分配器(WA)決定適合的位置。波長分配器會適時地調整將要產生的資料叢集的大小，藉此有效利用空隙並提升頻寬利用率。另一方面，為了減輕一個下游節點可能遭受到頻寬挨餓的問題，一個乏晰保護性預約產生器(FPRG)會參考過境訊務的情況，動態地預留一些頻寬給入境的聲音與影像訊務。然後，公平流速產生器(FRG)再根據剩餘的頻寬產生一個評估過的公平流速，並由訊務控制器(TC)決定出最終宣傳的公平流速來規範上游的節點。

關鍵詞：彈性分封環、跨環式彈性分封環網路、公平性、服務品質、光叢集交換環、波長分配、訊務控制

Abstract

The resilient packet ring (RPR) is a ring based network for high-speed metropolitan area networks which has properties of fault tolerance and high bandwidth utilization. In RPR, the issues of fairness, stability, and convergence timeare important in congestion control. A local fairRate generator using fuzzy logic and the moving average technique is proposed for the RPR. 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 preproduce a local fairRate by the moving average technique, a fuzzy congestion detector (FCD) to intelligently estimate the congestion degree of the station, and a fuzzy fairRate generator (FFG) to precisely generate the local fairRate. Simulation results show that only the FLAG can stabilize all flows inparking lot scenarios with different finite traffic demands, compared with the conventional aggressive mode (AM) and distributed bandwidth allocation (DBA) fairness algorithms.

Also, we propose an intelligent inter-ring route control, employed in the bridges which connect two RPRs, for the BRPR. The intelligent interring 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. 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).

Finally, the optical burst switching (OBS) ring network is mainly designed for high-speed metropolitan area network, which is expected to support many kinds of services. In the OBS rings, the issues of bandwidth utilization, service differentiation, fairness, and stability are important. A wavelength assignment and traffic control (WATC) scheme is proposed to achieve these considerations. The reservation mechanism is adopted to avoid burst collisions in the time-slot based system, and no transit buffers are deployed at each node for the reason of bypassing the traffic directly. The wavelength assigner (WA) cooperates with the scheduler to generate and then transmit data at the determined position, thereby increasing the bandwidth utilization and supporting service differentiation. In order to alleviate the bandwidth starvation problem a downstream node may suffer from, a fuzzy protective reservation generator (FPRG) preserves some bandwidth for ingress voice and video traffic referring to the network conditions. Afterwards, the fair rate generator (FRG) produces an estimated fair rate, and the traffic controller (TC) determines the advertised fair rate which regulates the upstream node. Since the observation window is applied, the convergence time is reduced.

Keywords： Resilient packet ring, Bridged resilient packet ring, Fairness, Optical burst switching ring, Wavelength assignment, Traffic control

Agenda

I. **Introduction...1**

II. **FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Ring 6**

III. **Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings ……….....11**

IV. **Reservation Slotted OBS Rings with Wavelength Assignment and Traffic Control 23**

V. **Reference... 35**

VI. **Comment 38**

I. Introduction and Motivation

In development of the Internet, the technology of the wavelength division multiplexing (WDM) has impacted the designment and realization of the next gerneration network. From the piont-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]-[6]. 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.

The ring network with the natural advantage, such as simple archtechure, 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, as shown in Fig. 1.1, and has been recently approved as the IEEE 802.17 Standard [7]. 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 [7, 8]. 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 [9, 10]. 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 [9, 10, 11]. The RPR is a superior candidate for MANs.

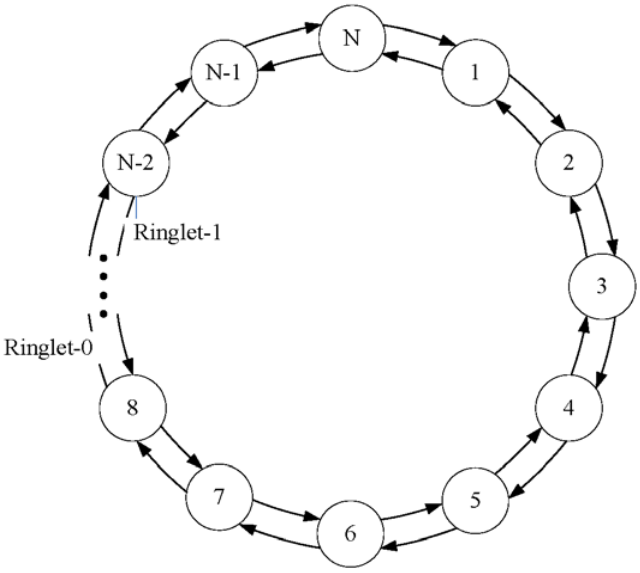


Figure 1.1: RPR structure

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 lowpriority 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 [9, 12]. 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 [9, 12]. 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 [13]. 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) [9, 13] and the aggressive mode (AM) [9, 10] 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 [9, 12]. 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 [7, 14]. 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 [7, 14, 15, 16].

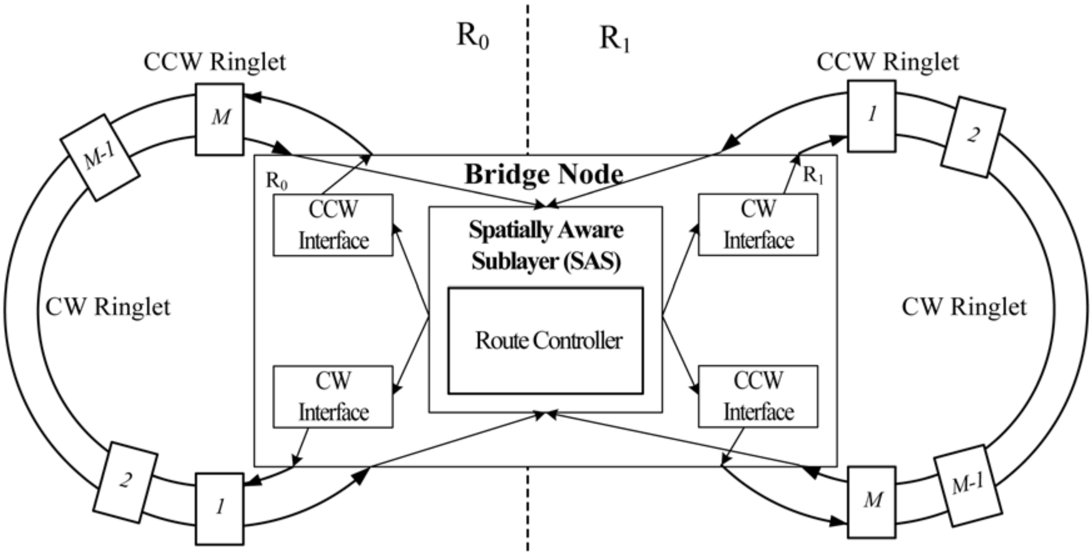


Figure 1.2: BRPR structure

Moreover, serving as a backbone that interconnects a number of access networks, OBS ring topologies have been a good choice for solving the current metro gap problem between core network and access network owing to its simplicity and scalability.

I.1. FLAG: A Fuzzy Local FairRate Generator for 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 [13]. The RPR defines two fairness algorithms, conservative mode (CM) [9, 13] and the aggressive mode (AM) [9, 10] 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 [13, 17]. 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 [13, 17-22]. The distributed virtualtime scheduling (DVSR) [13] 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(NlogN), 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) [17, 18]. 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 shortterm 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 [20]. 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.

I.2. Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings

Settawong and Tanterdtid proposed an enhancement by using a topology discovery and spanning tree algorithm [15]. 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 [23, 24, 25] 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 [25]. Also, Heiden et. al. analyzed the capacity of bidirectional optical packet ring networks, such as RPR, which employs the SPRC for multicast hotspot traffic [26]. 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 [7]. 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.

I.3. Reservation Slotted OBS Rings with Wavelength Assignment and Traffic Control

Each network node in an OBS ring network employs transmitters and receivers to send and receive data traffic. There are several architectures with variants combinations of transmitters and receivers in OBS nodes [27, 28, 29]. A more scalable and flexible system with tunable transmitter–tunable receiver (TT–TR) architecture was also proposed [27]. Its advantages come at the expense of a higher resource contention possibility and a higher packet loss probability. In this paper, we adopt the TT-TR architecture for the reason of scalability.

At each node, packets with the same destination are assembled into a data burst (DB) by assembly algorithm, such as length based algorithm, time based algorithm and hybrid algorithm [30]. The DB must be transmitted according to a specific medium access control (MAC) protocol to avoid burst collisions. Several proposed MAC protocols can be classified into two major categories: token based scheme [31, 32] and time-slot based scheme [33]. Unfortunately, the token based schemes cause low channel utilization. The time-slot based scheme solves this problem, but it does not assure class of service (CoS).

Wavelength assignment is an important work in OBS networks. An appropriate wavelength assignment method not only makes better use of link bandwidth but also supports CoS. Some methods of wavelength assignment have been proposed for OBS network [34, 35, 36]. For ring networks, however, there is still a problem of bandwidth sharing among the nodes. When traffic load increases, the downstream nodes may suffer from bandwidth starvation. A traffic control method can prevent this situation and provide fair access to the link bandwidth. There are some methods of traffic control proposed for resilient packet ring (RPR) which is also a ring based network [37, 38].

II. FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Ring

➢ Architecture of Intermediate OBS Node

Assume that a resilient packet ring (RPR) with N stations, shown in Fig. 1.1, 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.

Fig. 2.1 shows the station structure for ringlet-0 transmisson, 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, 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 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.

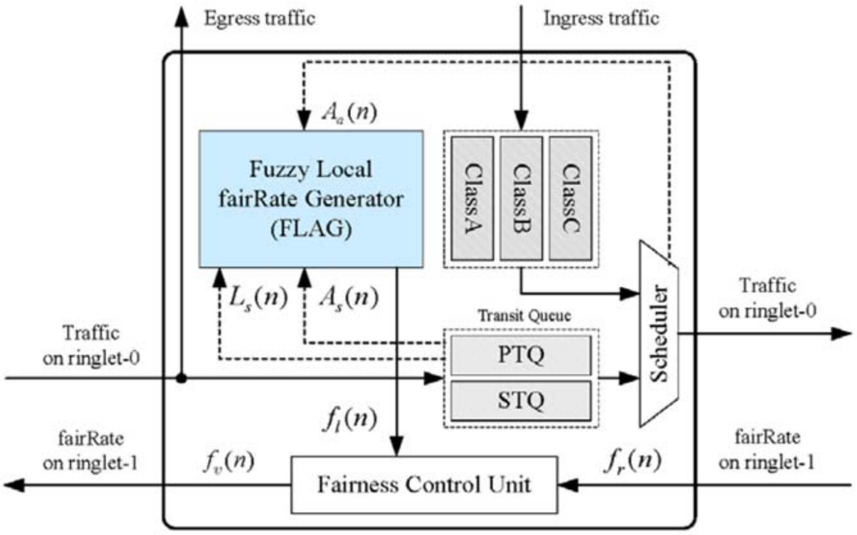


Figure 2.1: RPR station structure

The advertised fairRate generated by the fairness control unit are described as follows. The *fv* would be set to be *fl* if *fr* is smaller than *fl* 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(*fl, fr*). Here we also describe the advertised fairRate generated by AM below. When the station is congestion free, the *fv* is set to be the FullRate if the fr is larger than the bandwidth rate of the transit FE traffic flows which will pass through the originally congested station; to be *fr*, 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 *fv* is set to be *fl* if the *fr* is FullRate; to be min(*fl, fr*), 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. Also, the originally congested station is known to the observation station since the message of the advertised fairRate contains a field to record it; the fl is the added FE traffic flow rate to the network.

- **Fuzzy Local FairRate Generator (FLAG)**

The proposed fuzzy local fairRate generator (FLAG), shown in Fig. 2.2, is composed of an adaptive fairRate calculator (AFC), a fuzzy congestion detection (FCD), and a fuzzy fairRate generator (FFG). During the nth agingInterval which is from time ($n-1$)T to time nT, the FLAG determines $fl(n)$ 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..

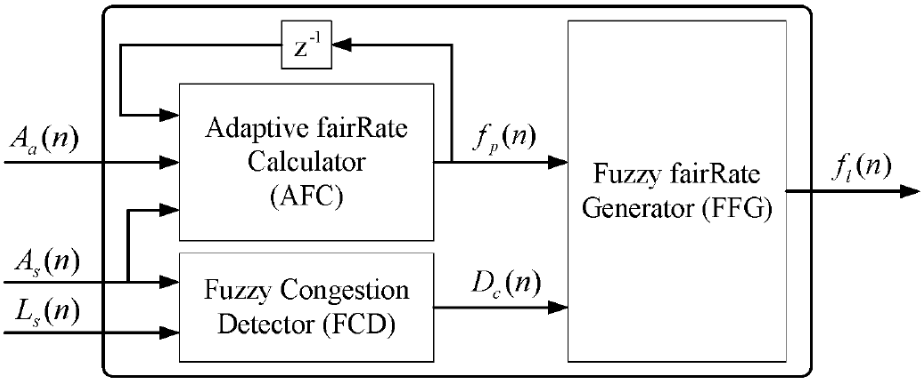

z^{-1}
$A_a(n)$
Adaptive fairRate Calculator (AFC)
$f_p(n)$
Fuzzy fairRate Generator (FFG)
$f_l(n)$
$A_s(n)$
$L_s(n)$
Fuzzy Congestion Detector (FCD)
$D_c(n)$

Figure 2.2: Functional blocks of FLAG

- Adaptive fairRate Calculator (AFC)

The adaptive fairRate calculator (AFC) adopts the moving average technique [14] 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 [5]. During the n-th agingInterval, the AFC first takes the moving average of arriving transit FE traffic flows to STQ, $A_s(n)$. Denote the average by $\tilde{A}_s(n)$ and give it by

$$\tilde{A}_s(n) = \sum_{i=n-k+1}^{n} A_s(n) / k \tag{2.1}$$

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

$$M(n) = \frac{\tilde{A}_s(n) + A_a(n)}{f_p(n-1)} \quad . \tag{2.2}$$

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

$$f_p(n) = \mathrm{Min}\left\{C, f_p(n-1)\left[C-(A_s(n)+A_a(n))\right]\right\} \tag{2.3}$$

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

- Fuzzy Congestion Detector (FCD)

The FCD refers not only the occupancy of STQ, $L_{\mathrm{S}}(n)$, as defined in the IEEE802.17, but also the arriving FE traffic flows to STQ, $A_s(n)$, to determine the congestion degree, $A_s(n)$. 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 $D_c(n)$. We define the term set for $L_{\mathrm{S}}(n)$ as $\mathrm{T}(L_{\mathrm{S}}(n))$ = {Short (S), Long (L)}; for $A_s(n)$ as $\mathrm{T}(A_s(n))$ = {low (L), Medium (M), High (H)}; for $A_s(n)$ as $\mathrm{T}(A_s(n))$ = {Very Low (VL), Low (L), Medium (M), High (H), Very High (VH)}.

Here, the triangular function $f(x : x_0, a_0, a_1)$ and the trapezoidal function $g(x : x_0, x_1, a_0, a_1)$ are used to define the membership functions for the terms in the term set. These two functions are

$$f(x;x_0,a_0,a_1)=\begin{cases}\dfrac{x-x_0}{a_0}+1, & \text{for } x_0-a_0<x\le x_0,\\ \dfrac{x_0-x}{a_1}+1, & \text{for } x_0<x\le x_0+a_1,\\ 0, & \text{otherwise.}\end{cases} \tag{2.4}$$

and

$$g(x;x_0,x_1,a_0,a_1)=\begin{cases}\dfrac{x-x_0}{a_0}+1, & \text{for } x_0-a_0<x\le x_0,\\ 1, & \text{for } x_0<x\le x_1,\\ \dfrac{x_1-x}{a_1}+1, & \text{for } x_1<x\le x_1+a_1,\\ 0, & \text{otherwise}\end{cases} \tag{2.5}$$

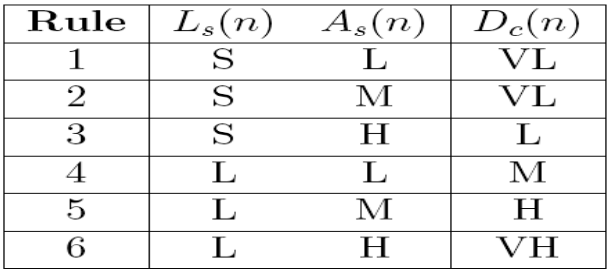
where x_0 in $f(\cdot)$ is the center of the triangular function; x_0 (x_1) in $g(\cdot)$ is the left (right) edge of the trapezoidal function; a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

The corresponding membership functions of S and L in T ($L_s(n)$) are denoted by $\mu_s(L_s(n)) = g(L_s(n); 0, 0.125Q, 0, 0.25Q)$ and $\mu_L(L_s(n)) = g(L_s(n); 0.35Q, Q, 0.25Q, 0)$, where Q is the size of STQ. As defined in IEEE 802.17 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 $\mu_L(A_s(n)) = g(A_s(n); 0, 0.125\mathrm{C}, 0, 0.375\mathrm{C})$, $\mu_M(A_s(n)) = f(A_s(n); 0.5\mathrm{C}, 0.25\mathrm{C}, 0.25\mathrm{C})$, and $\mu_H(A_s(n)) = g(A_s(n); 0.875\mathrm{C}, \mathrm{C}, 0.375\mathrm{C}, \mathrm{C})$, respectively. 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 $\mu_{VL}(D_c(n)) = f(D_c(n); 0, 0, 0)$, $\mu_L(D_c(n)) = f(D_c(n); 0.25, 0, 0)$, $\mu_M(D_c(n)) = f(D_c(n); 0.5, 0, 0)$, $\mu_H(D_c(n)) = f(D_c(n); 0.75, 0, 0)$, and $\mu_{VH}(D_c(n)) = f(D_c(n); 1, 0, 0)$, respectively.

There are 6 fuzzy rules for FCD. As shown in Table 2.1, the order of significance of the input linguistic variables is $L_s(n)$ then $A_s(n)$. The station with high occupancy 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).

Tabel 2.1 The fuzzy rules for FCD



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

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 1and 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

$$m_1(n) = \min\{\mu_S(L_s(n)),\ \mu_L(A_s(n))\} \tag{2.6}$$

$$m_2(n) = \min\{\mu_S(L_s(n)),\ \mu_M(A_s(n))\} \tag{2.7}$$

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

$$\omega_{VL}(n) = \max\{m_1(n),\ m_2(n)\} \tag{8}$$

The fuzzy inference results of the output indication L, M , H , and VH , denoted by $\omega_L(n)$, $\omega_M(n)$, $\omega_H(n)$, and $\omega_{VH}(n)$, 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 obtained by

$$z_0 = \frac{0.0\times\omega_{VL}(n) + 0.25\times\omega_L(n) + 0.5\times\omega_M(n) + 0.75\times\omega_H(n) + 1.0\times\omega_{VH}(n)}{\omega_{VL}(n) + \omega_L(n) + \omega_M(n) + \omega_H(n) + \omega_{VH}(n)}$$

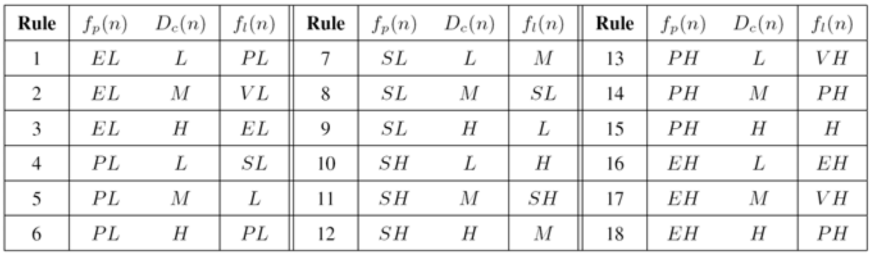
(2.9)

■ Fuzzy fairRate Generator (FFG)

The FFG refers 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($f_p(n)$) = {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($D_c(n)$) ={Low (L), Medium (M), High (H)}; and the term set with eleven terms for $f_l(n)$ as T($f_l(n)$) ={Extremely Low (EL), Very Low (V L), Pretty Low (PL), Low (L), Slightly Low (SL), Medium (M), Slightly High (SH), High (H), Pretty High (P H), Very High (V H), 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; P L; SL; SH; P H; and EH in T($f_l(n)$) are defined as $\mu_{EL}(f_p(n)) = f(f_p(n); 0, 0, 0.3\text{C})$, $\mu_{PL}(f_p(n)) = f(f_p(n); 0.2\text{C}, 0.2\text{C}, 0.2\text{C})$, $\mu_{SL}(f_p(n)) = f(f_p(n); 0.4\text{C}, 0.2\text{C}, 0.2\text{C})$, $\mu_{SH}(f_p(n)) = f(f_p(n); 0.6\text{C}, 0.2\text{C}, 0.2\text{C})$, $\mu_{PH}(f_p(n)) = f(f_p(n); 0.8\text{C}, 0.2\text{C}, 0.2\text{C})$, and $\mu_{EH}(f_p(n)) = f(f_p(n); \text{C}, 0.3\text{C}, 0)$, respectively. The membership functions for terms L; M; and H in T($D_c(n)$) are defined as $\mu_L(D_c(n)) = g(D_c(n); 0, 0.125, 0, 0.375)$, $\mu_M(D_c(n)) = f(D_c(n); 0.5, 0.25, 0.25)$, and $\mu_H(D_c(n)) = g(D_c(n); 0.875, 1, 0.375, 0)$, respectively. The membership functions for terms in T($f_l(n)$) are defined as fuzzy singletons, denoted by $\mu_T(f_l(n)) = f(f_l(n); \text{x}_T, 0, 0)$, where T = EL;V L; PL; L; SL; M; SH; H; PH; V H; or EH , and (x_{EL}, x_{VL}, x_{PL}, x_L, x_{SL}, x_M, x_{SH}, x_H x_{PH} x_{VH} x_{EH}) = (0, 0.1C, 0.2C, 0.3C, 0.4C, 0.5C, 0.6C, 0.7C, 0.8C, 0.9C, 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; P L; SL; SH; P H; and EH .

There are 18 fuzzy rules for FFG. As shown in Table 2.2, 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 fp(n) is ‘’EL’’ or “PL”, $f_l(n)$ is designed to raise two levels more than $f_p(n)$ (EL→PL or PL → 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 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 → 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 to generate a crisp-valued local fairRate.

Tabel 2.2 Rule Base for FFG



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

- **Simulation Result**

In the simulations, settings for the environment include 10 Gbps link capacity, 200μs 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 stqLowthreshold is 0.5 Mbytes. Simulations for the proposed FLAG, DBA with moving average technique (DMA), DBA, and AM 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. 2.3(a) shows a small parking lot scenario where there are 5 (0 ~ 4) greedy stations, and Figs. 2.3(b), 2.3(c), 2.3(d) and 2.3(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 atstation 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. ThusFLAG 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. 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 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.

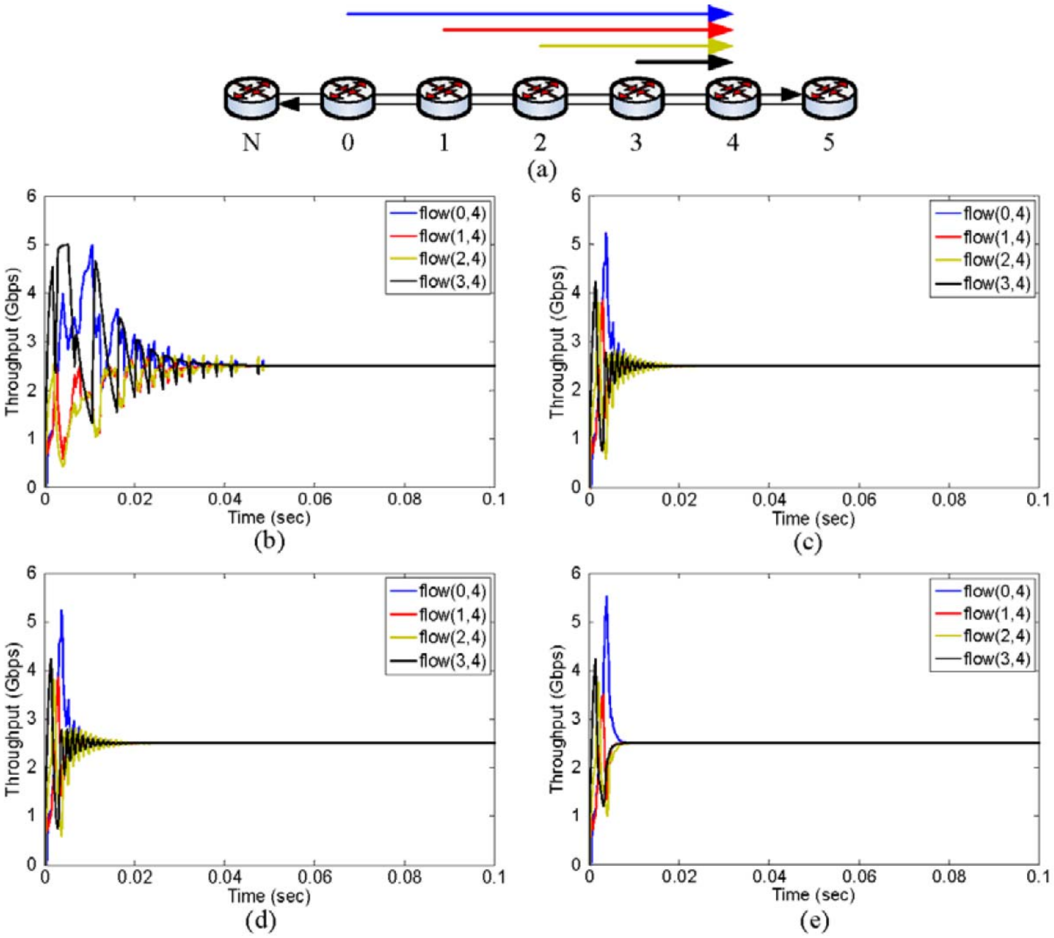


Figure 2.3: (a) Small parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DBA with moving average (DMA), and (e) FLAG.

Fig. 2.4(a) shows a large parking lot scenario where there are containing 8(0 ~ 7) greedy stations, and Figs. 2.4(b), 2.4(c), 2.4(d) and 2.4(e) presentthe 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. 2.3 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. 2.3(b) is given below. Since, here in Fig. 2.4(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.

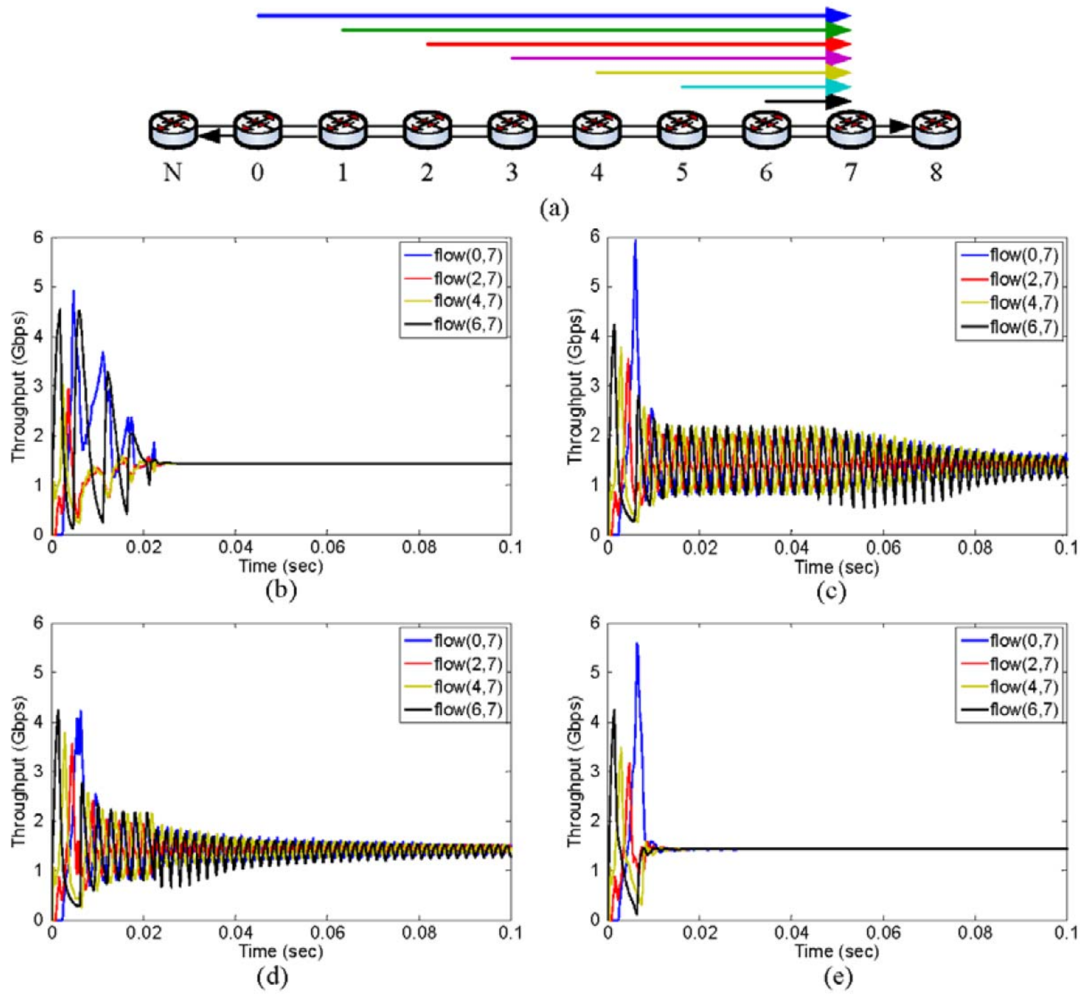


Figure 2.4: (a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.

Fig. 2.5(a) shows a large parking lot scenario where there are containing 8 (0 ~ 7), such as in Fig. 2.4 (a) but with various finite traffic demands, greedy stations, and Figs. 2.5 (b), 2.5 (c), 2.5 (d), and 2.5 (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, 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 freecongestion 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. 2.4(e), the damping amplitude is smaller than that in Fig. 2.4(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. 2.4(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. 2.4. 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 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.

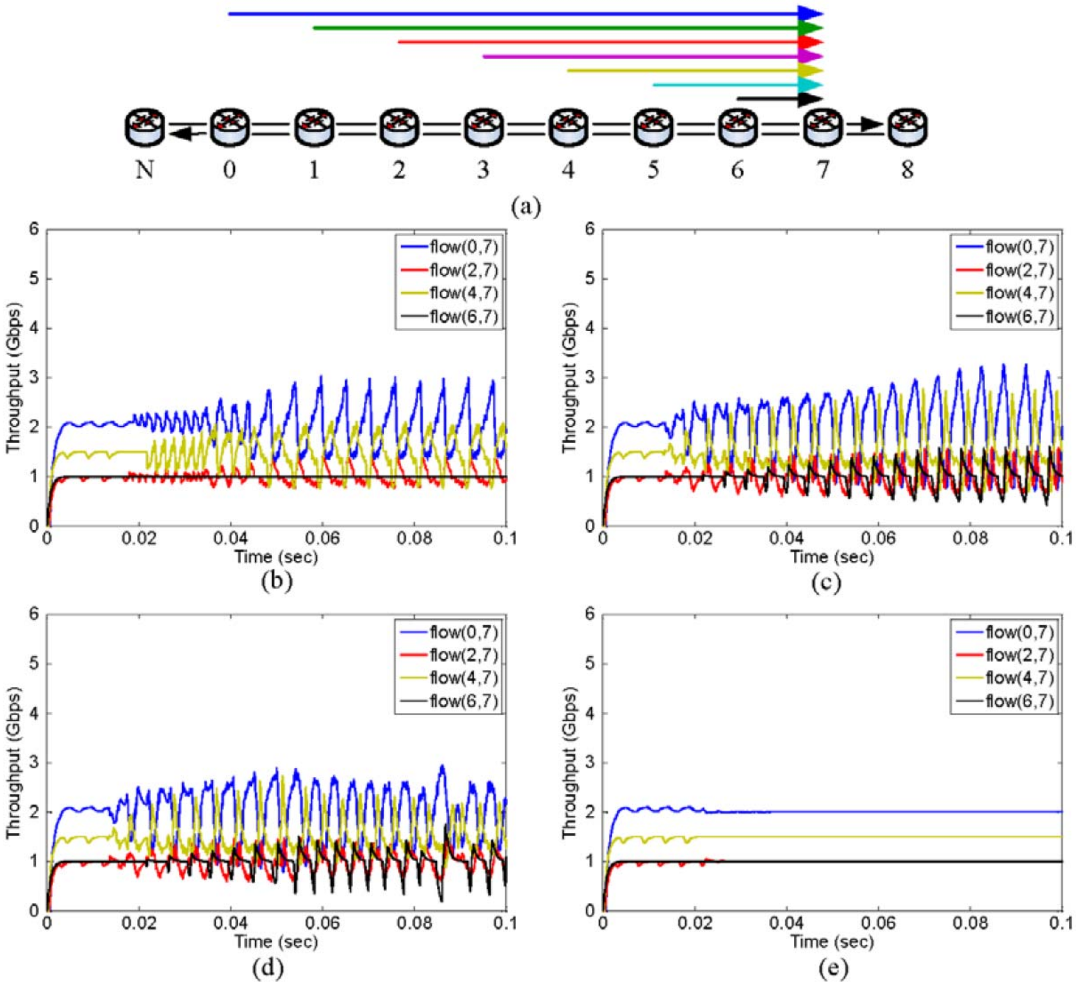


Figure 2.5: (a) Large parking lot scenario with greedy traffic, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG in a large parking lot scenario with various finite traffic flows.

Fig. 2.6(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. 2.6 (b), 2.6 (c), 2.6 (d) and 2.6 (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 Gbps. 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 withvarious finite traffic flows, given in Fig. 2.4, 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 Gbps traffic flow maximum, while station 2 has 5.2 Gbps traffic flow maximum. This phenomenon is conversed in Fig. 2.5, where station 1 has 4.2 Gbps traffic flow maximum, while station 2 has 5.2 Gbps maximum. Therefore, the 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. 2.6 would oscillate worse than in Fig. 2.5 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.

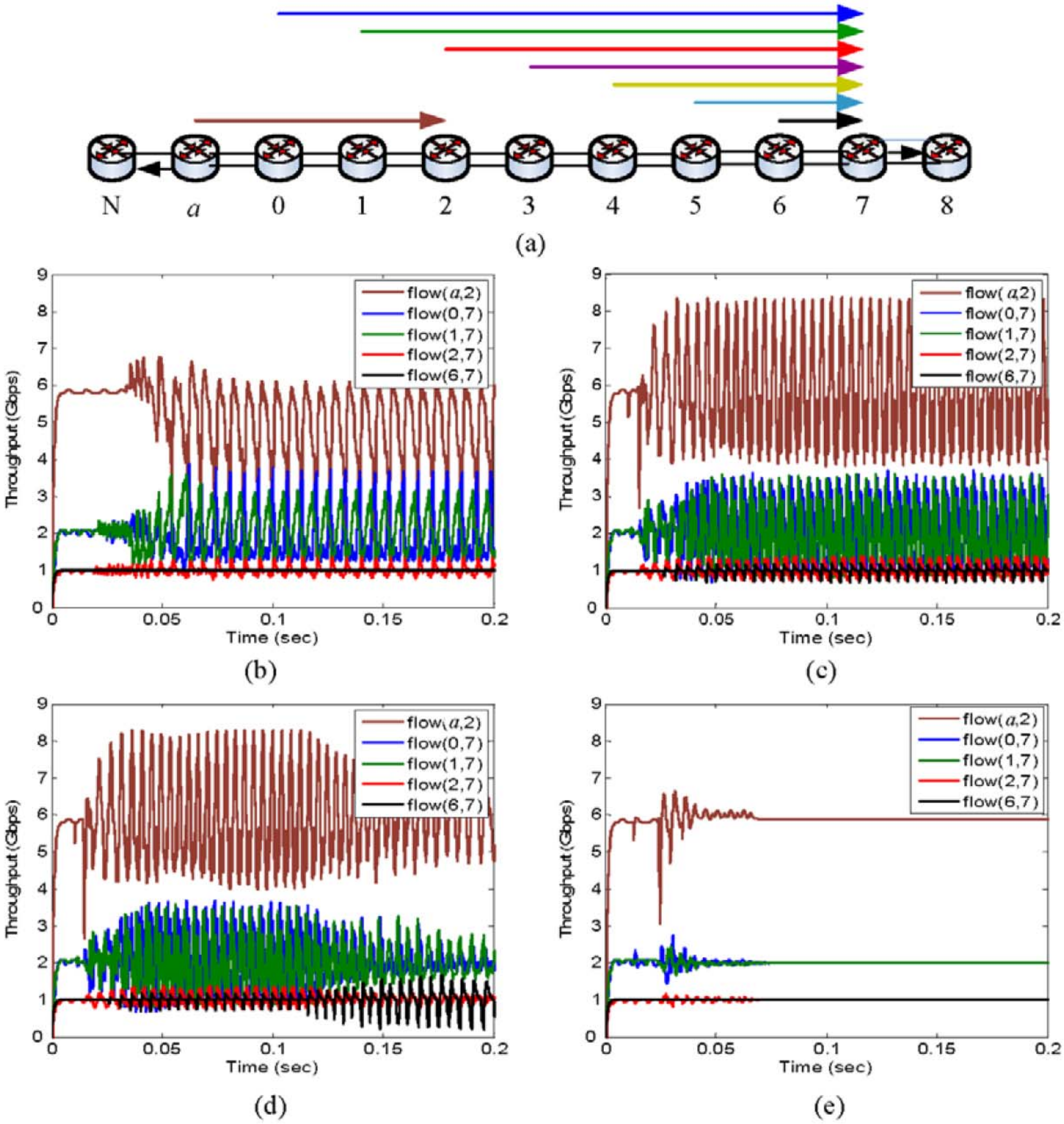


Figure 2.6: (a) Available bandwidth reclaiming scenario with finite traffic demand, and the throughput of (b) AM, (c) DBA, (d) DMA, and (e) FLAG.

III. Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings

➢ System Model

A. Architecture of Bridge Node

Fig. 3.1 shows a bridge node connecting R0 and R1 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 bridge node based on the determined route. The bridge node has one interface associated each ringlet, and as shown in Fig. 3.2, 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.

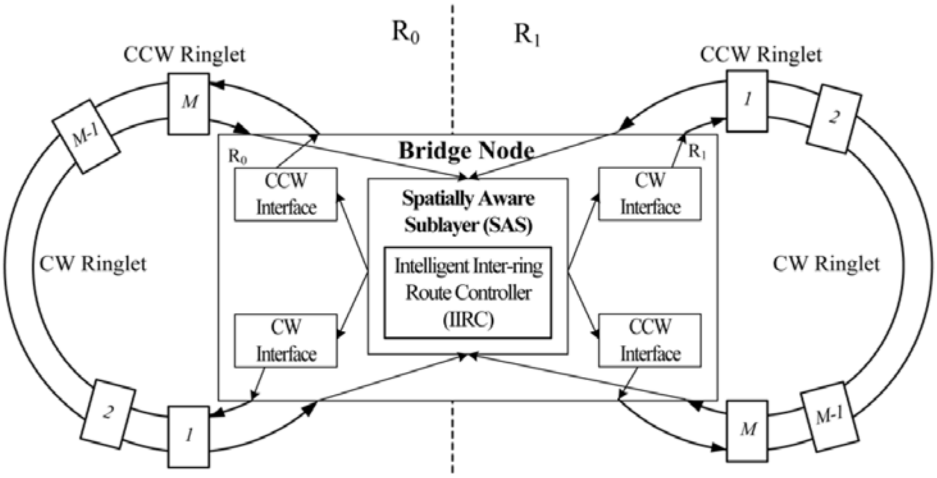


Figure 3.1: Architecture of the bridge node

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.

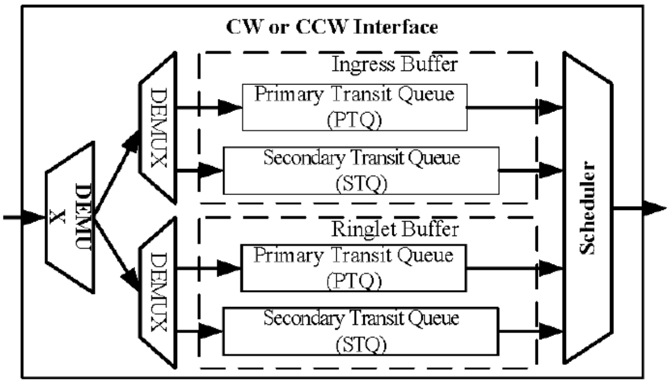

CW or CCW Interface
DEMUX
DEMUX
DEMUX
Ingress Buffer
Primary Transit Queue (PTQ)
Secondary Transit Queue (STQ)
Ringlet Buffer
Primary Transit Queue (PTQ)
Secondary Transit Queue (STQ)
Scheduler

Figure 3.2: Architecture of the interface

B. 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. For simplicity, we adopt the AM fairness algorithm in each node for simulations. The AM fairness algorithm is described as follows. As specified in, 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 μs. 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.

C. 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. 3.1 and 3.2. Thus as shown in Fig. 3.3, 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 $\hat{R}_f$, as input linguistic 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 H, as input linguistic variables. Notice that the ringlet service rate at the bridge node is related with the received fairRate and more hops consume more system bandwidth. The FRC calculates the preference value of route, denoted by Pv, for CW and CCW interfaces and selects the ringlet with larger Pv as the proper ringlet route for the incoming inter-ring new call request.

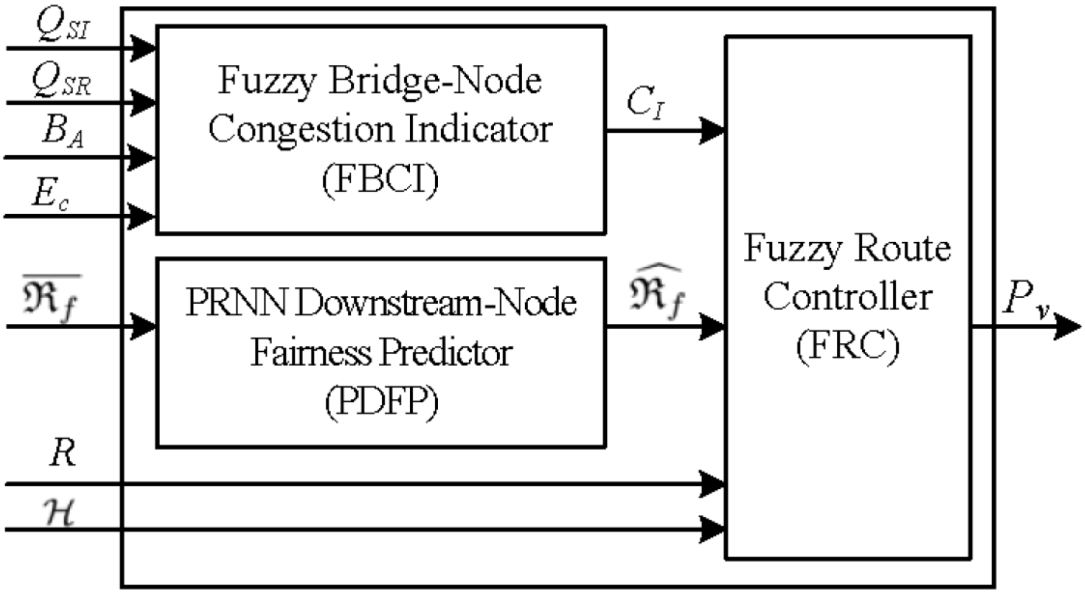

Q_{SI}
Q_{SR}
B_A
E_c
$\overline{\Re_f}$
Fuzzy Bridge-Node Congestion Indicator (FBCI)
C_I
PRNN Downstream-Node Fairness Predictor (PDFP)
$\widehat{\Re_f}$
Fuzzy Route Controller (FRC)
P_v
R
$\mathcal{H}$

Figure 3.3: Intelligent inter-ring route controller (IIRC)

- **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, denoted by *Ec*. 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. 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 BA 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 Ec 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})) = {Short (S), Medium (M), Long (L)}; T(B_A) = {Few (Fw), Many (Ma)}; T(*Ec*) = {Small (S), Large (L)}, and T(C_I) = {Very Low (V L), Low (L), Medium (M), High (H), Very High (VH)}.

Here, the triangular function $f(x : x_0, a_0, a_1)$ and the trapezoidal function $g(x : x_0, x_1, a_0, a_1)$ are used to define the membership functions for the terms in the term set. These two functions are

$$f(x;x_0,a_0,a_1)=\begin{cases}\dfrac{x-x_0}{a_0}+1, & \text{for } x_0-a_0<x\le x_0,\\ \dfrac{x_0-x}{a_1}+1, & \text{for } x_0<x\le x_0+a_1,\\ 0, & \text{otherwise.}\end{cases} \tag{3.1}$$

and

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

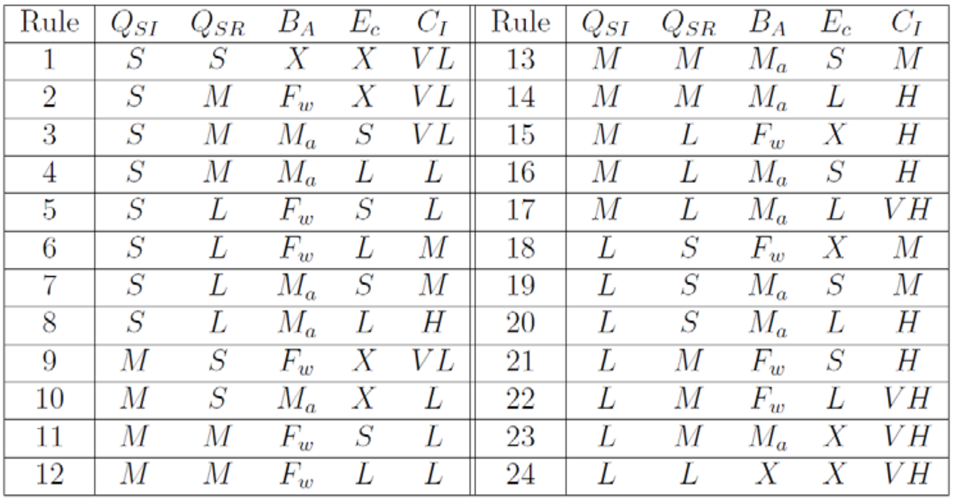
where x_0 in $f(\cdot)$ is the center of the triangular function; x_0 (x_1) in $g(\cdot)$ is the left (right) edge of the trapezoidal function; a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

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

$\mu_S(Q_{SI}) = g(\frac{Q_{SI}}{L_S}; 0, 0.25\,l_{th}, 0, 0.25\,l_{th})$, $\mu_M(Q_{SI}) = f(\frac{Q_{SI}}{L_S}; 0.5l_{th}, 0.5l_{th}, 0.5l_{th})$,

$\mu_L(Q_{SI}) = g(\frac{Q_{SI}}{L_S}; l_{th}, 1, 0.5l_{th}, 0)$,where *Ls* is the STQ queue size, and lth is the threshold in percentage. Note that if the STQ length is larger than $l_{\text{th}} \cdot Ls$, 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 Fw and Ma in T(B_A) are defined as $\mu_{Fw}(B_A) = g(B_A; 0, 0.025\text{C}, 0, 0.025\text{C})$ and $\mu_{Ma}(B_A) = g(B_A; 0.1\text{C}, \text{C}, 0.075\text{C}, 0)$, where C is the link capacity. Membership functions of terms in T(Ec), are defined as $\mu_S(Ec) = g(Ec; 0, \text{R}_{\text{voice}}, 0, \text{R}_{\text{video}})$ and $\mu_L(Ec) = g(Ec; \text{R}_{\text{h}}, \text{C}, \text{R}_{\text{video}}, 0)$, where R_{voice} and R_{video} are the minimum demand of the mean rates of the voice and video traffics, respectively, and Rh 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 $\mu_{VL}(C_I) = f(C_I; 0.1, 0, 0)$, $\mu_L(C_I) = f(C_I; 0.3, 0, 0)$, $\mu_M(C_I) = f(C_I; 0.5, 0, 0)$, $\mu_H(C_I) = f(C_I; 0.75, 0, 0)$, $\mu_{VH}(C_I) = f(C_I; 1, 0, 0)$.

Table 3.1: 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

As shown in Table 3.1, 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 *Ec* 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). Finally, FBCI adopts the max-min method for fuzzy inference. The defuzzification method adopted is the center of area defuzzification method.

- **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 considered here, thereceived 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 nth period, denoted by $\overline{R_f}(n)$, as the input variable, where m is the size of the observation window, $m \geq 1$. The $\overline{R_f}(n)$ could be appropriate to detect the congestion situation of the downstream nodes during a period and it is expressed by

$$\overline{R_f}(n) = \frac{R_f(n) + R_f(n-1) + \cdots + R_f(n-m+1)}{m},$$

where $R_f(n)$ 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) 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 $\hat{R}_f(n+1)$ or $\hat{R}_f$ for convenience, is given by

$$\hat{R}_f(n+1) = H(\overline{R_f}(n), \ldots, \overline{R_f}(n-p+1); \hat{R}_f(n), \ldots, \hat{R}_f(n-q+1))$$

where $\hat{R}_f(i)$ is the previously predicted mean fairRate at ith period, $n-q+1 \leq i \leq n$, and H(·) 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(·).

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, $\hat{R}_f(n+1)$, from the output of the first neuron of the first module, and receives the new fairRate $\overline{R_f}(n+1)$; 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. 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).

- **Fuzzy Route Controller (FRC)**

The fuzzy route controller (FRC) is to determine the route preference values, Pvs, 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, $\hat{R}_f$, the current service rate of the ringlet, R, and the number of hops to destination, H. The higher value Pv 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) = {Low (Lo), Medium (Me), High (Hi)}, T($\hat{R}_f$) = {Small (Sm), Medium (Me), Large (La)}, T(R) = {Low (Lo), High (Hi)}, T(H) = {Few (Fw), Many (Ma)}, and T(Pv) = {Unsuitable (U), Weakly Unsuitable (WU), Weakly Suitable (WS), Suitable (S)}. Membership functions for terms of Lo,Me, and Hi in T(C_I) are defined as $\mu_{Lo}(C_I) = g(C_I; 0, 0.25, 0, 0.25)$, $\mu_{Me}(C_I) = f(C_I; 0.5, 0.25, 0.25)$, and $\mu_{Hi}(C_I) = g(C_I; 0.75, 1, 0.25, 0)$. Membership functions for terms of Sm, Me, and La in T($\hat{R}_f$) are expressed as $\mu_{Sm}(\hat{R}_f) = g(\hat{R}_f; 0, 0.2v, 0, 0.25v)$, $\mu_{Me}(\hat{R}_f) = f(\hat{R}_f; 0.5v, 0.25v, 0.25v)$, and $\mu_{La}(\hat{R}_f) = g(\hat{R}_f; 0.6v, v, 0.1v, 0)$, where v denotes the unreserved bandwidth for the low priority traffic at the bridge node and v = C − B_A.

Similarly, membership functions for T(R) are defined as $\mu_{Lo}(R) = g(R; 0, 0.25\mathrm{C}, 0, 0.25\mathrm{C})$ and $\mu_{Hi}(R) = g(R; 0.6\mathrm{C}, \mathrm{C}, 0.2\mathrm{C}, 0)$, where C is the total capacity of the fiber link. Membership functions for terms of Fw and Ma in T(H) are defined as $\mu_{Fw}(H) = g(H; 0, N/3, 0, N/3)$ and $\mu_{Ma}(H) = g(H;\ 2N/3, N, N/3, 0)$, where N is the total number of nodes in a RPR network. Finally, membership functions for terms in T(Pv) are defined as

$\mu_U(Pv) = f(Pv; 0.1, 0, 0)$, $\mu_U(Pv) = f(Pv; 0.4, 0, 0)$, $\mu_U(Pv) = f(Pv; 0.1, 0, 0)$, and $\mu_U(Pv) = f(Pv; 1, 0, 0)$.

As shown in Table 3.2, there are 21 fuzzy rules. The notation ”X” in Table 3.2 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, $\hat{R_f}$, R, and H. The low congestion degree of ringlet interface (C_I = Lo) and the large or medium predicted mean received fairRate ($\hat{R_f}$ = La or Me) would make the inter-ring new call have more chance to enter the interface. However, the low congestion degree of ringlet interface (C_I = Lo), but the small predicted mean received fairRate ($\hat{R_f}$ = Sm) which means that the downstream nodes may incur congestion, and the high ringlet service rate (R = Hi) would make the variable of the number of hops to destination H significant. If H is Few, the new call will be weakly suitable for the ringlet, while if 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 = Hi) and the small predicted mean received fairRate ($\hat{R_f}$ = Sm) would make the inter-ring new call have less chance to enter the interface. However, the high congestion degree of ringlet interface (C_I= Hi), but the large predicted mean received fairRate ($\hat{R_f}$= La) which means that the downstream nodes are free of congestion, and the high ringlet service rate (R = Hi) would similarly make the variable of the number of hops to destination H significant. The fuzzy inference algorithm also adopts the max-min inference method, and the defuzzification method, the center of area defuzzification method.

- **Simulation Result**

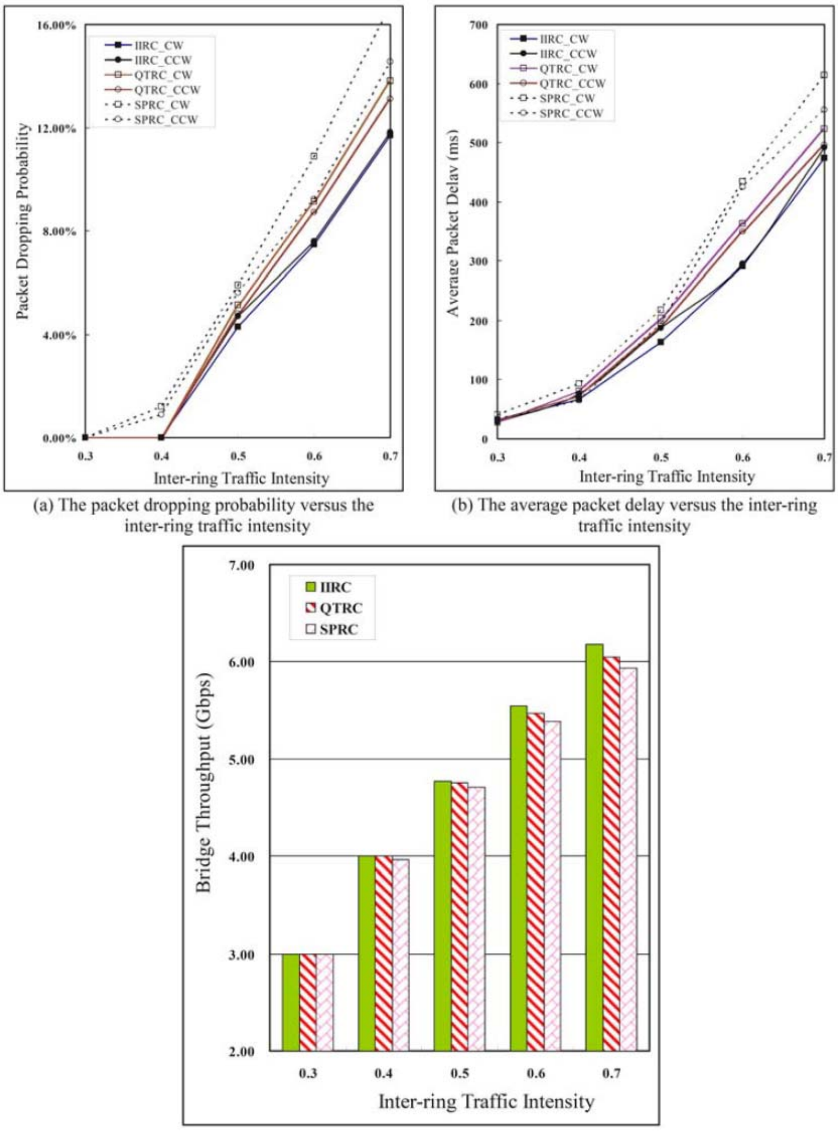
Simulations are here conducted to compare the performances of proposed IIRC, and SPRC. Also, an intuitive queue-length threshold route controller (QTRC) is included, which determines a proper ringlet depending on the shorter STQ length of ingress buffer. Traffic flows from R1 to R0 at the bridge node are considered. Referring to Fig. 3.1, assume that there are M = 16 non-bridge nodes on R0, the link capacity is C = 10.0 Gbps, and sizes of the two PTQs and the two STQs are 40 Mbyte with threshold lth = 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

Rp, the mean rate Rm, and the mean burst period Tp with the four previous rates can be obtained.

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 Rp = 21×10^{-4}, Rm = 7×10^{-4}, and Tp = 1.3s. 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 Rp = 5×10^{-2}, Rm = 1×10^{-2}, and Tp = 0.1s. 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 θB and θP , respectively. For B-frame-bit-rate of the B-frame of video packet source, it was assumed that Rp = 2×10^{-2}, Rm = 2×10^{-3}, and Tp = 0.01s, which is given θB = 0.1; for P-frame-bit-rate of the P-frame of video packet source, it was assumed that Rp = 1×10^{-2}, Rm = 2×10^{-4}, and Tp = 0.01s, which given θP = 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 θ1 and θ2, respectively. For highbit- rate of data source, it was assumed that Rp = 7×10^{-2}, Rm = 7×10^{-3}, and Tp = 0.03s, which is given θ1 = 0.1; for low-bit-rate of data source, it was assumed that Rp = 3.5×10^{-2}, Rm = 7×10^{-4}, and Tp = 0.03s, which is given θ2 = 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, Rvoice, Rvideo, and Rh, are set to 64kbps, 640kbps, and 6.4Mbps, respectively.

Fig. 3.4(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 R1 to R0 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 R0, 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 probability of the destination of the incoming new calls is uniformly distributed over nodes on R0. 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 pathes 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. 3.5(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 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 R0 exceed a threshold and thus SPRC gets a worse throughput.

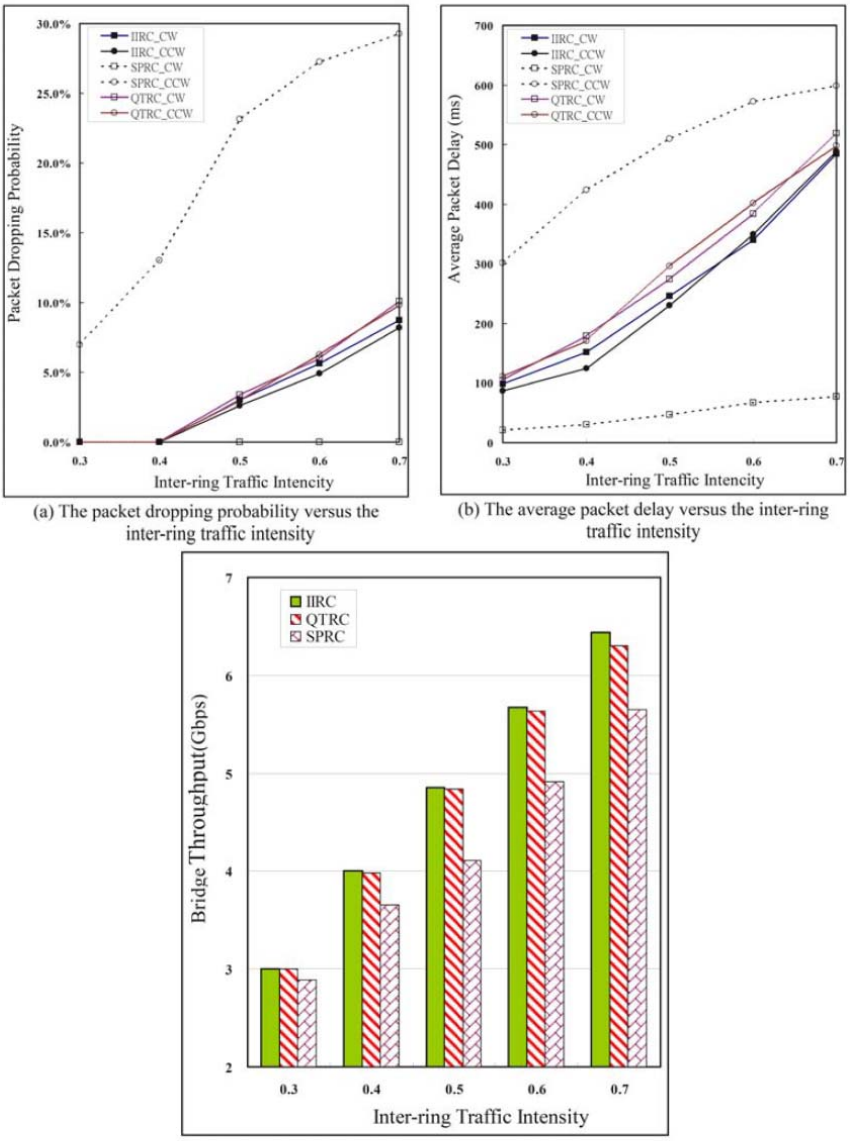


(a) The packet dropping probability versus the inter-ring traffic intensity

(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 3.4: The performance comparison for IIRC, SPRC, and QTRC in the balanced scenario



(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 3.5: The performance comparison for IIRC, SPRC, and QTRC in the unbalanced scenario

Fig. 3.6 shows the bridge throughputs under IIRC, IIRC without considering EC and/or BA 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 EC, IIRC without considering BA, and IIRC without considering BA and EC, respectively. These can justify that the input linguistic variables BA and EC are essential, and the BA input linguistic variable is more important than EC.

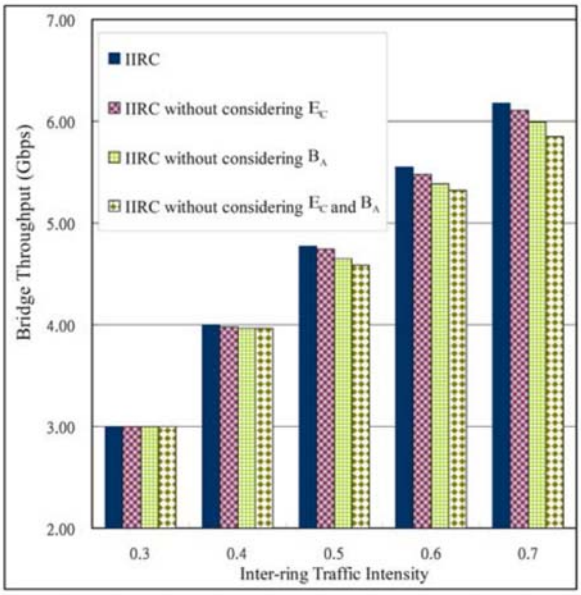


Figure 3.6: The comparison of bridge throughputs under different schemes versus the inter-ring traffic intensity

IV. Reservation Slotted OBS Rings with Wavelength Assignment and Traffic Control

➢ System Model

Assume that the slotted OBS ring is with N nodes and is constructed by two unidirectional, counter-rotating ringlets, named ringlet-0 and ringlet-1. Each node has two pairs of input and output ports to communicate with neighbor nodes. Node X (Y) is said to be an upstream (downstream) node of node Y (X) on ringlet-0 or ringlet-1 if the node Y (X) traffic becomes the received traffic of node X (Y) on the referenced ringlet. There are four classes of service considered: voice, video, the hypertext transfer protocol (HTTP), and the file transfer protocol (FTP).

■ *Slotted OBS Rings*

The slotted OBS ring contains W data wavelengths, denoted by λ_1, …, λ_W, and one control wavelength, denoted by λ_c, on each ringlet. As shown in Fig 4.1, there is a frame structure on each wavelength; each frame of data wavelength is composed of S slots; and there is a wavelength reservation (WR) transited each frame in control wavelength and rotated around the ringlet for nodes to make wavelength reservation. The WR comprises four kinds of messages of the next frame: total number of available free time slots, denoted by E, the advertised fair rate, denoted by F_a, the status of slots in the ith wavelength, denoted by S_i, $1 \le i \le W$, and the CB_j relating to the DB_j, $1 \le j \le J$, where J is the number of the transit DBs at the next frame. Note that F_a is used to avoid the congestion and to achieve the fairness between each node by limiting the amount of the ingress traffic into ringlet. It will be transmitted to upstream nodes. The S_i contains S bits, and each bit in S_i represents the status of the corresponding slot with bit 0 (1) to indicate free (busy) in the wavelength λ_i at the next frame. The CB_j comprises information of DB_j such as its destination node, source node, wavelength number, start slot position, burst length in unit of slots, and service class.

■ *Node Architecture*

As shown in Fig. 4.2, each node i is in the slotted ring, $1 \le i \le N$, contains two optical add/drop multiplexers (OADMs), four pairs of optical-to-electronic (O/E) converter and electronic-to-optical (E/O) converter, a fiber delay line (FDL), ingress buffers, two egress buffers, a scheduler, and a central processor (CP) equipped with the proposed wavelength assignment and traffic control (WATC) scheme. The OADM is used to add (drop) the traffic to (from) the fiber links. The O/E converts the optical traffic to electronic domain, and the E/O inverses the conversion. The FDL is used to delay the transit traffic on data wavelengths to synchronize with the control signal. The buffer k in the ingress buffers of node i stores its ingress packets destined to node k, $1 \le k \le N, k \ne i$. It contains four completely-partitioned queues for voice, video, HTTP, and FTP, denoted by V$_o$, V$_i$, H, and F, respectively. The WATC in CP determines how to assign (reserve) the empty slots of the wavelength for the ingress traffic and to advertise the upstream nodes of traffic limitation in the next frame.

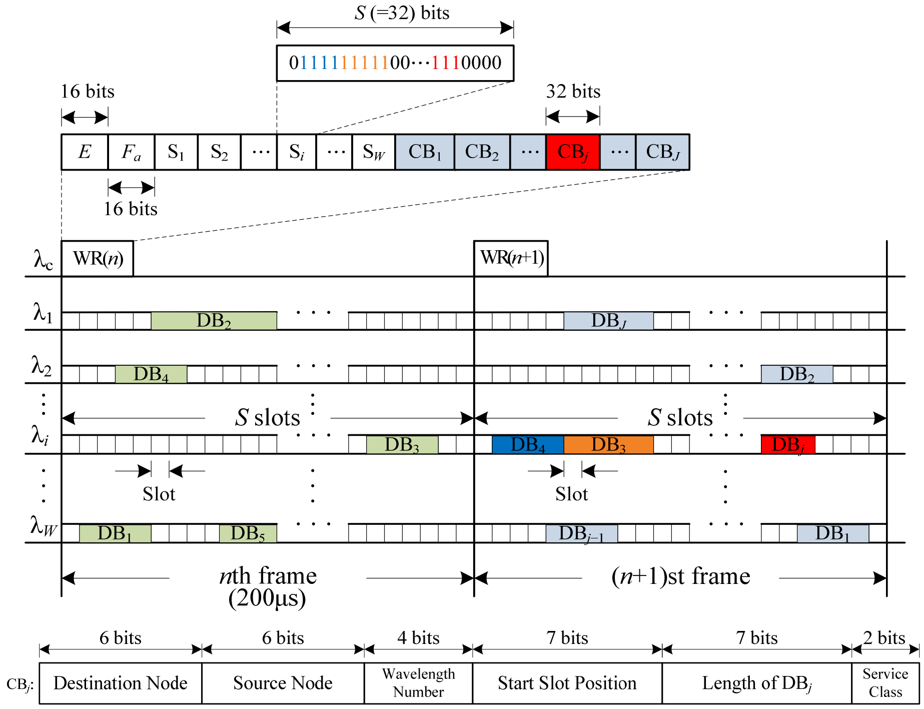


Fig. 4.1 Frame structure in slotted OBS rings

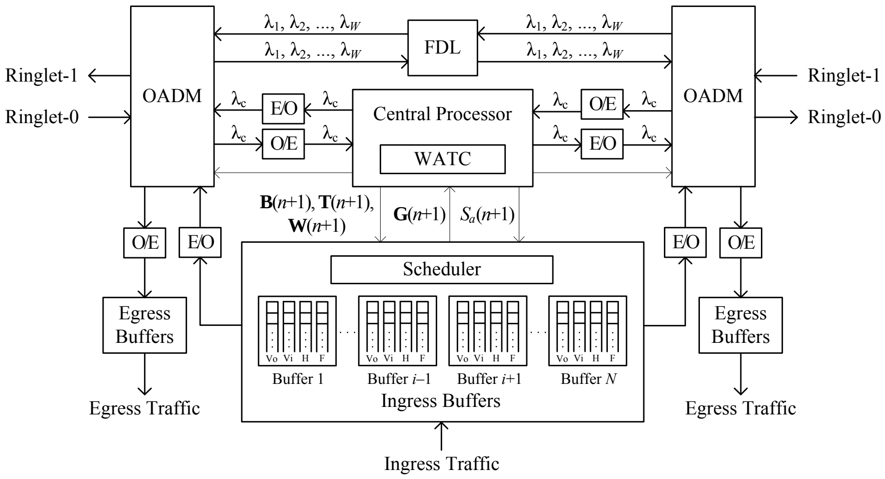


Fig. 4.2 Architecture of node *i* operating at the *n*th frame

Each node adds its ingress traffic via E/O converter onto the ringlet and bypasses transit traffic to the next node. The ingress packets are first put into ingress buffers according to their destinations and are stored in the different queue based on their service classes. Several packets in the same ingress buffer would be assembled into one DBs based on the result of the wavelength assignment by WATC. When DB from upstream nodes arrives at this node as its destination, it will be converted via O/E converter and stored in the egress buffer and be disassembled into original packets.

- *Scheduler*

The scheduler is informed the total number of available slots in the $(n+1)$st frame, denoted by $S_a(n+1)$, by WATC in CP at the nth frame. It determines the granted bandwidth of each ingress queue in unit slots based on a weighted round robin method.

Denoted, the weights of the V_o, V_i, H, and F ingress queues of node i, denoted by $w_{i,1}$, $w_{i,2}$, $w_{i,3}$, and $w_{i,4}$, respectively. Usually, $w_{i,1} > w_{i,2} > w_{i,3} > w_{i,4}$. However, if the HTTP traffic in the H ingress queue has the starvation problem, which is that the first packet of the H ingress queue exceeds its maximum tolerance delay time, the w_{i3} will be promoted to be equal to $w_{i,2}$. The granted bandwidth of the *j*th queue in the *k*th buffer at node *i*, denoted by $g_{k,j}(n+1)$, in unit slots at the (*n*+1)st frame is obtained by

$$g_{k,j}(n+1)=\min\left\{q_{k,j}(n),\left\lfloor \frac{w_{k,j}(n)}{\sum_{k}\sum_{j} w_{k,j}(n)}\cdot S_a(n+1)\right\rfloor\right\}\quad 1\le k\le N, k\ne i; 1\le j\le 4 \qquad (4.1)$$

where $q_{k,j}(n)$ is the length of the *j*th queue in the *k*th buffer at the *n*th frame at node *i* and $\lfloor x\rfloor$ is the larger integer less than *x*. The remaining bandwidth would be proportionally granted to those queues which are starvation. The granted bandwidths of all queues are arranged in a vector in order of their service classes. The granted bandwidth vector, denoted by **G**(*n*+1), is expressed as

$$\mathbf{G}(n+1)=[g_{1,1}(n+1),\ \ldots,\ g_{i-1,1}(n+1),\ g_{i+1,1}(n+1),\ \ldots,g_{N,1}(n+1),\ldots,\ g_{1,4}(n+1),\ \ldots,\ g_{i-1,4}(n+1),\ g_{i+1,4}(n+1),\ \ldots,\ g_{N,4}(n+1)], \qquad (4.2)$$

and sent to the WATC.

The WATC will determine how many and which available free time slots on which wavelength will be assigned to each queue according to the granted bandwidth vector **G**(*n*+1) and WR. As shown in Fig. 4.2, three vectors for operation in the next frame, **B**(*n*+1), **T**(*n*+1), and **W**(*n*+1), will be generated and sent back to the scheduler. The first vector **B**(*n*+1) is to indicate how many available time slots for each burst in all queues; the second vector **T**(*n*+1) is to indicate the which time slot each burst begins to transmit; the last vector **W**(*n*+1) is to indicate which wavelength each burst will be carried on . Three vectors **B**(*n*+1), **T**(*n*+1), and **W**(*n*+1) will be introduced in the next section.

■ *Reservation Scheme*

Whenever a node has packets to transmit, it must reserve appropriate slots in advance and then begins to transmit at the next frame if the reservation is successful. When the WR(n) arrives the node at the nth frame, the CP will calculate the scheduler the total number of available slots in the next frame $S_a(n+1)$, and inform immediately, at the time point t_1 as shown in Fig. 4.3. The scheduler would quickly compute the granted bandwidth vector $\mathbf{G}(n+1)$ and send it back to the WATC at the time point t_2 as shown in Fig. 4.3. The WATC will generate the three vectors, $\mathbf{B}(n+1)$, $\mathbf{T}(n+1)$, and $\mathbf{W}(n+1)$, to inform the Scheduler how to assemble the packets in each queue into data bursts and to transmit these bursts in appropriate time slots and wavelengths at the time point t_3. The CP will generate the WR(n+1) which is included the reserved time slots by this current node. Notice that the FDL shown in Fig. 4.2, whose length is equal to the processing time of the WATC to generate those vectors, is used to delay the frame time of outgoing link in order to synchronize the control burst in control wavelength and data burst in data wavelengths. In other words, the outgoing frame sent out of the node will begin at the time t_3. The detail timing sequence is shown in Fig. 4.3.

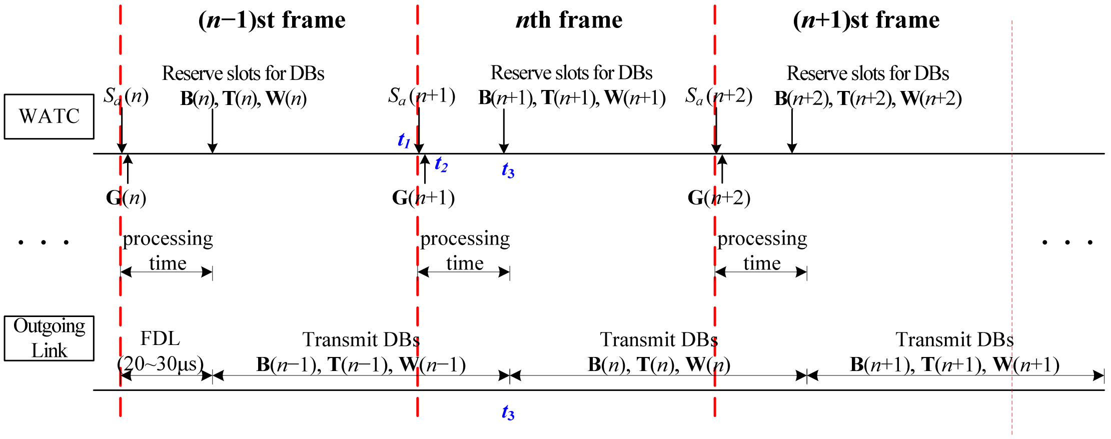


Fig. 4.3 Timing sequence in a node

■ *Fairness Reference Model*

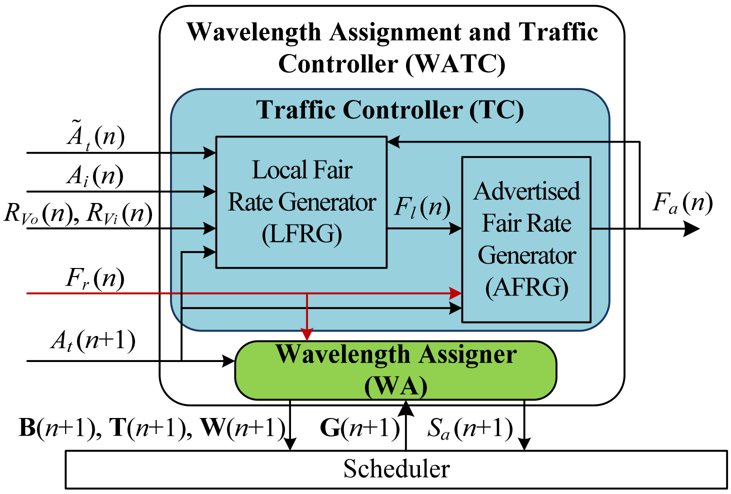
Here, the ring ingress aggregated with spatial reuse (RIAS) reference model is adopted to achieve fairness bandwidth allocation between the nodes. The available bandwidth in current link will be fairly allocated among all ingress aggregated (IA) flows, where IA flow represents the aggregation of all flows originating from a specific ingress node. The, RIAS model can ensure maximal spatial reuse.

➢ WAVELENGTH ASSIGNMENT AND TRAFFIC CONTROL

The proposed wavelength assignment and traffic controller (WATC), shown in Fig 4.4, is composed of a wavelength assigner (WA) to schedule the time slots in data wavelength for ingress DBs and a traffic controller (TC) to intelligently determine an advertised fair rate for traffic control. The WATC can make better use of link bandwidth in terms of throughput and fairness.

■ *Wavelength Assigner* (WA)

The WA not only grants the bandwidth to ingress DBs in queues according to the priority of the traffic but also assigns the proper void to the granted bandwidths of queues as much as possible. At the beginning of *n*th frame, the WA first computes the



Wavelength Assignment and Traffic Controller (WATC)
Traffic Controller (TC)
$\tilde{A}_t(n)$
$A_i(n)$
$R_{Vo}(n)$, $R_{Vi}(n)$
$F_r(n)$
$A_t(n+1)$
Local Fair Rate Generator (LFRG)
$F_l(n)$
Advertised Fair Rate Generator (AFRG)
$F_a(n)$
Wavelength Assigner (WA)
$\mathbf{B}(n+1)$, $\mathbf{T}(n+1)$, $\mathbf{W}(n+1)$
$\mathbf{G}(n+1)$
$S_a(n+1)$
Scheduler

Fig. 4.4 The structure of WATC

total number of available slots in the next frame, $S_a(n+1)$, which is the minimum of the slots not being reserved or the slots claimed by a received fair rate sent from the downstream node at the next frame. The $S_a(n+1)$ is given by

$$S_a(n+1) = \min\{\ C - A_t(n+1), F_r(n)\}, 0 \leq S_a(n+1) \leq C, \qquad (4.3)$$

where C is the link capacity in unit slots per frame, $A_\mathrm{t}(n+1)$ is the amount of the reserved slots by the upstream nodes (the arrival rate of transit traffic in unit slots) in the $(n+1)$st frame, and $F_r(n)$ is the received fair rate, which is the advertised fair rates sent from the downstream nodes and is obtained from the WR of the other ringlet. The $S_a(n+1)$ will be sent to the scheduler to generate the granted bandwidth $\mathbf{G}(n+1)$, as given eq.(4.2).

The WA will allocate slots to each queue according to its grated bandwidth received from the scheduler. The procedure of wavelength assignment for the kth buffer at node i in WA is shown in Fig. 4.5, $1 \leq k \leq N, k \neq i$. Initially, let M be the granted bandwidth $g_{k,j}$ of the jth queue in the kth buffer, and the WA searches for whether the length of possible voids is longer than $g_{k,j}$. If the voids exist, the WA will choose one of the smallest possible voids as the best position, denoted by (λ_b, t_b), for the queue j. Then, all packets in this queue will be assembled in to a DB, denoted by $b_{k,j}^1 = M$; the DB will be arranged to a wavelength, denoted by $w_{k,j}^1 = \lambda_b$, at the time point, denoted by $t_{k,j}^1 = t_b$. If the voids do not exist, WA will sequentially choose several possible voids to satisfy $g_{k,j}$. In other words, the packets in this queue will be assembled into several DBs. At first, the WA will choose the biggest void with length l_v as the best position (λ_b, t_b) for the queue. Also, the first assembled DB in this queue is with burst length $b_{k,j}^1 = l_v$, and will be transmitted on the wavelength $w_{k,j}^1 = \lambda_b$ at the time $t_{k,j}^1 = t_b$. Notice that the remaining granted bandwidth for the queue is $M - l_v$. Next, the WA does the same process until the remaining granted bandwidth is zero. For simplicity, at h cycle, if the WA searches the available void with length l_v at the position (λ_b, t_b) for the queue, the associated DB is denoted by $b^h_{k,j} = l_v$ and it will be transmitted on the wavelength $w^h_{k,j} = \lambda_b$ at the time $t^h_{k,j} = t_b$.

The size of each DB, which will be transmitted at the (n+1)st frames at node i is obtained as

$\mathbf{B}(n+1)=[b_{1,1}(n+1), \ldots, b_{i-1,1}(n+1), b_{i+1,1}(n+1), \ldots, b_{N,1}(n+1), \ldots, b_{1,4}(n+1), \ldots, b_{i-1,4}(n+1), b_{i+1,4}(n+1), \ldots, b_{N,4}(n+1)]$, (4.4)

$$b_{k,j}(n+1) = [b^1_{k,j}(n+1), b^2_{k,j}(n+1), \ldots b^{n_{kj}}_{k,j}(n+1)] \qquad (4.5)$$

where 0<$b^h_{k,j}(n+1) \le g_{k,j}$ (n+1), $\sum_{h=1}^{n_{kj}} b^h_{k,j}(n+1) = g_{k,\mathrm{j}}$ n_{kj} is the number of DBs for the jth queue in the kth buffer at the (n+1)st frames. The start time position relating to each DB and the used wavelength are obtained, respectively, as

$\mathbf{T}(n+1)=[t_{1,1}(n+1), \ldots, t_{i-1,1}(n+1), t_{i+1,1}(n+1), \ldots, t_{N,1}(n+1), \ldots, t_{1,4}(n+1), \ldots, t_{i-1,4}(n+1), t_{i+1,4}(n+1), \ldots, t_{N,4}(n+1)]$, (4.6)

$$t_{k,j}(n+1) = [t^1_{k,j}(n+1), t^2_{k,j}(n+1), \ldots t^{n_{kj}}_{k,j}(n+1)] \qquad (4.7)$$

$\mathbf{W}(n+1)=[w_{1,1}(n+1), \ldots, w_{i-1,1}(n+1), w_{i+1,1}(n+1), \ldots, w_{N,1}(n+1), \ldots, w_{1,4}(n+1), \ldots, w_{i-1,4}(n+1), w_{i+1,4}(n+1), \ldots, w_{N,4}(n+1)]$,

(4.8)

$$w_{k,j}(n+1) = [w^1_{k,j}(n+1), w^2_{k,j}(n+1), \ldots w^{n_{kj}}_{k,j}(n+1)] \qquad (4.9)$$

where $1 \le t^h_{k,j}(n+1) \le S$, $1 \le w^h_{k,j}(n+1) \le W$, $1 \le h \le n_{kj}$.

```
for $j \leftarrow 1$ to 4
    initialize $h \leftarrow 1$, $M \leftarrow g_{k,j}$
    while ($M \neq 0$)
        search for possible voids
        if (exist a possible void)      // case 1
            choosed one of the smallest possible voids as the best position ($\lambda_b$, $t_b$)
            set $b_{k,j}^h \leftarrow M$, $w_{k,j}^h \leftarrow \lambda_b$, $t_{k,j}^h \leftarrow t_b$, $M \leftarrow 0$
        else                            //case 2
            choose the biggest void $V$ with length $l_V$ as the best position ($\lambda_b$, $t_b$)
            set $b_{k,j}^h \leftarrow l_V$, $w_{k,j}^h \leftarrow \lambda_b$, $t_{k,j}^h \leftarrow t_b$, $h \leftarrow h+1$, $M \leftarrow M - l_V$
        end if
    end while
    $j \leftarrow j+1$
end for
```

Fig. 4.5 Pseudocode of wavelength assignment for the *k*th buffer at node *i*

Since the length of each void is an integer in the range from 0 to *S*, the counting sort could be used here to reduce the computational complexity because counting sort runs in time that is *linear* in the size of the inputs. Also, we adopt the binary search algorithm to search the possible voids because it is simple and its computational complexity is only *O*(lg*X*), where *X* is the number of the voids. Therefore, the computational complexity of WA to generate three vectors is $\mathrm{O}(X + L\log_2 X)$, where $L = \arg \max_{\substack{k \in \{1,2...N\},\, k \neq i \\ j=1,2,3,\text{and}4}} n_{kj}$.

■ *Traffic Controller*

As shown in Fig. 4.4, the TC further contains the local fair rate generator (LFRG) to generate a local fair rate, denoted by $\mathrm{F}_l(n)$, which satisfies the RIAS fairness, and the advertised fair rate generator (AFRG) to decide the advertised fair rate, denoted by $\mathrm{F}_a(n)$, referring to the local fair rate $\mathrm{F}_l(n)$ and the received fair rate $\mathrm{F}_r(n)$. The design of the LFRG imitates the generation of local fair rate by FLAG, but it additionally consider the protective reservation bandwidth for future voice and video ingress traffic to avoid or decrease the voice and video packet loss. The AFRG generates F$_a$(n) to inform the upstream nodes of the traffic limitation for effectively utilizing the bandwidth and enhancing the fairness between nodes.

- *Local Fair Rate Generator* (*LFRG*)

The local fair rate generator (LFRG) adopts a fuzzy protective reservation generator (FPRG) on the protective reservation bandwidth for voice and video traffic, trying to decrease the packet loss of the voice and video to guarantee the quality of service (QoS). The LFRG imitates the FLAG to generate the local fair rate. It is due to the fact that the FLAG can achieve the RIAS fairness and have better preference than DBA. The LFRG contains a fuzzy protective reservation generator (FPRG) and a fair rate generator (FRG), shown in Fig. 4.6.

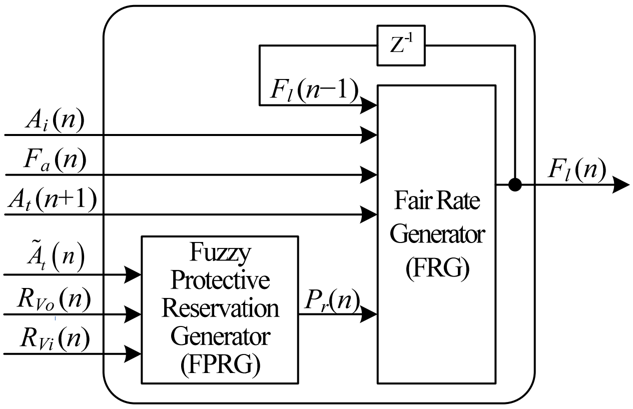


Fig. 4.6 The structure of LFRG

- Fuzzy Protective Reservation Generator (FPRG)

In our considered network, the transit traffic will bypass current node directly without any O/E and E/O conversions. Therefore, once the slot is fully used by upstream nodes, current node would hardly have bandwidth to transmit its own data, even if the service class of the data is more important than those of the transit traffic. Although this problem can be alleviated by the advertised fair rate, it takes much propagation time to see the effect. In order to further alleviate the mentioned problem, we protectively reserve some bandwidth, denoted by $P_r(n)$, in advance for future voice and video traffic in ingress nodes.

Here, we use a fuzzy inference system to determine the protective reservation. FPRG has three inputs: the moving average of transit traffic, denoted by $\tilde{A}_t(n)$, the ratio of moving average of voice transit traffic to moving average of total transit traffic at the nth frame, denoted by $R_{Vo}(n)$, and the ratio of moving average of video transit traffic to moving average of total transit traffic at the nth frame, denoted by $R_{Vi}(n)$. $\tilde{A}_t(n)$ is obtained by

$$\tilde{A}_t(n) = \frac{1}{L}\sum_{k=n-L+1}^{n} A_t(k), \tag{4.10}$$

where $A_t(n)$ is the arrival rate of transit traffic in unit slots at the nth frame, W is the observation window. In other words, $\tilde{A}_t(n)$ is a moving average of $A_t(n)$. The $R_{Vo}(n)$ and $R_{Vi}(n)$ are obtained by

$$R_{Vo}(n) = \frac{\tilde{A}_t^{Vo}(n)}{\tilde{A}_t(n)},\ R_{Vi}(n) = \frac{\tilde{A}_t^{Vi}(n)}{\tilde{A}_t(n)},\ 0 \le R_{Vo}(n) \le 1,\ 0 \le R_{Vi}(n) \le 1, \tag{4.11}$$

respectively, where $A_t^{Vo\,(Vi)}(n)$ is the arrival rate of voice (video) transit traffic in unit slots at the nth frame. The output of FPRG is $P_r(n)$, which is the protective reservation in unit slots at the nth frame for future voice and video traffic in ingress node.

We define the term set for $\tilde{A}_t(n)$ as $T(\tilde{A}_t(n))$={Low (L), Medium (M), High (H)}; for $R_{Vo}(n)$ as $T(R_{Vo}(n))$= {Low (L), High (H)}; for $R_{Vi}(n)$ as

$T(R_{Vi}(n))$= {Low (L) , Medium (M), High (H)}; for $P_r(n)$ as $T(P_r(n))$= {Low (L) , Medium (M), High (H)}.

There are 18 fuzzy rules for FPRG. As shown in Table I, the order of significance of the input linguistic variables is $\tilde{A}_t(n)$, $R_{Vo}(n)$, and $R_{Vi}(n)$. The node with more moving average of transit traffic, which is composed by the high voice or video traffic would be reserved more bandwidth to avoid that the voice or video traffic at this node could not be transmitted due to no reservation bandwidth.

TABLE 4.1
THE RULE BASE OF FPRG

Rule	$\tilde{A}_t(n)$	$R_{\mathrm{Vo}}(n)$	$R_{\mathrm{Vi}}(n)$	$P_r(n)$	Rule	$\tilde{A}_t(n)$	$R_{\mathrm{Vo}}(n)$	$R_{\mathrm{Vi}}(n)$	$P_r(n)$
1	*L*	*L*	*L*	*L*	10	*M*	*H*	*L*	*L*
2	*L*	*L*	*M*	*L*	11	*M*	*H*	*M*	*L*
3	*L*	*L*	*H*	*VL*	12	*M*	*H*	*H*	*VL*
4	*L*	*H*	*L*	*VL*	13	*H*	*L*	*L*	*VH*
5	*L*	*H*	*M*	*N*	14	*H*	*L*	*M*	*VH*
6	*L*	*H*	*H*	*N*	15	*H*	*L*	*H*	*H*
7	*M*	*L*	*L*	*H*	16	*H*	*H*	*L*	*H*
8	*M*	*L*	*M*	*M*	17	*H*	*H*	*M*	*M*
9	*M*	*L*	*H*	*M*	18	*H*	*H*	*H*	*M*

We use the triangular function $f(x;\, x_0,\, a_0,\, a_1)$ and the trapezoidal function $g(x;\, x_0,\, x_1,\, a_0,\, a_1)$ to define the membership functions for terms in the term set. These two functions are given by

$$f\left(x;x_0,a_0,a_1\right)=\begin{cases}(x-x_0)/a_0+1, & \text{if } x_0-a_0<x\le x_0,\\ (x_0-x)/a_1+1, & \text{if } x_0<x<x_0+a_1,\\ 0, & \text{otherwise,}\end{cases} \tag{4.12}$$

$$g\left(x;x_0,x_1,a_0,a_1\right)=\begin{cases}(x-x_0)/a_0+1, & \text{if } x_0-a_0<x\le x_0,\\ 1, & \text{if } x_0<x\le x_1,\\ (x_1-x)/a_1+1, & \text{if } x_1<x<x_1+a_1,\\ 0, & \text{otherwise,}\end{cases} \tag{4.13}$$

respectively, 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; a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

The corresponding membership functions of L, M, and H in $T(\tilde{A}_t(n))$ are denoted by $\mu_L(\tilde{A}_t(n)) = g\,(\tilde{A}_t(n);\, 0,\, 0.45C,\, 0,\, 0.2C)$, $\mu_M(\tilde{A}_t(n)) = f\,(\tilde{A}_t(n);\, 0.65C,\, 0.2C,\, 0.2C)$, $\mu_H(\tilde{A}_t(n)) = g\,(\tilde{A}_t(n);\, 0.85C,\, C,\, 0.2C,\, 0)$, where C is the channel capacity in unit slots. The corresponding membership functions of L and H in $T(R_{Vo}(n))$ are denoted by $\mu_L(R_{Vo}(n)) = g\,(R_{Vo}(n);\, 0,\, 0.3,\, 0,\, 0.2)$, $\mu_H(R_{Vo}(n)) = g\,(R_{Vo}(n);\, 0.5,\, 1,\, 0.2,\, 0)$. The corresponding membership functions of L, M, and H in $T(R_{Vi}(n))$ are denoted by $\mu_L(R_{Vi}(n)) = g\,(R_{Vi}(n);\, 0,\, 0.2,\, 0,\, 0.2)$, $\mu_M(R_{Vi}(n)) = g\,(R_{Vi}(n);\, 0.4,\, 0.5,\, 0.2,\, 0.2)$, $\mu_H(R_{Vi}(n)) = g\,(R_{Vi}(n);\, 0.7,\, 1,\, 0.2,\, 0)$. For the reason of simplicity in computation of defuzzification, let the membership functions for N, VL, L, M, H, VH in $T(P_r(n))$ be fuzzy singletons. Define these membership functions by $\mu_N(P_r(n)) = f\,(P_r(n);\, 0,\, 0,\, 0)$, $\mu_{VL}(P_r(n)) = f\,(P_r(n);\, 10,\, 0,\, 0)$, $\mu_L(P_r(n)) = f\,(P_r(n);\, 20,\, 0,\, 0)$, $\mu_M(P_r(n)) = f\,(P_r(n);\, 30,\, 0,\, 0)$, $\mu_H(P_r(n)) = f\,(P_r(n);\, 40,\, 0,\, 0)$, $\mu_{VH}(P_r(n)) = f\,(P_r(n);\, 50,\, 0,\, 0)$. The membership functions of $T(P_r\,(n))$ are discrete uniformly distributed between 0 and 50.

FPRG adopts the max-min inference method for inference engine. To explain max-min inference method, we consider rule 5 and rule 6 which have the same control action "P_r (n) is N." Applying "min" operator, we obtain the membership function values of the control action "P_r (n) is N" of rule 5 and rule 6 respectively by

$$m_5(n) = \min\{\mu_L(\tilde{A}_t(n)), \mu_H(R_{Vo}(n)), \mu_M(R_{Vi}(n))\}, \tag{4.14}$$

$$m_6(n) = \min\{\mu_L(\tilde{A}_t(n)), \mu_H(R_{Vo}(n)), \mu_H(R_{Vi}(n))\}, \tag{4.15}$$

Then, applying the “max” operator, we obtained the overall membership value of the control action “P_r (n) is N” , denoted as $w_N(n)$, by

$$w_N(n) = \max\{m_5(n), m_6(n)\}. \tag{4.16}$$

Similarly, the overall membership function of the control action VL, L, M, H, and VH, denoted as $w_{VL}(n)$, $w_L(n)$, $w_M(n)$, $w_H(n)$, and $w_{\mathrm{VH}}(n)$, respectively, can be obtained. After inferring all the rules, FPRG uses center of area (COA) method for defuzzifier. The output value Pr (n) is obtained by

$$P_r(n) = \frac{0 \times w_N(n) + 10 \times w_{VL}(n) + 20 \times w_L(n) + 30 \times w_M(n) + 40 \times w_H(n) + 50 \times w_{VH}(n)}{w_N(n) + w_{VL}(n) + w_L(n) + w_M(n) + w_H(n) + w_{VH}(n)}. \tag{4.17}$$

- Fair Rate Generator (FRG)

Here, we define a parameter, denoted by $M(n)$, to measure the equivalent number of IA transit flows traversing node i , whose rate is equal to previous advertised fair rate. This parameter is determined by

$$M(n) = \frac{A_i(n) + \tilde{A}_t(n)}{\tilde{F}_a(n-1)}, \tag{4.18}$$

where $A_i(n)$ is the arrival rate of ingress traffic in unit slots at the nth frame, $A_i(n) = \sum_{\substack{k=1 \\ k \neq i}}^{N} \sum_{j=1}^{4} g_{k,j}(n+1)$, is the granted bandwidth of the jth queue in the kth buffer in node i at the (n+1)st frames; $\tilde{F}_a(n-1)$ is the moving average of $F_a(n-1)$, which is the advertised fair rate generated at the (n–1)st frame, $\tilde{F}_a(n-1) = \frac{1}{L}\sum_{k=n-L}^{n-1} F_a(k)$. The local fair rate at the nth frame is calculated as

$$F_l(n) = \min\left\{C - P_r(n),\ F_l(n-1) + \frac{1}{M(n)}(C - P_r(n) - (A_l(n) + A_t(n+1)))\right\}, \quad (4.19)$$

where $P_r(n)$ is the protective reservation in unit slots at the *n*th frame for future voice and video traffic in ingress node.

- *Advertised Fair Rate Generator (AFRG)*

The traffic condition in the fiber link is always changing. In order to alleviate the overuse of a link as well as make better use of the link capacity, AFRG is design to adapt the IA flows. In order to decide the traffic condition, AFRG observes the incoming transit traffic, say $A_t(n+1)$, and compares it with the minimum of the local fair rate $F_l(n)$ and the received advertised fair rate from the other ringlet, called $F_r(n)$. The main reason of using minimum operation is to be a little more conservative in order not to incur overuse of a link too often. Depending on the traffic conditions, AFRG takes two possible actions.

- Traffic Suppression

If $A_t(n+1)$ is bigger than or equal to the minimum of $F_l(n)$ and $F_r(n)$, the link is considered as overused. Therefore, AFRG will suppress the traffic and decrease the rates of IA flows. The advertised fair rate $F_a(n)$ will choose the minimum of $F_l(n)$ and $F_r(n)$ given by

$$F_a(n) = \min\{F_l(n),\ F_r(n)\}. \quad (4.20)$$

- Traffic Promotion

If $A_t(n+1)$ is smaller than the minimum of $F_l(n)$ and $F_r(n)$, the link is considered as not sufficiently used. Therefore, AFRG will promote the traffic and increase the rates of IA flows. The advertised fair rate $F_a(n)$ will choose the maximum of $F_l(n)$ and $F_r(n)$ given by

$$F_a(n) = \max\{F_l(n),\ F_r(n)\}. \quad (4.21)$$

- **Simulation Results and Discussions**

In the simulations, settings for the environment include 10 Gbps link capacity, 200μs propagation delay between nodes, 200μs frame time, and 7 data wavelengths (*W*), and one control wavelength per ringlet. A frame time contains 125 time slots per wavelength, and a bandwidth per slot is 2.0 k bits. Four traffic service classes, voice video, HTTP, FTP, are considered and their corresponding weights are 4, 3, 2, and 1, respectively. Simulations for the proposed WATC and the distributed bandwidth allocation (DBA) [13] are conducted for performance comparison. Simulation results are recorded per frame time to show the bandwidth usage on a congested link.

Fig. 4.7(a) shows a parking lot scenario where there are 8 (1~8) greedy nodes with four types of traffic. Note that this parking lot scenario assumes that flows with four types of traffic are generated from node 1 to node 7 but terminated at node 8. The propagation delay is long. Fig. 4.7(b) shows the throughputs of the WATC and the group scheduling with traffic controller (GS_TC) at the output link of node 7, and Fig. 4.7(c) shows the throughputs of the wavelength assigner with DBA (WA_DBA) and the group scheduling with DBA (GS_DBA) at the output link of node 7. It can be seen in Fig. 4.7(b) that the throughput of the WATC quickly approaches to 100% of the link capacity; while the throughput of the GS_TC achieves 95% of the link capacity as the throughput stabilizes. In other words, the GS_TC suffers some oscillation at the beginning. It is due to the fact that the GS_TC only searches and utilizes the earliest possible voids. The GS would leave a DB, whose length is equal to the granted bandwidth of a queue, behind in the queue unless it finds a proper void for this DB. Also, the bursts are bigger at the beginning while the received fair rate is larger. Since the bursts are too long to find available voids for them, the throughput of the GS_TC is not stable. Instead, the WATC would divide the DB into several smaller pieces of DB and assign each of them an approximate void. Furthermore, it is found in Fig.

4.7(c) that the WA_DBA has higher throughput than GS_DBA which also suffers from throughput oscillation. The reason is that the variation of the regulated (advertised) fair rate generated by DBA is large when the propagation delay is long and GS cannot fully utilize the bandwidth. These indicate that the WA performs better than GS no matter what traffic control scheme, TC or DBA, are adopted.

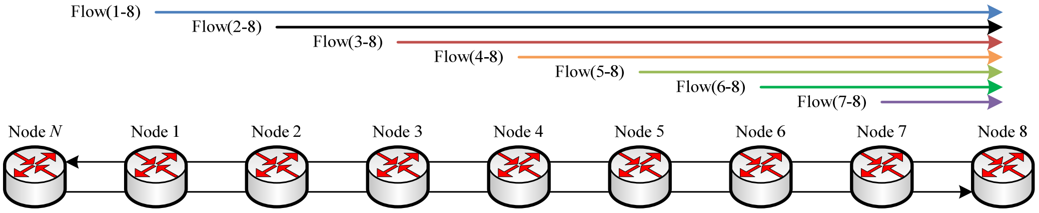


Figure 4.7(a): Large parking lot scenario

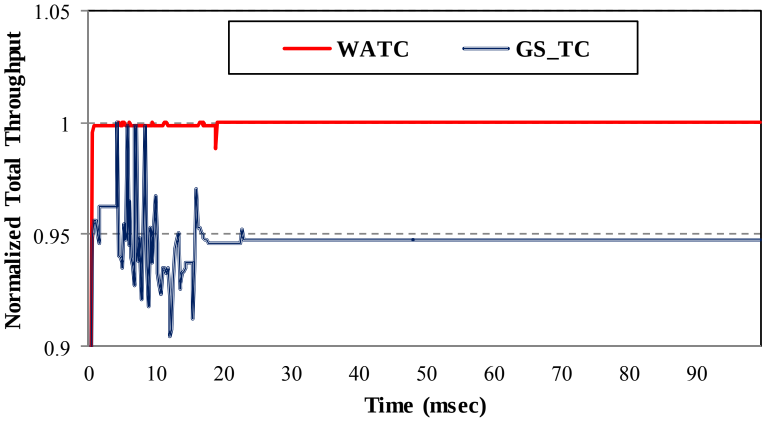


Figure 4.7(b): The normalized throughputs of WATC and GS_TC

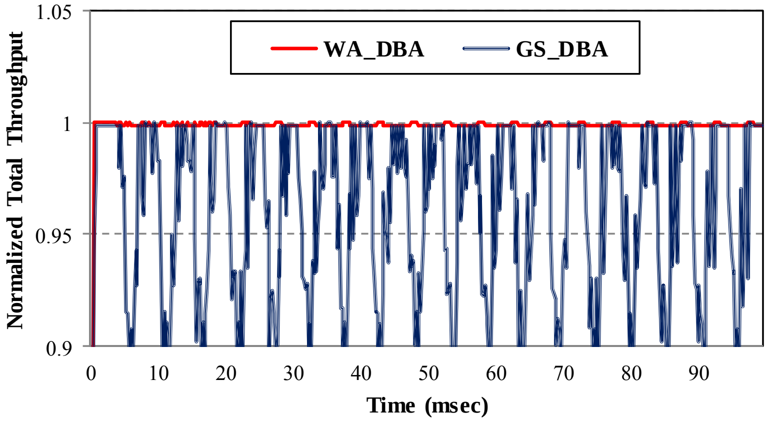


Figure 7(c): The normalized throughputs of WA_DBA and GS_DBA

Figs. 4.8(a) and 4.8(b) present throughputs of flow(1, 8), flow(3, 8), flow(5, 8), and flow(7, 8) of WATC and WA_DBA, respectively, in a parking lot scenario which contains 8 nodes as in Fig. 4.7(a). It can be seen that all flows of WATC takes about 21ms to stabilize but all flows of WA_DBA is hard to do. It is due to the fact that the WATC can diminish the effect of the propagation delay by using the moving average technique. On the other hand, the WA_DBA computes the number of effective IA flows referring to both the short-term aggregating traffic (a frame time) and the previous local fair rate to generate the current local fair rate. However, due to the large propagation delay, the correlation between the short-term traffic and the previous local fair rate becomes low. Therefore, the WA_DBA cannot generate a correct local fair rate to regulate the flows. As a result, the flows oscillate and hardly converge.

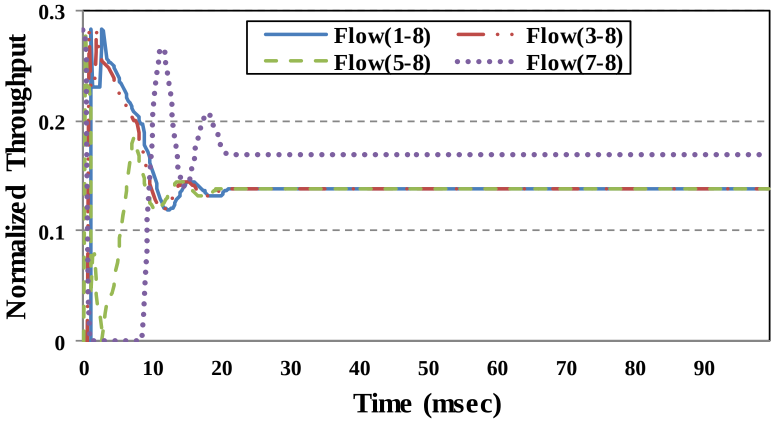


Figure 4.8(a): The normalized throughputs of WATC

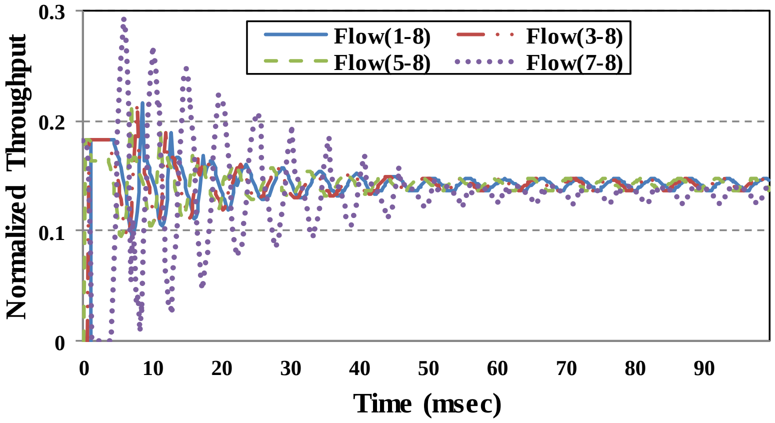


Figure 4.8(b): The normalized throughputs of WA_DBA

Figs. 4.9(a) and 4.9(b) present throughputs of flow(1, 8), flow(3, 8), flow(5, 8), and flow(7, 8) at node 8 by WATC and WA_DBA, respectively, in a parking lot scenario which contains 8 nodes as in Fig. 4.7(a), but each node has with various traffic demands with constant-bit-rate traffic model for voice and the exponentially distributed traffic model for the video, HTTP, and FTP. It can be seen that the WATC takes 22.9ms to stabilize all flows, while the WA_DBA cannot make the flows stable. The slotted OBS ring used one-way reservation signaling protocol to reserve the bandwidth. If the reserved bandwidth could not fully be used, the utilization of the link would be decreased. Since the WATC protectively reserves some bandwidth $P_r(n)$

for the voice and video traffic by the FPRG, the regulated bandwidth by the received fair rate for the HTTP and FTP traffics would not be sacrificed for the voice and video traffic. In other words, the regulated bandwidth could be almost utilized by the HTTP and FTP traffics, and this will reduce the starvation of the HTTP and FTP traffic and decrease the traffic oscillation. Besides, the WATC also could effectively diminish the effect of the propagation delay. On the other hand, the WA_DBA cannot generate correct fair rate as the propagation delay is large and the voice and video traffic are with the highest priority. Therefore, the flows by WA_DBA are with the great oscillations.

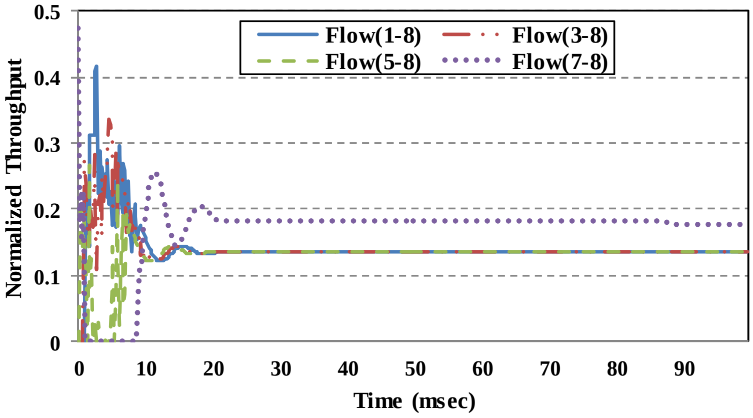


Figure 4.9(a): The normalized throughputs of WATC

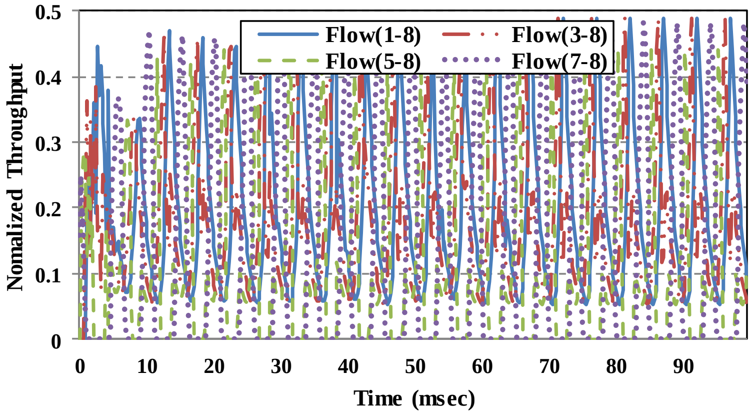


Figure 4.9(b): The normalized throughputs of WA_DBA

Figs. 4.10(a) and 4.10(b) present throughputs of flow(1, 8), flow(3, 8), flow(5, 8), and flow(7, 8) at node 8 by WATC and WA_DBA, respectively, in a parking lot scenario, which contains 8 nodes, as in Fig. 4.7(a), but with *finite* traffic demands and four types constant-bit-rate traffic. Assume that flow(1, 8), flow(2, 8), flow(4, 8), flow(5, 8), flow(6, 8), and flow(7, 8) require 1.0 Gpbs but flow(3, 8) requires 12 Gbps. It would be facts that node 4 would be the first one to incur congestion, and the traffic of node 1 and node 2 cannot always match its received fair rate due to the light finite traffic demands, while the other nodes are easily in congestion because of the heavy traffic demand of the node 3. It can be seen that the WATC takes about 15.8ms to stabilize all flows but the WA_DBA cannot make the flows stable. Moreover, the WATC stably realizes the RIAS fairness and has higher throughput by about 5.1% than WA_DBA. It is because that WATC protectively reserves a reasonable bandwidth by using the fuzzy logic system in the FPRG to avoiding the starvation of the HTTP and FTP traffic and reduce their variance, and indeed diminishes the effect of the propagation delay by using the moving technology. Therefore, it could generates a correct local fair rate at each frame. Also, since each traffic flow, except the flow(3, 8), is much less than that of the greedy case in Fig. 4.8(a), the damping amplitude is smaller than that in Fig. 4.8(a). On other hand, the generation accuracy of local fair rate in the WA_DBA is susceptible to the effect of the propagation delay and node 3 is with the largest traffic demand and far from node 7. Therefore, the result, as in Fig. 10(b), proves the fact that the flow(3, 8) cannot quickly be regulated by the node 7. This violent varying aggregation traffic per frame thus makes that WA_DBA cannot generate the local fair rate properly.

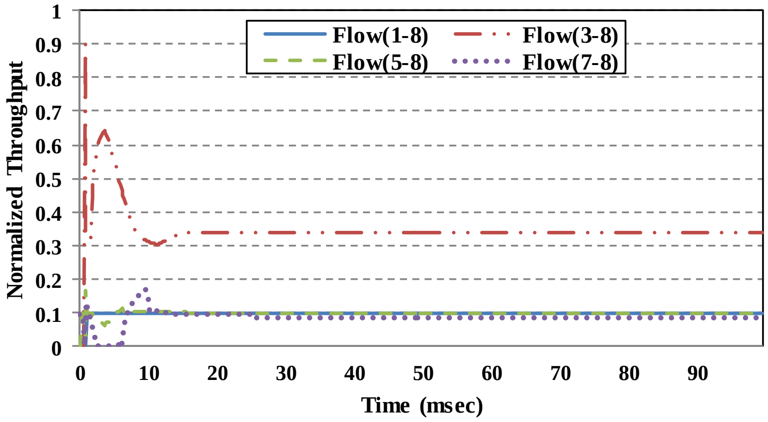


Figure 4.10(a): The normalized throughputs of WATC

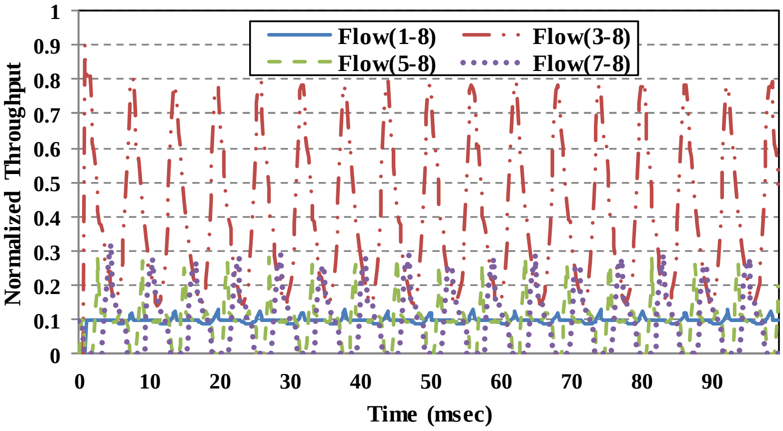


Figure 4.10(b): The normalized throughputs of WA_DBA

V. Reference

[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] T. Battestilli and H. Perros, “An introduction to optical burst switching,” *IEEE Commun. Mag.*, Vol. 41, No. 8, pp. S10–S15, Aug. 2003.

[7] IEEE Standard 802.17, “Resilient packet ring (RPR) access method and physical layer specifications,” 2004.

[8] 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.

[9] 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.

[10] M. Maier, M. Herzog, M. Scheutzow, and M. Reisslein, “Protectoration: 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.

[11] 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.

[12] F. Davik, A. Kvalbein, and S. Gjessing, “An analytical bound for convergence of the resilient packet ring aggressive mode fairness algorithm,” *IEEE ICC’05*, pp. 281–2878, 2005.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] F. Alharbi and N. Ansari, "Distributed bandwidth allocation for resilient packet ring networks," *Comp. Networks*, Vol. 49, No. 2, pp. 161–171, Oct. 2005.

[18] F. Alharbi and N. Ansari, "SSA: simple scheduling algorithm for resilient packet ring networks," *Computer NetworksIEE Proceedings Communications*, Vol. 153, No. 2, pp. 183–188, Apr. 2006.

[19] F. Davik, A. Kvalbein, and S. Gjessing, "Congestion Domain Boundaries in Resilient Packet Rings," *Tech. Rep., Simula Research Laboratory*, Feb. 2005.

[20] M. Yilmaz and N. Ansari, "Weighted Fairness in Resilient Packet Rings," *IEEE ICC'09*, 2007.

[21] 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.

[22] 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.

[23] 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.

[24] G. R´etv´ari, J. J. B´ır´o, and T. Cinkler, "On shortest path representation," *IEEE/ACM Trans. Networking*, Vol. 15, No. 6, pp. 1293–1306, Dec. 2007.

[25] 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.

[26] 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.

[27] L. Xu, H. G. Perros, and G. N. Rouskas, "A Simulation Study of Optical Burst Switching and Access Protocols for WDM Ring Network," *Computer Networks*, vol. 41, no. 2, pp. 143-160, Feb. 2003.

[28] Y. Arakawa, N. Yamanaka, and I. Sasase, "Performance of Optical Burst Switched WDM Ring Network with TTFR System," *International Federation*

for Information Processing (IFIP'05), vol. 164, pp. 95–102, 2005.

[29] L. M. Peng, B. B. Li, W. H. Yang, Y. C. Kim, and X. W. Li, "Novel Queue-Grouping Based Round-Robin Protocols for Reducing Collision of OBS Ring Networks," *International Conference on Networking and Services (ICNS'09)*, pp. 97-101, 2009.

[30] K. Ramantas, K. Vlachos, Ó. G. de Dios, and C. Raffaelli, "Window-based Burst Assembly Scheme for TCP Traffic over OBS," *IEEE/OSA Journal of Optical Networking*, vol. 7, no. 5, pp. 487-495, Apr. 2008.

[31] J. Kim, M. Maier, T. Hamada, and L. G. Kazovsky, "OBT: Optical Burst Transport in Metro Area Networks," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 44-51, Nov. 2007.

[32] L. M. Peng, K. M. Yoo, K. E. Han, and Y. C. Kim, "Design and Performance Comparison of Multiple-token Based MAC Protocols for Optical Burst Switched Ring Networks," *Photonic Network Communications*, vol. 15, no.3, pp. 213-225, Jun. 2008.

[33] W. P. Chen, W. F. Wang, and W. S. Hwang, "A Novel and Simple Beforehand Bandwidth Reservation (BBR) MAC Protocol for OBS Metro Ring Networks," *Journal of High Speed Networks*, pp. 59-72, Jan. 2008.

[34] H. J. Lin, Utility-based Fuzzy Wavelength Assignment in OBS Network. Master Thesis, 2009.

[35] Y. Xiong, M. Wandenhoute, and H. C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838-1851, Oct 2000.

[36] J. Li, C. Qiao, and Y. Chang, "Recent Progress in the Scheduling Algorithms in Optical-Burst-Swtiched Networks," *Journal of Optical Networking*, vol. 3, no. 4, pp. 229-241, Mar. 2004.

[37] R. G. Cheng and C. J. Chang, "Design of a Fuzzy Traffic Controller for ATM Networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 460–469, Jun. 1996.

[38] W. S. Tang and C. J. Chang, "Intelligent Interring Route Control in Bridged Resilient Packet Rings," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 24, pp. 5631–5638, Dec. 2009.

IV. Comment

◆ FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Ring

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

◆ Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings

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 deter mine a proper ringlet for an incoming new call. The design philosophy of IIRC can be applied to any kind of bridged optical packet rings..

◆ Reservation Slotted OBS Rings with Wavelength Assignment and Traffic Control

A wavelength assignment and traffic control (WATC) scheme is proposed for slotted OBS ring. It can increase the link utilization, satisfy RIAS fairness criterion, make flows converges to the fair rate, and guarantee stability. It is sophisticatedly composed of two function blocks: a wavelength assigner (WA) and a traffic controller (TC). The WA effectively schedules the ingress DBs in the wavelength closed to the granted bandwidth and the TC uses the local fair rate generator (LFRG) and the advertised fair rate generator (AFRG) to decide a properly advertised fair rate. The LFRG generates a local fair rate not only satisfying the RIAS fairness but also diminishing the effect of the propagation delay; the AFRG decides the advertised fair rate referring to the local fair rate generated by LFRG. Moreover, the LFRG pre-reserves some bandwidth for the voice and video traffic to avoid their dropping rate or the starvation of the HTTP and FTP. Simulation results show that WA indeed outperforms GS, and TC can stabilize all flows but DBA could not.

The perfomances of this proposal are as follows:

Jonual:

1. W. S. Tang, C. J. Chang, P. L. Tien, and W. C. Wang, "FLAG: A Fuzzy Local FairRate Generator for Resilient Packet Rings," *IEEE/OSA Journal of Optical Communications and Networking*, Vol. 2, No. 5, pp. 230–240, April 2010

2. W. S. Tang and C. J. Chang, "Intelligent Inter-Ring Route Control in Bridged Resilient Packet Rings," *IEEE/OSA Journal of Lightwave Technology*, Vol. 27, No. 24, pp. 5631–5638, Dec. 2009.

Conference:

1. W.-S. Tang, Y.-T. Hsieh, C. J. Chang, and W.-C. Chung “A Wavelength Assignment and Traffic Control Scheme in Reservation Slotted OBS Rings,” *Proc. CHINACOM 2011*, Aug. 17-19, 2011, Harbin, China.

2. W. S. Tang and C. J. Chang, “A Fuzzy Inter-Ring Route Control with PRNN Predictor Bridged Resilient Packet Rings,” *Proc. IEEE Globecom’10*, Dec. 2010.

3. W. S. Tang, W. C. Wang, C. J. Chang, and Y. T. Hsieh, “An Effective Fair Bandwidth Allocation in Resilient Packet Ring,” *Proc. ChinaCom 2010, Aug. 2010* (with Best Paper Award)

4. W. S. Tang, C. J. Chang, and S. H. Liu, "A Fuzzy Fairness Control Mechanism in Resilient Packet Ring," *Proc. ChinaCom 2009*, Xian, China