

# **Bandwidth Allocation and Fairness in Resilient Packet Rings**

by

Arash Shokrani-Chaharsoughi

B.Sc., Isfahan University of Technology, 1996

M.Sc., Isfahan University of Technology, 1999

A thesis submitted to the faculty of Graduate Studies and Research  
in partial fulfilment of the requirements for the degree of

**Doctor of Philosophy**

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Department of Systems and Computer Engineering  
Faculty of Engineering  
Carleton University  
Ottawa, Ontario

December 2006

© Copyright by Arash Shokrani-Chaharsoughi 2006



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-23300-9*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-23300-9*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# Abstract

This thesis investigates different aspects of bandwidth allocation and fairness in Resilient Packet Rings (RPR). RPR supports spatial reuse which increases throughput of the ring, however, it can result in unfairness in bandwidth allocation throughout the ring. Therefore, a rate-based fairness algorithm is employed at each RPR node to maintain fairness among nodes while achieving high throughput. We start by developing a new fairness reference model for RPR which can be used to examine fairness of a set of rate allocations. We show that our proposed fairness reference model is a general form of an existing one and allows each node to implement its own policy in distributing its share of bandwidth among its local flows to achieve different objectives. We then study and evaluate the performance of different policies which can be employed by an RPR node to divide its bandwidth share among its local flows. Afterwards, we model the fair rate calculation process in the RPR standard bandwidth allocation algorithms. The developed models provide insights on dynamics of these algorithms. We also study and analyze the effect of different system parameters on the performance of the these algorithms through analytical results. Finally, we propose a new bandwidth allocation algorithm for RPR called Virtual Queueing (VQ) which aims at stabilizing the virtual queue model of each RPR node. We show that the RPR standard bandwidth allocation algorithms suffer from severe performance limitations. We also study the performance of another RPR bandwidth allocation algorithm called DVSR and show that this algorithm also exhibits low performance characteristics in a dynamic traffic environment. We then show that VQ outperforms these algorithms with lower computational complexity.

# Acknowledgements

For their continued support and encouragement, I would like to express my appreciation and gratitude to my supervisors Dr. Ioannis Lambadaris and Dr. Jérôme Talim. This work could never have been completed without their guidance and advice.

I am thankful to the members of the examining board of my thesis, Dr. Ioannis Viniotis of North Carolina State University, Dr. Jiyang Zhao of University of Ottawa, Dr. Evangelos Kranakis of the School of Computer Science, and Dr. Roshdy Hafez and Dr. Peter Liu of the Department of Systems and Computer Engineering, for their helpful comments and suggestions.

My gratitude goes to Dr. Hossein Saidi of Isfahan University of Technology for his support and encouragement during my graduate studies. I also wish to thank all my friends especially Pirouz Zarrinkhat, Kayvan Mosharaf, Mohammadreza Yazdani, Kamran Farzan, Saied Hemati, and Ashraf Matrawy for providing me the moral support during my time at Carleton. As well, I thank all helpful staffs of our department especially Blazenka Power, Anna Lee, Darlene Hébert, and Narendra Mehta.

I am also thankful to Communications and Information Technology Ontario (CITO), Mathematics of Information Technology and Complex Systems (MITACS), the Government of Ontario, Carleton University, and Nortel Networks for providing the financial support of this work.

Last, and definitely most, I want to thank my family for their constant love and support. I cherish them and I am always grateful to them.

*To my parents*

# Table of Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Notations</b>	<b>xiv</b>
<b>List of Abbreviations</b>	<b>xviii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivations . . . . .	3
1.2 Contributions . . . . .	4
1.3 Organization of Thesis . . . . .	5
<b>2 Preliminaries and Background</b>	<b>7</b>
2.1 Overview of Medium Access Methods for Ring Networks . . . . .	8
2.1.1 Ring Networks without Spatial Reuse . . . . .	8
2.1.2 Ring Networks with Spatial Reuse . . . . .	9
2.2 Overview of IEEE 802.17 Resilient Packet Rings . . . . .	13
2.2.1 Service Classes . . . . .	15
2.2.2 RPR Node Architecture . . . . .	16
2.2.3 Fairness Control in RPR . . . . .	18
2.2.4 RPR Standard Bandwidth Allocation Algorithms . . . . .	24

<b>3</b>	<b>A New Fairness Reference Model for Resilient Packet Rings</b>	<b>29</b>
3.1	Preliminaries . . . . .	31
3.1.1	Max-Min Fairness . . . . .	32
3.1.2	Proportional Fairness . . . . .	34
3.1.3	Ring Ingress-Aggregated with Spatial Reuse (RIAS) Fairness . . . . .	35
3.2	Principles of Bandwidth Allocation and Fairness in RPR . . . . .	38
3.2.1	Bandwidth Allocation at Ring Level . . . . .	39
3.2.2	Bandwidth Allocation at Source Level . . . . .	39
3.3	Ring Ingress-Aggregated Max-Min Fairness . . . . .	41
3.3.1	Single-Destination Scenarios . . . . .	42
3.3.2	Multiple-Destination Scenarios . . . . .	49
3.4	RPR Source Behaviors . . . . .	54
3.4.1	Simple Single-Rate (SSR) . . . . .	55
3.4.2	Equal Partitioning (EP) . . . . .	56
3.4.3	Max-Min Partitioning (MMP) . . . . .	58
3.4.4	Performance Comparison of SSR, EP, and MMP Source Behaviors . . . . .	61
3.5	Comparison of RIAMM and RIAS Fairness Reference Models . . . . .	68
3.5.1	Equal Opportunity Source Behavior: A Case Study on Comparing RIAS and RIAMM Fairness Models . . . . .	71
3.6	Conclusion . . . . .	73
<b>4</b>	<b>Analysis of RPR Standard Bandwidth Allocation Algorithms</b>	<b>74</b>
4.1	Analysis of RPR-AM Fair Rate Calculations in Parking-lot Scenario . . . . .	77
4.1.1	Problem Description . . . . .	77
4.1.2	Analysis of Model without Link Propagation Delay . . . . .	79
4.1.3	Analysis of Model with Link Propagation Delay . . . . .	84
4.1.4	Numerical Results and Discussions . . . . .	90
4.2	Analysis of RPR-CM Fair Rate Calculations in Parking-lot Scenario . . . . .	98
4.2.1	Problem Description . . . . .	98
4.2.2	Analysis of Model without Link Propagation Delay . . . . .	98
4.2.3	Analysis of Model with Link Propagation Delay . . . . .	103
4.2.4	Numerical Results and Discussions . . . . .	108
4.3	Discussion on Generalization of Analyses . . . . .	115
4.4	Conclusion . . . . .	116
<b>5</b>	<b>Virtual Queuing: An Efficient Bandwidth Allocation Algorithm for Resilient Packet Rings</b>	<b>118</b>
5.1	DVSR Algorithm . . . . .	119

5.1.1	Performance Evaluation of DVSR . . . . .	121
5.2	Virtual Queuing Algorithm . . . . .	127
5.2.1	Model of Rate Control System in VQ . . . . .	127
5.2.2	VQ Algorithm Description . . . . .	131
5.2.3	Control Message Transmission . . . . .	135
5.2.4	Complexity of VQ Algorithm . . . . .	136
5.2.5	Drawbacks of VQ Algorithm . . . . .	137
5.2.6	Characterizing Fairness Properties of VQ Algorithm . . . . .	138
5.3	Performance Evaluation of VQ Algorithm . . . . .	143
5.3.1	Spatial Reuse and Steady-State Performance . . . . .	143
5.3.2	Comparison with RPR-AM and RPR-CM . . . . .	146
5.3.3	Comparison with DVSR . . . . .	152
5.3.4	Starvation Time and ATT Upper bounds in VQ . . . . .	162
5.4	Conclusion . . . . .	165
<b>6</b>	<b>Concluding Remarks</b>	<b>167</b>
6.1	Summary of Contributions . . . . .	167
6.2	Future Work . . . . .	169
<b>A</b>	<b>Stability of Developed Analytical Models for Resilient Packet Ring Aggressive Mode</b>	<b>171</b>
A.1	Stability Analysis of Model without Link Propagation Delay . . . . .	172
A.2	Stability Analysis of Model with Link Propagation Delay . . . . .	173
<b>B</b>	<b>Upperbounds for Characterizing Fairness Properties of VQ Algorithm</b>	<b>178</b>
B.1	Proof of Proposition 5.2.3 . . . . .	178
B.2	Proof of Proposition 5.2.4 . . . . .	180
	<b>Bibliography</b>	<b>182</b>

# List of Tables

2.1	Service Classes in RPR . . . . .	16
5.1	DVSR fair rate calculation . . . . .	120

# List of Figures

2.1	The architecture of a typical node on a buffer insertion ring . . . . .	10
2.2	A typical RPR node architecture . . . . .	17
2.3	Illustration of fairness objectives in RPR . . . . .	20
2.4	Control (fairness) message format . . . . .	21
2.5	Fair rate advertisement in RPR in a scenario with a single congested node . . . . .	21
2.6	Fair rate advertisement in RPR in a scenario with multiple congested nodes . . . . .	22
3.1	Illustration of max-min fairness . . . . .	32
3.2	An example to illustrate the single resource max-min fairness . . . . .	34
3.3	Illustration of proportional fairness . . . . .	35
3.4	Illustration of RIAS fairness . . . . .	38
3.5	Illustration of ring-level and source-level bandwidth allocation in RPR	41
3.6	A single-destination scenario . . . . .	43
3.7	A multiple-destination scenario . . . . .	53
3.8	An example for the SSR source behavior . . . . .	56
3.9	An example for the EP source behavior . . . . .	58
3.10	Parallel parking-lot scenario for $K = 4$ and $H = 3$ . . . . .	62
3.11	Throughput loss versus $H$ for the SSR and EP source behaviors in a parallel parking-lot scenario with $K = 10$ . . . . .	62

3.12	Throughput loss versus $H$ for the SSR and EP source behaviors in a parallel parking-lot scenario with $K = 7$ . . . . .	63
3.13	Throughput loss versus $K$ for the SSR and EP source behaviors in a parallel parking-lot scenario with $H = 2$ . . . . .	64
3.14	Mixed parking-lot scenario for $K = 4$ and $H = 2$ . . . . .	65
3.15	Throughput loss versus $K$ for the SSR and EP source behaviors in a mixed parking-lot scenario with $H = 1$ . . . . .	65
3.16	Throughput loss versus $K$ for the SSR and EP source behaviors in a mixed parking-lot scenario with $H = 2$ . . . . .	66
3.17	Two-exit parking-lot scenario for $K = 3$ . . . . .	67
3.18	The relative throughput difference between node 1 and other nodes versus $d$ in a two-exit parking-lot scenario for $K = 3, 4, 5$ . . . . .	68
3.19	An example to illustrate the EO source behavior . . . . .	72
4.1	Parking-lot scenario . . . . .	76
4.2	A low-pass filter model for fair rate calculations . . . . .	83
4.3	Effect of propagation delay on advertising the fair rate in RPR-AM . . . . .	85
4.4	Normalized calculated fair rate for $n = 8$ , $\tau_\ell = 0.95$ , and $d = 0.1 \cdot T$ . . . . .	91
4.5	Normalized calculated fair rate for $n = 4$ , $\tau_\ell = 0.95$ , and $\alpha = 0.5$ . . . . .	92
4.6	Variations of $\alpha_n^*$ versus the normalized propagation delay . . . . .	94
4.7	Convergence time for various values of $n$ and $\alpha$ with $\varepsilon = 0.01$ and $d = 0.1 \cdot T$ . . . . .	95
4.8	Convergence time versus $\alpha$ for $n = 6$ and various values of $d$ with $\varepsilon = 0.01$ . . . . .	97
4.9	Effect of the propagation delay on advertising the fair rate in RPR-CM . . . . .	103
4.10	An example to illustrate dynamics of RPR-CM . . . . .	109
4.11	Effect of $\beta$ on convergence of fair rate . . . . .	110
4.12	Effect of $d$ on convergence of fair rate . . . . .	112

4.13	Effect of $\tau_h - \tau_\ell$ on convergence of fair rate . . . . .	113
4.14	Comparison between the RPR-AM and the RPR-CM . . . . .	114
5.1	A two-node parking-lot scenario . . . . .	121
5.2	Throughput of the nodes in an unbalanced static scenario in RPR-AM	123
5.3	Throughput of the nodes in an unbalanced static scenario in RPR-CM	123
5.4	Throughput of the nodes in an unbalanced static scenario in DVSR .	123
5.5	Performance of RPR-AM in a dynamic traffic scenario . . . . .	126
5.6	Performance of RPR-CM in a dynamic traffic scenario . . . . .	126
5.7	Performance of DVSR in a dynamic traffic scenario . . . . .	126
5.8	A virtual queue model for rate control mechanism in RPR . . . . .	129
5.9	Parallel parking-lot scenario . . . . .	144
5.10	Upstream parking-lot scenario . . . . .	145
5.11	Two-exit parking-lot scenario . . . . .	146
5.12	Convergence times of the add-rate of the head node with RPR-AM .	147
5.13	Convergence times of the add-rate of the head node with RPR-CM .	147
5.14	Convergence times of the add-rate of the head node with VQ . . . . .	147
5.15	Convergence times of RPR-AM, RPR-CM, and VQ . . . . .	148
5.16	Throughput loss versus $\frac{T_L}{T_H}$ with Poisson traffic . . . . .	151
5.17	Throughput loss versus $\delta_L$ for $T_L = 2 \cdot T_H$ with Poisson traffic . . . . .	151
5.18	Throughput loss versus $\delta_L$ for $T_L = 10 \cdot T_H$ with Poisson traffic . . . . .	151
5.19	Throughput loss versus $\delta_L$ for $T_L = 2 \cdot T_H$ with Pareto traffic . . . . .	152
5.20	Throughput loss versus $\delta_L$ for $T_L = 10 \cdot T_H$ with Pareto traffic . . . . .	152
5.21	Throughput with DVSR (node 4 reclaims unused bandwidth) . . . . .	154
5.22	Throughput with VQ (node 4 reclaims unused bandwidth) . . . . .	154
5.23	ATT in DVSR and VQ (node 4 reclaims unused bandwidth) . . . . .	154
5.24	Throughput with DVSR (node 4 does not reclaim unused bandwidth)	156

5.25	Throughput with VQ (node 4 does not reclaim unused bandwidth)	156
5.26	ATT in DVSR and VQ (node 4 does not reclaim unused bandwidth)	156
5.27	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $\delta_L$	159
5.28	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $T_H + T_L$	159
5.29	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $N$ for $\delta_L = 0.01 \cdot C$ with Poisson traffic	159
5.30	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $N$ for $\delta_L = 0.05 \cdot C$ with Poisson traffic	160
5.31	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $N$ for $\delta_L = 0.01 \cdot C$ with Pareto traffic	161
5.32	Comparison of $\langle ATT \rangle$ in DVSR and VQ for various values of $N$ for $\delta_L = 0.05 \cdot C$ with Pareto traffic	162
5.33	Comparing upper bounds on starvation time with simulation results	164
5.34	Comparing upper bounds on ATT with simulation results	165
A.1	Locations of $H_d(z)$ poles in the $z$ -plane due to variations of $\alpha$ for $n = 8$ and $d = 0.05 \cdot T$ .	177
A.2	Locations of $H_d(z)$ poles in the $z$ -plane due to variations of $\alpha$ for $n = 8$ and $d = 0.1 \cdot T$ .	177

# List of Notations

Notation	Description	First Use
C	Unreserved rate	24
T	Length of a control interval	24
$\alpha$	Low-pass filtering coefficient	24
<i>lpCoeef</i>	Low-pass filtering coefficient	24
$r_i^a(k)$	Add-rate of node $i$ in control interval $k$	24
$\tilde{r}_i^a(k)$	Low-pass filtered version of the add-rate of node $i$ in control interval $k$	24
$r_i^f(k)$	Forward-rate of node $i$ in control interval $k$	25
$r_i^u(k)$	Rate of unreserved traffic at node $i$ in control interval $k$	25
$\tilde{r}_i^u(k)$	Low-pass filtered version of the rate of unreserved traffic at node $i$ in control interval $k$	25
$F_i(k)$	Fair rate of node $i$ calculated at the end of control interval $k$	25
$\beta$	Ramping coefficient	25
<i>rampUpCoeef</i>	Ramp-up coefficient	25
<i>rampDnCoeef</i>	Ramp-down coefficient	25
$\tau_l$	Low rate threshold	25
$\tau_h$	High rate threshold	25
$\ell_i(k)$	<i>allowedRate</i> value of node $i$ in control interval $k$	25

Notations	Description	First Use
$\ell_i^c(k)$	<i>allowedRateCongested</i> value of node $i$ in control interval $k$	25
$\Delta_i(k)$	Number of active nodes at node $i$ in control interval $k$	25
$l(i, i + 1)$	link between node $i$ and node $i + 1$	31
$f(i, j)$	Flow between node $i$ and node $j$	31
$r_{ij}$	Instantaneous rate of flow $f(i, j)$	31
$d_{ij}$	Demand of flow $f(i, j)$	31
$\mathbf{R}$	Set of allocated rates	32
$IA(i, n)$	An IA-flow at node $n$ originated from node $i$	36
$r_i^n$	Instantaneous rate of IA-flow $IA(i, n)$	36
$F_n$	Fair rate of node $n$	44
$\mathbf{R}_n$	Set of allocated rates of flows transiting node $n$	44
$\Phi(C, \mathbf{X})$	Max-min fair rate of flows with the set of demands $\mathbf{X} = \{x_i\}$ served at a server with capacity $C$	44
$ \cdot $	The number of elements in a set	45
$\eta$	A real number in $(0, 1]$	48
$\Phi^+(\cdot, \cdot, \eta)$	Adjusted max-min fair rate with parameter $\eta$	48
$(x)^+$	Maximum of $x$ and 0	48
$\check{\mathbf{R}}$	Set of allocated rates of IA-flows at all nodes	49
$\check{\mathbf{R}}_n$	Set of allocated rates of all IA-flows at node $n$	49
$\Theta_{nj}$	Rate limiter value of flow $f(n, j)$	54
$\Lambda_i^n$	Number of flows within IA-flow $IA(i, n)$	57
$L_n$	Max-min fair rate for flows within an IA-flow transiting node $n$	59
$\rho$	Throughput	61
$\hat{k}$	Control interval index of congestion detection time	81

Notations	Description	First Use
$\lfloor \cdot \rfloor$	Floor Function	81
$H(z)$	Transfer function	84
$d$	One-hop round-trip propagation delay	85
$r_i^a(k, j)$	Add-rate of node $i$ in subinterval $j$ of control interval $k$	86
$\alpha_n^*(d)$	Supremum value of $\alpha$ for an $n$ -node parking-lot scenario as a function of $d$	93
$k_{c,n}$	Convergence time of fair rate in an $n$ -node parking-lot scenario	94
$\varepsilon$	A real number in $(0, 1]$	94
$r_{m,i}^a(k, j)$	Add-rate of node $m$ observed by node $i$ in subinterval $j$ of control interval $k$	106
$r_i^u(k, j)$	Usage rate of output link of node $i$ in subinterval $j$ of control interval $k$	106
$r_i^n(k)$	Rate of IA-flow $IA(i, n)$ in control interval $k$	119
$T_H$	Duration of high-state of a dynamic source	124
$T_L$	Duration of low-state of a dynamic source	124
$\delta_H$	Average rate of a dynamic source in high-state	124
$\delta_L$	Average rate of a dynamic source in low-state	124
$A^n(k)$	Average arrival rate to the virtual queue of node $n$ in control interval $k$	128
$Q_i(k)$	Amount of traffic from IA-flow $IA(i, n)$ entering the virtual queue of node $n$ in control interval $k$	131
$E_i(k)$	Amount of eligible traffic of IA-flow $IA(i, n)$ conforming to the fair rate of node $n$	131
<b>I</b>	Set of input-limited IA-flows at node $n$	132
<b>R</b>	Set of rate-limited IA-flows at node $n$	132

Notations	Description	First Use
$E^I(k)$	Total amount of traffic arrived to virtual queue of node $n$ from input-limited IA-flows	132
$E^R(k)$	Total amount of traffic arrived to virtual queue of node $n$ from rate-limited IA-flows	132
$f(k)$	Rate increase/decrease factor in control interval $k$	132
$\sigma(k)$	Amount of throttled traffic of head-node in control interval $k$	139
$ATT(k)$	Accumulated amount of throttled traffic of head-node over $k$ control intervals	139
$\langle ATT \rangle (k)$	Average rate of throttled traffic of head-node over $k$ control intervals	139
$S$	Starvation time of head-node	139
$D$	Total demand of head-node	140
$m_1$	A real number in $[1, \infty)$	142
$m_2$	A real number in $(m_1, \infty)$	142
$U_S(\cdot, \cdot, \cdot)$	An upper bound function for starvation time	142
$\lceil \cdot \rceil$	Ceiling function	142
$U_{ATT}(\cdot, \cdot, \cdot)$	An upper bound function for ATT	142
$\bar{r}_{ij}$	Average allocated rate of flow $f(i, j)$ obtained from simulation results	143
$FI$	Fairness Index	143
$\mathbf{a}$	Shape parameter of Pareto distribution	144
$\mathbf{b}$	Scale (location) parameter of Pareto distribution	144
$\phi(x)$	A nonlinear function of $x$	171
$g(\cdot)$	A real function	174

# List of Abbreviations

Abbreviation	Description	First Use
ATT	Accumulated Throttled Traffic	139
BIR	Buffer Insertion Ring	9
CIR	Committed Information Rate	16
CQMA	Cyclic Queue Multiple Access	12
CRMA	Cyclic Reservation Multiple Access	12
DVSR	Distributed Virtual-time Scheduling in Rings	119
EIR	Excess Information Rate	16
EO	Equal Opportunity	71
EP	Equal Partitioning	56
FDDI	Fiber Distributed Data Interface	9
FE	Fairness Eligible	16
FRTT	Fairness Round Trip Time	27
GPS	Generalized Processor Sharing	33
HOL	Head of Line Blocking	17
IA	Ingress Aggregated	36
MAC	Medium Access Control	2
MMP	Max-Min Partitioning	58
PTQ	Primary Transit Queue	17
RIAMM	Ring Ingress-Aggregated Max-Min	41
RIAS	Ring Ingress-Aggregated with Spatial-reuse	35

<b>Abbreviation</b>	<b>Description</b>	<b>First Use</b>
RPR	Resilient Packet Ring	2
RPR-AM	RPR Aggressive Mode	24
RPR-CM	RPR Conservative Mode	24
SONET	Synchronous Optical Network	1
SSR	Simple Single-Rate	55
STP	Spanning Tree Protocol	2
STQ	Secondary Transit Queue	17
TCP	Transfer Control Protocol	34
VDQ	Virtual Destination Queuing	17
VQ	Virtual Queuing	119

# Chapter 1

## Introduction

Ring is the prevailing topology for deploying high-speed metro-area networks. One of the primary reasons is the inherent resiliency and fault tolerance advantage of the ring networks. That is, by employing bidirectional connections in a ring, all nodes can remain connected even in the presence of a link failure. Another reason is the deployment cost of rings, which is much less than complex mesh or star networks as a node is only connected to its neighboring nodes in a ring.

The current alternative technologies for high-speed metro-ring networks such as SONET (Synchronous Optical Network) and Ethernet do not exhibit satisfactory performance characteristics. A SONET ring consists of two ringlets which support a fast recovery protocol in case of a link or a node failure [1]. However, one of the ringlets is considered as the backup and is not used during normal operation of the network. Moreover, SONET was originally designed for circuit-switched applications and provisions point-to-point circuits between ring nodes. This guarantees a minimum bandwidth between any pair of nodes, however, each circuit is allocated a fixed amount of bandwidth and cannot be reclaimed by other nodes if not used. This static bandwidth allocation results in low utilization in SONET rings.

Ethernet is another alternative technology for deploying metro-ring networks which provides a simple and low-cost solution for data transport in a network [2]. However,

it relies on the Spanning Tree Protocol (STP) [3], which requires one of the links on the ring being disabled to eliminate all loops from the network. Therefore, all traffic should be forwarded in the same direction and not along the true shortest paths to the destinations. In other words, Ethernet does not take advantage of the ring topology and, as a result, can not provide high bandwidth utilization in ring networks. Further, it cannot guarantee a minimum share of bandwidth for any node.

In early 2000, the IEEE 802.17 Resilient Packet Ring (RPR) group was formed to develop a new Medium Access Control (MAC) standard for metro-ring networks. The RPR standard is devised to offer key features of SONET such as fast fault recovery and efficient support for ring topology as well as simplicity and cost advantages that are typical to Ethernet. Furthermore, RPR aims at providing high throughput and fair ring access for all nodes [4, 5].

RPR supports *spatial reuse* which is multiple concurrent transmissions over different parts of the ring. In other words, multiple nodes are allowed to send traffic simultaneously as long as bandwidth is available. Although spatial reuse improves the achieved throughput of the ring, it may cause congestion and unfairness among nodes in accessing the ring. Therefore, a fairness control mechanism is required in RPR in order to provide a fair ring access to all nodes. To maintain fairness in accessing the ring among nodes, a bandwidth allocation algorithm is employed at each RPR node. The bandwidth allocation algorithm in RPR operates based on the concept of rate feedback control in which an explicit rate is calculated and advertised to all nodes competing for the ring bandwidth. In addition to maintaining fairness among nodes, the bandwidth allocation algorithm in RPR aims at providing high bandwidth utilization on the ring. Thus, it should exhibit a fast response time to instantaneous changes in traffic load on the ring to ensure high utilization of the ring bandwidth. Furthermore, the complexity of implementing the bandwidth allocation algorithm should be relatively low as it is supposed to operate in a high-speed network.

## 1.1 Motivations

Bandwidth allocation and fairness are two major issues in resilient packet rings. In order to examine fairness of a set of allocated rates in RPR, an exact definition of fairness is required. On the other hand, to achieve fairness while providing high bandwidth utilization, efficient bandwidth allocation algorithms should be employed in RPR. The importance of defining fairness and designing efficient bandwidth allocation algorithms in RPR are the main motivations behind this research.

In the first part of the research (Chapter 3), we focus on fairness definition in RPR. In the RPR standard, fairness is defined as providing equal opportunity for competing nodes on any given link throughout the ring. However, this definition does not provide a detailed guideline in order to determine fairness of a given solution. We referred to a set of criteria under which fairness of a set of allocated rates can be examined as fairness reference model. In this thesis, we show that the existing fairness reference model proposed for RPR is not a general fairness reference model. Then, we develop a general fairness reference model for RPR.

In the second part of the research (Chapter 4 and Chapter 5), we focus on different aspects of bandwidth allocation in RPR.

The RPR standard defines two bandwidth allocation algorithms used to maintain fairness among nodes. To evaluate the fairness characteristics of these algorithms, it is necessary to study the effect of different system parameters on performance of these algorithms. As RPR is a complex dynamic system, the performance of the RPR standard bandwidth allocation algorithms has been investigated in literature only through simulation results [4–9]. However, an exact analysis of the RPR standard bandwidth allocation algorithms can provide an insight on dynamics of these algorithms. Moreover, the impact of different parameters on fairness characteristics of these algorithm can be studied through analytical results. In this thesis, we model

the fair calculation process in the RPR standard bandwidth allocation algorithms and study the performance of these algorithms through the developed models (Chapter 4).

Performance evaluations of the RPR standard bandwidth allocation algorithms show that these algorithms suffer from a number of performance limitations such as slow response times and oscillations. Although a number of modifications have been proposed to improve the performance of these algorithms, most of the modified algorithms still inherit some performance limitations of the RPR standard bandwidth allocation algorithms. A new proposal in [6] outperforms the RPR standard bandwidth allocation algorithms in terms of response time and bandwidth utilization at the expense of a complex implementation. However, it still suffers from some performance limitations in dynamic traffic scenarios. In this thesis, we also devise a new bandwidth allocation algorithm for RPR which outperforms the RPR standard bandwidth allocation algorithms and the one in [6] in terms of fairness characteristics specially in dynamic traffic environments (Chapter 5).

## 1.2 Contributions

The contributions of this research are summarized as follows:

- Defining the notion of fairness in RPR mathematically and developing a general fairness reference model for RPR (Chapter 3)
- Evaluating the performance of different policies employed for distributing bandwidth share of each RPR node among its local flows (Chapter 3)
- Modeling the fair rate calculation process in the RPR standard bandwidth allocation algorithms in the parking-lot scenario (Chapter 4)
- Studying the effect of different system parameters on the performance of the

RPR standard bandwidth allocation algorithms through the developed analytical models (Chapter 4)

- Stability analysis of the RPR standard bandwidth allocation algorithms through developed analytical models (Appendix A)
- Evaluating the performance of different RPR bandwidth allocation algorithms in various traffic scenarios to highlight their performance limitations (Chapter 5)
- Proposing a new bandwidth allocation algorithm for RPR with better performance characteristics compared to the existing algorithms (Chapter 5)
- Characterizing fairness properties of the proposed algorithm (Chapter 5, Appendix B)
- Evaluating the performance of the proposed algorithm extensively through simulation results in various traffic scenarios (Chapter 5)

Parts of the results of this research have been presented in a number of conferences or submitted for possible publication in journal [10–18].

### 1.3 Organization of Thesis

In **Chapter 2**, we provide an overview of medium access methods in ring networks. We review the access schemes with and without spatial reuse in ring networks. We also review the basic concepts of IEEE 802.17 resilient packet ring standard which will be referred to in the rest of this dissertation.

The contributions of this research are presented in **Chapters 3, 4, and 5**. Chapter 3 is concerned with the definition of fairness in RPR while **Chapter 4** and **Chapter 5** deal with bandwidth allocation algorithms in resilient packet rings.

**Chapter 3** presents a new fairness reference model for resilient packet ring. In this chapter, we first briefly review some of the well-known fairness reference models in the context of ring networks. We then develop a new fairness reference model based on the RPR design goals. We also compare the developed fairness reference model with the existing RPR fairness reference model. We show that our developed model is a generalized form of the existing RPR fairness reference model. Some of the results from this part of research were presented in [10, 11].

In **Chapter 4**, we model the fair rate calculation process in the RPR standard bandwidth allocation algorithms in parking-lot scenario. We then study the effect of various system parameters on the performance of these algorithms using the developed models. These results were presented in [13–17] and accepted for publication in [12].

In **Chapter 5**, we propose a new bandwidth allocation algorithm for resilient packet rings. We first study the performance of some well-known RPR bandwidth allocation algorithms including those recommended by the RPR standard. We show that these algorithms suffer from a number of performance limitations. We then design a new bandwidth allocation algorithm for RPR which outperforms those studied earlier. We also derive some bounds to characterize fairness properties of the proposed algorithm. Some of the results from this part of research were presented in [18].

**Chapter 6** briefly summarizes the contributions of this proposal and suggests some directions for future research work.

# Chapter 2

## Preliminaries and Background

Medium access control protocols coordinate the access to a shared medium in a communication network [19]. In packet ring networks, in particular, it is also essential to provide a fair access to the ring bandwidth for all nodes. Achieving fairness among nodes in traditional ring networks, which do not support spatial reuse, is not a complex task. These networks do not allow concurrent access to the ring and, consequently, there is no contention amongst nodes when accessing the ring. Therefore, in such ring networks, fairness can be maintained by limiting the ring access time for each node and/or controlling the amount of traffic each node is allowed to add to the ring. On the other hand, ring networks with spatial reuses allow multiple nodes to access simultaneously the ring, as a result, they require more complex protocols to ensure a fair access to the ring for all nodes.

In this chapter, we first review some existing MAC schemes for packet rings with and without spatial reuse. We then review basic concepts of fairness control in IEEE 802.17 RPR standard, which will be used in the rest of this dissertation.

## 2.1 Overview of Medium Access Methods for Ring Networks

In this section, we briefly describe some well-known medium access methods for ring networks. In general, according to the employed access schemes, ring networks can be classified into two different categories: ring networks which do not support spatial reuse; and those which allow simultaneous transmission of different users and, hence, support spatial reuse.

### 2.1.1 Ring Networks without Spatial Reuse

In ring networks without spatial reuse, only one node is allowed to transmit traffic at a time. These ring networks, which are now regarded as *token rings*, were first introduced in 1969 [20]. In a token-ring network, a single control message called *token* circulates along the ring. Upon receiving the token, each node is allowed to send traffic for a limited period of time and, then, it has to pass the token to its neighboring node. If a node does not have any traffic to transmit, it immediately forwards the token to the next node. As the time each node can hold the token is limited in token rings, a fair access to the ring bandwidth is easily ensured among all nodes. However, the achieved throughput in a token-ring network is relatively low and it is limited to the bandwidth of a single link. In the following, we briefly review two well-known token-ring standards.

#### IEEE 802.5 Token Ring

The IEEE 802.5 Token Ring is a MAC protocol for ring networks standardized in 1989. It was originally designed to operate at 4Mbps or 16Mbps modes although other variations for high-speed token rings were developed in the late 90's. The

frames are removed from the ring by the sender after circulating around the ring. A node is allowed to pass the token to its neighbor only when it removes its frames from the ring. The token holding time, which is the maximum time each node is allowed to possess the token, is 10msec in this standard. In a token ring, nodes can be given priority to possess the token and transmit data. Furthermore, token-ring networks support a token reservation scheme which enables nodes to reserve the token at a specific priority level. In a token-ring network, one of the nodes called *active monitor* oversees the ring to ensure that it functions properly [21–24].

### **Fiber Distributed Data Interface (FDDI)**

FDDI is a 100Mbps fiber optic token-ring network proposed as ANSI X3.139 standard in 1987. It can be employed to connect workstations within a network as well as to serve as a high-speed backbone to connect other networks over distances up to 200km. FDDI consists of two counter-rotating fiber ringlets where one of the ringlets is used as a backup. Unlike IEEE 802.5, each node in FDDI passes the token to its neighbor immediately after it finishes its data transmissions. This improves the performance of the network in terms of access delay and achieved throughput when compared to IEEE 802.5 standard [22, 25–28].

#### **2.1.2 Ring Networks with Spatial Reuse**

Ring networks with spatial reuse allow multiple transmissions over different ring segments and, hence, achieve significantly higher throughput compared to token rings. Two common approaches for achieving spatial reuses are *slotted rings* and *buffer insertion rings* (BIR). Both schemes support spatial reuse by removing packets at destinations and allowing multiple simultaneous transmissions by different nodes. In

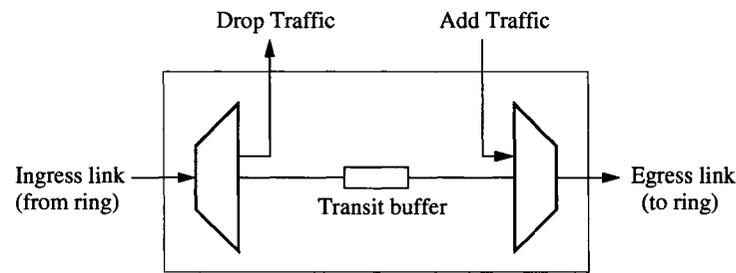


Figure 2.1: The architecture of a typical node on a buffer insertion ring

slotted rings, which were originally introduced in 1972, a fixed number of slots constantly circulate on the ring [29]. A node can use a slot to transmit a message, if that slot is either empty or contains a message destined to that node. As a result, different nodes are allowed to transmit data at the same time on different segments of the ring. The main disadvantage of a slotted ring is that all messages should have equal sizes in order to be transmitted in a slot.

On the other hand, buffer insertion rings support variable size packets. In this method, which was introduced in 1974 in [30], each node (station) has an insertion (transit) buffer which can accommodate a few maximum size packets (see Figure 2.1). At each node, traffic from the ring is removed if destined to that node; otherwise, it is temporarily stored in the transit buffer. Each node may start adding its own local traffic (add traffic) to the ring only when its transit buffer is empty. If the ring traffic transiting a node (transit traffic) arrives while the node is in the middle of a packet transmission, the ring traffic is temporarily stored in the transit buffer. Therefore, the transit buffer should be able to accommodate at least one maximum size packet to ensure that packets are not dropped on the ring before reaching their destinations. The performance of buffer insertion rings has been widely studied in literature such as [31–35].

Note that in slotted and buffer insertion rings, spatial reuse results in giving priority to transit traffic over add traffic at each node. Clearly, the add traffic of a

node can be blocked as a result of heavy transit traffic. This situation in which a node cannot add its traffic to the ring is known as *starvation* or *congestion*. Therefore, a medium access control protocol must be employed in slotted and buffer insertion rings to ensure a fair allocation of the ring bandwidth to all nodes. Such access schemes define the manner in which nodes share the ring and control the amount of traffic each node is allowed to add to the ring in order to prevent starvation. In the following, we review a number of well-known medium access protocols proposed which control fairness in rings with spatial reuse.

### **MetaRing**

The MetaRing protocol is a quota-based allocation scheme for a bidirectional slotted or buffer insertion ring which was introduced in 1993 [36]. The MetaRing protocol supports spatial reuse by employing a control message called SAT (satisfied). The SAT message circulates around the ring on the opposite direction of data flow. In MetaRing, the goal is to provide equal opportunities for all nodes during a SAT rotation time around the ring. That is, every node should be able to send the same number of packets in each SAT rotation time. This algorithm has two parameters:  $k$  and  $l$ ,  $k \leq l$ . Each node is allowed to send up to  $k$  packets since the last receipt of the SAT message. After transmitting  $k$  packets, a node is not allowed to send any more packets until it receives the SAT message. Upon receipt of the SAT message, if a node has sent at least  $l$  packets, it immediately forwards the SAT message to its neighbor. However, if a node has not transmitted at least  $l$  packets (while it has traffic to send), it holds the SAT message until  $l$  packets are transmitted. Although the performance of the MetaRing protocol in terms of the achieved throughput is significantly better than token rings, it has a number of shortcomings. For example, the ring access delay of each node increases with  $l$  and  $k$ . Further, its performance depends on the traffic pattern of nodes as well as control parameters  $l$  and  $k$  [37]. Two mechanisms were

proposed in [38] to improve the performance of MetaRing protocol in terms of the access delay and the achieved throughput.

### **Cyclic Reservation Multiple Access II (CRMA-II)**

CRMA-II is an access protocol devised for ring networks [39]. It is a generalized version of CRMA protocol which was originally introduced as an access scheme for bus networks [40]. The CRMA-II protocol is developed for a slotted ring with insertion buffer at each node. It supports spatial reuse by removing frames at their destinations. The fairness control in CRMA-II is based on a cyclic reservation scheme. The empty slots throughout the ring are marked as *free* or *reserved*. A reserved slot can only be used by nodes which have confirmation for a reserved slot within the current reservation cycle. At the end of each reservation cycle, each node sends a request to a certain node called *scheduler*. Based on the requests from all nodes and the number of slots previously used by each node, the scheduler calculates a threshold which is advertised to all nodes. The scheduler also marks a number of slots as reserved according to the threshold value. Upon receiving the threshold, each node is allowed to use a number of slots up to the threshold value to send its frames.

One of the drawbacks of this protocol is its reservation process and threshold calculations which are performed globally at the scheduler. In fact, the threshold calculation can be very complex in a large network. It is also shown in [41] that the CRMA-II protocol results in a high access delay.

### **Cyclic Queue Multiple Access (CQMA)**

CQMA was proposed as an alternative medium access control protocol for RPR [42]. It is a rate allocating protocol which calculates the rate of each individual source-destination flow throughout the ring according to demands. Therefore, traffic of each node should be buffered in a per-destination basis.

The CQMA protocol operates in two modes. In the first mode, a single control message circulates around the ring which contains the current demands of all source-destination flow. Assume that  $N$  is the number of nodes on the ring and  $P$  is the number of supported service classes. In this case, the control message should be able to carry a table with  $(N - 1)^2 \cdot P$  entries for the demand information of all individual flows. When a node receives the control message, it updates the corresponding fields of the table with the latest demand information of all flows originating at that node and keeps a copy of the table information locally. The control message is then forwarded to the neighboring node. Based on the demand information of all nodes, each node calculates the allocated rate of all flows originating at that node. The rate allocation is performed at the end of every fixed time interval called *calculation interval*.

Clearly, as the size of table in the control message grows with  $N^2$ , this protocol cannot be employed when the number of nodes is large. Therefore, in the second mode of operation of this protocol (which is an approximation of the first mode), the size of table in the control message is reduced such that this protocol can be used for large networks. It is shown in [43] that the complexity of both modes of the CQMA protocol is of  $O(N^3)$ . As a result, this protocol is not a good candidate for rate allocation in a high-speed network such as RPR.

## 2.2 Overview of IEEE 802.17 Resilient Packet Rings

In the previous section, several MAC protocols for ring networks were introduced. However, none of them provides a high-performance and efficient (yet simple) solution for medium access control in high-speed metro-area ring networks. This was the primary motivation behind developing the IEEE 802.17 Resilient Packet Ring

standard as a MAC protocol for metro-ring networks.

The RPR standard is devised to achieve several objectives including high throughput, fault tolerance, fairness, and spatial reuse which are not simultaneously available in current protocols such as those introduced in the previous section [4, 5].

RPR is a buffer insertion ring technology which supports spatial reuse by removing packets from the ring at destination nodes. Similar to other MAC protocols, the RPR standard specifies how available bandwidth on a link should be allocated to contending nodes. Moreover, it defines how a node reacts to congestion and starvation to maintain fairness among all nodes.

An RPR network employs two unidirectional counter-rotating ringlets and can accommodate up to 256 nodes [44]. Unlike FDDI in which one of the ringlets is always idle during normal operations, both ringlets in RPR are operational and used for data transmission among nodes. Each node can send traffic to a destination using the ringlet which has the shortest path to that destination. The dual-ring architecture is also used to protect traffic against any link or node failure. When a node detects that a link or its adjacent neighbor has failed, it immediately wraps the ring at the breakpoint and sends the traffic back in the other direction using the opposite ringlet.

Each RPR node has an insertion (transit) buffer where the ring traffic transiting that node is temporarily stored. A node can add its local traffic (add-traffic) when its transit buffer is empty. This means that the ring traffic transiting a node can block local traffic of that node in accessing the ring bandwidth. In order to control fairness among nodes in utilizing the ring bandwidth, RPR employs a distributed rate allocation algorithm termed *bandwidth allocation algorithm* or *fairness algorithm*.<sup>1</sup> The bandwidth allocation algorithm specifies how each node should limit the insertion

---

<sup>1</sup>In this thesis, we always use the former term to avoid any confusion with the concept of fairness reference model addressed in Chapter 3.

rate of its local traffic (add-rate) to the ring to let other nodes located in *downstream*<sup>2</sup> direction inject traffic to the ring as well. When a node detects congestion, it calculates a fair rate which is advertised to its upstream nodes through a control message. The control message is transmitted in the opposite direction of traffic flow using the opposite ringlet. Upon receiving the control message, upstream nodes limit the add-rate of their local traffic transiting the congested node to the advertised fair rate. As upstream nodes reduce their add-rates to the fair rate, the congested node will be able to add its traffic to the ring. When congestion clears at the congested node, upstream nodes are allowed to increase their rates gradually [4, 5, 44].

During normal operation of the ring in RPR, the fairness control on each ringlet is performed independent of the one on the other ringlet. That is, two separate instances of the bandwidth allocation algorithm are employed at each node where each one is in the charge of fairness control and bandwidth management operations corresponding to one of the ringlets [44].

In the following, we continue by reviewing other specifications of the RPR standard such as service classes, node architecture, and bandwidth allocation algorithm functionality.

### 2.2.1 Service Classes

The RPR MAC supports three service classes: high priority class A; medium priority class B; and low priority class C [44].

Class A provides an allocated, guaranteed data rate with low end-to-end delay and jitter bounds. This class is divided into two subclasses: A0 and A1. Subclass A0 has reserved bandwidth throughout the ring while bandwidth is allocated (not reserved) for subclass A1 which can be reclaimed if not used.

---

<sup>2</sup>Downstream is the direction of traffic flow on a ringlet. Similarly, the opposite direction of traffic on a ringlet is termed *upstream*.

Table 2.1: Service Classes in RPR

class	subclass	guaranteed bandwidth	delay/jitter	bandwidth type	bandwidth subtype	fairness eligible
A	A0	yes	low	allocated	reserved	no
	A1				reclaimable	
B	B-CIR	no	bounded	opportunistic		yes
	B-EIR		unbounded			
C	—					

Class B is also divided into two subclasses: B-CIR (Committed Information Rate) and B-EIR (Excess Information Rate). Subclass B-CIR has allocated bandwidth and provides guaranteed data rate with bounded end-to-end delay and jitter. Any class B traffic beyond allocated subclass B-CIR is considered as subclass B-EIR which does not have guaranteed or allocated bandwidth on the ring.

Class C provides a best-effort traffic service with no allocated or guaranteed data rate and with no bounded end-to-end delay or jitter.

It is necessary to mention that in RPR, traffic subclass B-EIR and class C are subject to the bandwidth allocation algorithm and called *fairness eligible* (FE) traffic. Table 2.1 summarizes the RPR service classes and their characteristics.

*Remark 2.2.1.* In this thesis, we only consider FE traffic as class A and subclass B-CIR have allocated bandwidth and are not considered for fairness determination in RPR.

## 2.2.2 RPR Node Architecture

Figure 2.2 presents an RPR node architecture, where only one direction of the ring traffic associated with one of the ringlets is shown. At each node, MAC client traffic (termed local traffic) is buffered in the local queues based on different priorities. In addition, local traffic destined to different nodes may be buffered in separate queues.

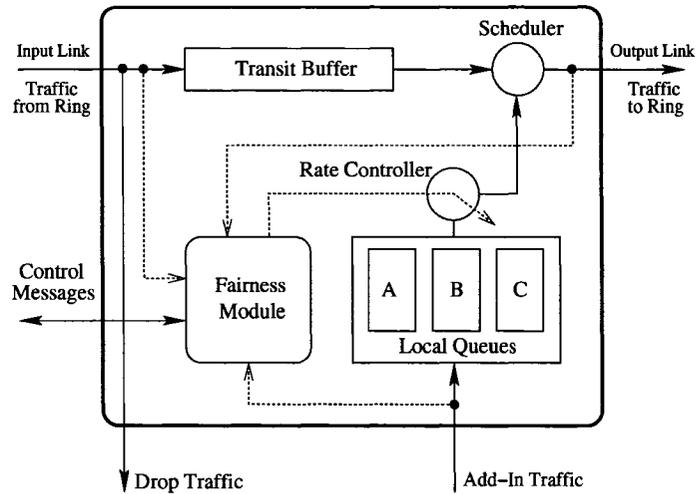


Figure 2.2: A typical RPR node architecture

This buffering method is termed virtual destination queuing (VDQ), which is a solution to the head-of-line (HOL) blocking problem [45]. At each node, arriving traffic from the ring is removed if destined to that node; otherwise, it is forwarded to the transit buffer. The transit buffer may be implemented in two modes: *single-queue* or *dual-queue*.

In the single-queue mode, transit traffic of all classes is forwarded into a single queue called primary transit queue (PTQ), which can only accommodate a few maximum size packets [44]. The scheduler gives service priority to transit traffic over local traffic in this mode. The advantage of the single-queue mode is that it simplifies the hardware implementation. However, the transit traffic can block the local traffic waiting in the local buffers to access the ring.

In the dual-queue mode, the transit buffer consists of two queues: primary transit queue (PTQ) and secondary transit queue (STQ). In this mode, class A transit traffic is buffered in PTQ and traffic of other service classes is forwarded into STQ. In the dual-queue mode, the scheduler always gives the utmost service priority to PTQ and class A local traffic. When there is no class A transit or local traffic and STQ length

is less than a high threshold, class B and class C local traffic have priority over STQ. If STQ reaches its high threshold, STQ has service priority over class B and class C local traffic. The advantage of the dual-queue transit buffer is that class A traffic will not usually experience significantly more than the propagation delay and some occasional transit delay on the ring. Hence, employing the dual-queue transit buffer improves the delay performance of class A traffic [46]. However, as STQ has the lowest service priority in this mode, its size should be large enough to prevent ring traffic being dropped. The complexity of implementing the dual-queue mode in terms of scheduling and required buffer size is more than the single-queue mode.

The rate of local traffic at each node is controlled by a set of rate controllers. The fairness module implements the bandwidth allocation algorithm and adjusts the rate controllers corresponding to the local FE traffic. In case of congestion, the fairness module sends a control (fairness) message containing a fair rate to the upstream nodes contributing to congestion. Control messages are transmitted on the opposite direction of the traffic flow using the opposite ringlet. Control messages use subclass A0 reserved bandwidth.

### **2.2.3 Fairness Control in RPR**

As mentioned earlier, RPR employs a distributed rate-based approach in order to control fairness around the ring. In this section, we describe some basics of fairness and congestion control in RPR. Details of fairness control in RPR can also be found in [44].

#### **Congestion Detection**

Congestion occurs when demand for bandwidth on a link exceeds the capacity of that link. The congestion detection conditions in an RPR node vary based on the

implementation of its transit path as follows.

- An RPR node deploying a single-queue transit buffer is congested when either of the following conditions are met:
  - Low-pass filtered version of the link usage (defined in Section 2.2.4) at the output of its scheduler exceeds a certain threshold.
  - Local traffic is delayed more than a predefined time threshold while awaiting to get access to the ring.
- A node employing a dual-queue transit buffer is congested when the occupancy of the STQ exceeds a certain threshold.

When congestion is triggered, bandwidth on the output link of the congested node should be fairly divided among all competing nodes in order to maintain fairness. In the following, we describe the notion of fairness and fairness objectives in RPR.

### **Fairness Objectives in RPR**

The bandwidth allocation algorithm in RPR regulates the addition of FE traffic to a ringlet to control congestion and to maintain fairness among all nodes. The objective is to achieve *source-based weighted fairness* among competing nodes (sources) on any segment of the ring. That is, bandwidth on any given link should be allocated to local traffic of each node in proportion to the relative weight assigned to that node. For instance, if all nodes have equal weights and all demands are equal, the available bandwidth on a given link should be distributed equally among all competing nodes on that link. In addition, if one node has a larger weight, the allocated bandwidth to that node should be in proportion to its weight divided by the sum of all weights. In fact, the main goal of maintaining source-based fairness in RPR is to provide a fair access to the ring for all nodes.

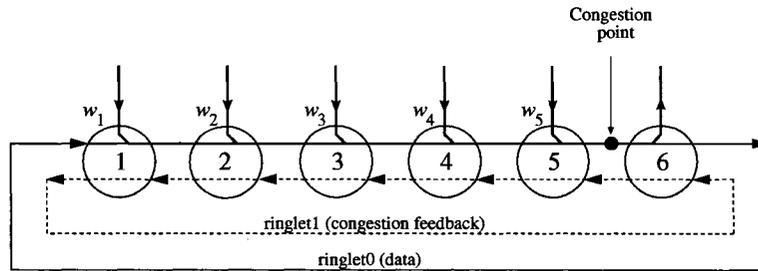


Figure 2.3: Illustration of fairness objectives in RPR

**Example 2.2.1.** Figure 2.3 presents an example in which nodes 1, 2, ..., and 5 send traffic destined to node 6. In this case, the output link of node 5 is the most congested point where all nodes are competing for bandwidth. Assume that all nodes have equal weights, i.e.  $w_1 = w_2 = \dots = w_5$ . Moreover, assume that the link capacity is normalized and all nodes can send traffic to node 6 with rate 1. In order to maintain fairness, each node should send traffic to node 6 with rate  $\frac{1}{5}$  as they have equal weights. Now assume that  $w_1 = w_2 = 3$  and  $w_3 = w_4 = w_5 = 1$ . In this case, the allocated rate to each node is proportional to its weight and, as a result, nodes 1 and 2 can add traffic to the ring with rate  $\frac{1}{3}$  and nodes 3, 4, and 5 should send traffic with rate  $\frac{1}{9}$ . ◀

### Fair Rate Advertisement

To achieve fairness objectives and control the rate of all nodes contributing to congestion in RPR, a congested node calculates a fair rate at the end of every fixed time interval called *control interval*. During each control interval, a congested node collect rate statistics which are used for fair rate calculations. (We will review how a congested node in RPR calculates its fair rate in Section 2.2.4). The fair rate is then advertised to all upstream nodes contributing to congestion at the end of every control interval. The fair rate is transmitted through a control (fairness) message on the opposite ringlet. An RPR control message is a 16-byte frame which includes several fields as presented in Figure 2.4. Each control message carries a 48-bit address

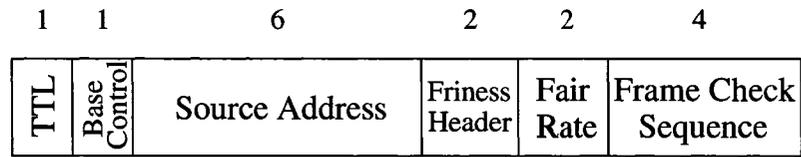


Figure 2.4: Control (fairness) message format

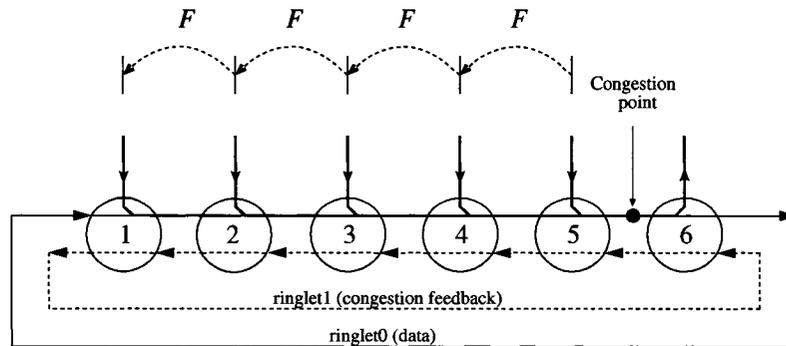


Figure 2.5: Fair rate advertisement in RPR in a scenario with a single congested node of the originating congested node along with the advertised fair rate of that node. (see [44] for detailed description of each field in the control message.)

The fair rate is propagated hop by hop in the upstream direction (opposite direction of traffic flow) until all contributing nodes receive the fair rate. Upon receiving the fair rate, upstream nodes limit the addition of FE traffic transiting the congested node to the advertised fair rate. After contributing nodes limit their rates, the congested node adjusts its fair rate based on the new measurements and advertises it to upstream nodes. This feedback between the congested node and the contributing nodes allows continuous adjustment of the rates to meet fairness objectives. Thus, the fair rate advertisement is essential in order to control congestion and maintain fairness in RPR. Figure 2.5 illustrates the fair rate advertisement on ringlet1 in order to control congestion on ringlet0. It can be seen that the fair rate  $F$ , which has been computed by congested node 5, is propagated back to the contributing upstream nodes.

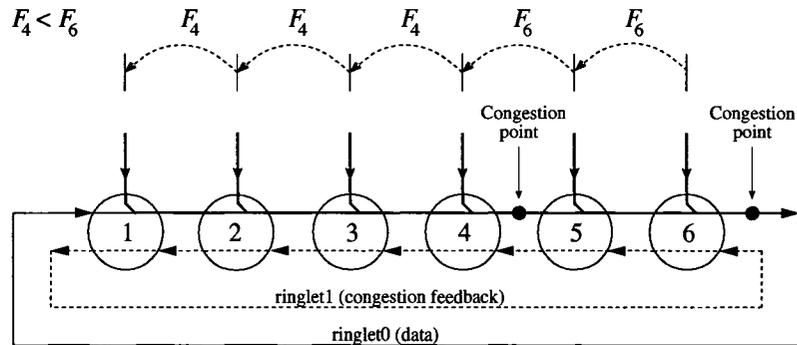


Figure 2.6: Fair rate advertisement in RPR in a scenario with multiple congested nodes

Note that when there are multiple congested nodes on the ring, each one calculates a local fair rate independently. When a congested node receives a fair rate from one of its downstream nodes, it advertises the minimum of its own local fair rate and the received fair rate from the downstream congested node. Figure 2.6 presents a scenario in which nodes 4 and 6 are congested and the fair rate of node 4 is less than the local fair rate of node 6, i.e.  $F_4 < F_6$ . It can be seen that node 4 does not forward the received fair rate of node 6 and instead advertises its own fair rate. This ensures that upstream nodes of node 4 do not send traffic more than the fair rates of nodes 4 and 6. On the other hand, if the fair rate of node 4 was greater than the fair rate of node 6 and the upstream nodes of node 4 were contributing to congestion at node 6, node 4 would forward the fair rate of node 6.

When a node receives a fair rate from a downstream congested node and determines that its upstream nodes are not contributing to congestion at that downstream congested node, it does not propagate the fair rate. Instead, it advertises full line rate as the fair rate to inform its upstream nodes that they are not contributing to congestion or their contribution is much less than the advertised fair rate of the congested node. It is necessary to mention that a control message (carrying a fair rate) always includes the identity of the origin whose fair rate is being advertised.

This means that each node only receives the fair rate information corresponding to the most congested downstream node which has the minimum fair rate. However, to achieve spatial reuse and utilizing unused capacity, each RPR node may report its calculated local fair rate by broadcasting a specific control message to all other nodes on the ring.

### **Congestion Domain**

A congestion domain (or congestion span) consists of a congested node and a set of nodes contributing to congestion. A node which is congested and advertises its local fair rate to its upstream nodes is known as the *head* of the congestion domain. A congestion domain starts with a head node and extends upstream and ends at a *tail* node. The tail node is the most upstream node which contributes to the congestion at the head node and regulates addition of its fairness eligible traffic according to the advertised fair rate of the head node. Therefore, the tail node does not propagate the advertised fair rate of the head node upstream as its upstream nodes do not contribute to congestion. In Figure 2.5, node 5 is the head node and node 1 is the tail node of the congestion domain.

When multiple congestion points exist, a tail node can be the head node for another congestion domain. It happens when the tail node in a congestion domain is also congested and its calculated local fair rate is less than the advertised fair rate of the head node. This can be seen in Figure 2.6 where node 4 is the tail node for the congestion domain in which node 6 is the head node. In addition, node 4 is the head node of another congestion domain which extends upstream and ends at node 1.

## 2.2.4 RPR Standard Bandwidth Allocation Algorithms

In the previous section, we described some basics of fairness control in RPR. It was mentioned that in order to maintain fairness among nodes, a congested node calculates and advertises a fair rate to its upstream nodes contributing to congestion. The fair rate is calculated by the bandwidth allocation algorithm (which is implemented at all nodes) based on some collected rate statistics. The RPR standard defines two bandwidth allocation algorithms: *Aggressive Mode* (RPR-AM) and *Conservative Mode* (RPR-CM). There are also a number of proposals for improving the performance of RPR standard bandwidth allocation algorithms [6, 8, 47–51]. In this section, we describe the fair rate calculations in RPR-AM and RPR-CM.

Congestion detection and fair rate calculations at each RPR nodes are performed at the end of every control interval. Each RPR node measures the service rate of its FE local traffic and that of the transit traffic, at the output of its scheduler (see Figure 2.2). The *add-rate* (respectively, the *forward-rate*) is the *average* amount of FE local traffic (respectively, transit traffic) serviced per unit of time; both are measured at the end of every control interval at each node. Let us define the following notations:

- ▶  $C$  is the unreserved link capacity called *unreserved* rate.
- ▶  $T$  is the length of a control interval in seconds and the  $k$ th ( $k > 0$ ) control interval is time interval  $((k - 1) \cdot T, k \cdot T]$ .
- ▶  $\alpha$  is a low-pass filtering coefficient ( $0 < \alpha < 1$ ). In the RPR standard,  $\alpha^{-1}$  is denoted as *lpCoef*.
- ▶  $r_i^a(k)$  is the add-rate of node  $i$  in control interval  $k$  and  $\tilde{r}_i^a(k)$  is its low-pass filtered version defined as follows:

$$\tilde{r}_i^a(k) = (1 - \alpha) \cdot \tilde{r}_i^a(k - 1) + \alpha \cdot r_i^a(k). \quad (2.2.1)$$

- ▶  $r_i^f(k)$  is the forward-rate of node  $i$  in control interval  $k$ .
- ▶  $r_i^u(k)$  is the average aggregate rate of the unreserved traffic (all traffic except traffic of subclass A0), added and/or forwarded by node  $i$  in control interval  $k$ . Its low-pass filtered version denoted by  $\tilde{r}_i^u(k)$  is given by

$$\tilde{r}_i^u(k) = (1 - \alpha) \cdot \tilde{r}_i^u(k - 1) + \alpha \cdot r_i^u(k). \quad (2.2.2)$$

- ▶  $F_i(k)$  is the fair rate of node  $i$  calculated at the end of control interval  $k$ .
- ▶  $\beta$  is a ramping coefficient ( $0 < \beta < 1$ ). In the RPR standard, there are two coefficients for ramping the fair rate which are denoted by *rampUpCoef* and *rampDnCoef* and are equal in the default mode of operation. Here, we have assumed that  $\text{rampUpCoef} = \text{rampDnCoef} = \beta^{-1}$ .
- ▶  $\tau_\ell$  and  $\tau_h$  ( $\tau_\ell < \tau_h < C$ ) are two rate thresholds used for congestion detection and fair rate adjustments and denoted as *rateLowThreshold* and *rateHighThreshold*, respectively, in the RPR standard.
- ▶  $\ell_i(k)$ , called *allowedRate*, is the rate at which node  $i$  is allowed to add local FE traffic to the ring in control interval  $k$ .
- ▶  $\ell_i^c(k)$ , called *allowedRateCongested*, is the rate at which node  $i$  is allowed to add local FE traffic to the ring in control interval  $k$  intended for destinations *beyond* the downstream congested node.
- ▶  $\Delta_i(k)$  is the *number of active nodes* at node  $i$  in control interval  $k$ . A node is called active at node  $i$  if it has sent traffic during control interval  $k$  transiting node  $i$ . Node  $i$  is considered active when calculating  $\Delta_i(k)$ .

*Remark 2.2.2.* System parameters such as  $T$ ,  $\alpha$ ,  $\beta$ ,  $\tau_\ell$ , and  $\tau_h$  are set at each node and we can assume that they do not change during the operation of the RPR network.

*Remark 2.2.3.* The rate of local FE traffic added by a given node is restricted by two rate limiters at that node: (i) *allowedRate*, and (ii) *allowedRateCongested*. It is always limited to the *allowedRate* of that node. Furthermore, when congestion exists, the rate of local FE traffic transiting the downstream congested node is also limited to *allowedRateCongested*.

In the following, we describe fair rate calculation and rate adjustment in RPR-AM and RPR-CM.

### Aggressive Mode

RPR-AM is devised based on the Spatial Reuse Protocol (SRP) introduced in [52] and favors achieved throughput over fair rate stability [4, 5, 44]. In RPR-AM, each node calculates its fair rate at the end of every control interval. When node  $i$  detects congestion in control interval  $k$ , it sets its fair rate to the low-pass filtered value of its add-rate, that is,

$$F_i(k) = \tilde{r}_i^a(k). \quad (2.2.3)$$

The *allowedRate* of node  $i$  in RPR-AM is always set to the unreserved rate, i.e.  $\ell_i(k) = C, \forall k$ . Moreover, the *allowedRateCongested* of a node is always set to the advertised fair rate of its downstream congested node. When congestion does not exist at the end of control interval  $k - 1$ , the *allowedRateCongested* of node  $i$  is ramped up as

$$\ell_i^c(k) = \beta \cdot C + (1 - \beta) \cdot \ell_i^c(k - 1). \quad (2.2.4)$$

### Conservative Mode

RPR-CM provides highly dampened fair rate adjustments which favors fair rate stability over achieved throughput [4, 5, 7, 44]. In RPR-CM, before calculating a new fair

rate, a congested node waits to ensure that the effect of the previous fair rate adjustment is observed. This waiting period is known as *fairness round-trip time* (FRTT). In other words, FRTT is the time needed for the congested node to observe the effect of its last advertised fair rate on the add-traffic of the furthest node contributing to congestion. In RPR-CM, node  $i$  calculates its fair rate at the end of control interval  $k$  as follows:

- If node  $i$  is in the *uncongested* state and detects congestion for the first time, its fair rate is initially set to the unreserved rate divided by the number of active nodes, i.e.  $C/\Delta_i(k)$ . Node  $i$  then enters the *congested* state.
- If node  $i$  is in the congested state and at least an FRTT has elapsed since the last fair rate adjustment, the new fair rate, i.e.  $F_i(k)$ , is calculated according to the value of  $r_i^u(k)$  as follows:

- if  $r_i^u(k) > \tau_h$ , the fair rate is ramped down such that

$$F_i(k) = (1 - \beta) \cdot F_i(k - 1). \quad (2.2.5)$$

- if  $r_i^u(k) < \tau_\ell$ , the fair rate is ramped up and we have

$$F_i(k) = \min \{C, F_i(k - 1) + \beta \cdot (C - r_i^u(k))\}. \quad (2.2.6)$$

- if  $\tau_\ell \leq r_i^u(k) \leq \tau_h$ , we have

$$F_i(k) = F_i(k - 1). \quad (2.2.7)$$

- If node  $i$  is in the congested state and its fair rate is fully ramped up (e.g. it is greater than 95% of the unreserved rate), node  $i$  enters the uncongested state and  $F_i(k) = C$ .

In RPR-CM, at the beginning of control interval  $k$ , node  $i$ , which is in the congested state, sets its *allowedRate* to its fair rate calculated at the end of control interval  $k - 1$ , i.e.  $\ell_i(k) = F_i(k - 1)$ . When node  $i$  is in the uncongested state, its *allowedRate* is gradually ramped up. In RPR-CM, the *allowedRateCongested* of a node is set similar to the one in RPR-AM.

*Remark 2.2.4.* Unlike RPR-AM, in RPR-CM a node remains in the congested state (even if congestion conditions are cleared) until its fair rate is fully ramped up. This provides a hysteresis in the transition between the congested and the uncongested states and decreases rate oscillations.

*Remark 2.2.5.* In RPR-CM, the fair rate of congested node  $i$  is decreased as long as  $r_i^u(k) > \tau_h$ . This means that the maximum link usage at the congested node in the steady-state is  $\tau_h$ . In addition, the fair rate does not change when  $\tau_\ell \leq r_i^u(k) \leq \tau_h$ . Hence, the bandwidth loss at the congested node in the steady-state can be as high as  $C - \tau_\ell$ .

## Chapter 3

# A New Fairness Reference Model for Resilient Packet Rings

The primary goal of bandwidth allocation schemes in a data network is to achieve high bandwidth utilization while maintaining a certain *fairness* in distributing bandwidth among users. The definition of fairness, however, varies according to stated objectives of bandwidth allocation in a network. In literature, there are a number of definitions for fairness which are based on different objective functions (e.g. see [53–55]). Thus, to examine fairness of a set of allocated rates, the notion and criteria of fairness should be introduced. The set of criteria under which fairness of a set of allocated rates can be examined is regarded as *fairness reference model*. Note that the fairness reference model does not necessarily specify how the fair solution can be achieved. In fact, a bandwidth allocation algorithm should be employed in order to achieve the fair solution. In this chapter, a new fairness reference model for resilient packet rings is proposed.

In RPR, the aim is to maintain *source based fairness* among competing nodes on any segment of the ring. That is, the available bandwidth on a given link should be distributed fairly among all competing nodes on that link. In other words, the aggregate allocated rate of all flows originating from a node should be considered for fairness determination on a given link. This means that maintaining fairness among

different nodes on a given link should not depend on how each node distributes its bandwidth share among its local flows on that link. The existing proposed fairness model for RPR, dictates to all nodes how they should distribute their bandwidth share among their local flows such that a certain fairness be maintained among local flows. Our proposed fairness reference model, however, provides fairness criteria which do not depend on the policy employed by each node to distribute the bandwidth share of that node among its local flows. In fact, we show that our proposed fairness reference model is the generalized form of the existing fairness reference model proposed for RPR. Based on the new fairness reference model, each node is allowed to implement its own policy in distributing its bandwidth share among its local flows to achieve different objectives. It is worth mentioning that this chapter does not address the bandwidth allocation algorithm which is used at RPR nodes. In fact, this chapter is only concerned with examining fairness of a solution which has been achieved using an arbitrary bandwidth allocation algorithm.

The outline of this chapter is as follows. In Section 3.1, we briefly review two traditional fairness reference models in the context of the ring networks. We also provide an overview of the existing RPR fairness reference model. We then discuss the bandwidth management principles in RPR in Section 3.2. Our new fairness reference model for RPR is presented in Section 3.3. Afterwards, in Section 3.4, we study different policies which may be used by a node to distribute its share of bandwidth among its local flows. Section 3.5 compares the existing RPR fairness reference model with our developed model. Some concluding remarks are given in Section 3.6.

## 3.1 Preliminaries

In this section, we start by introducing notations and stating assumptions which are used throughout this chapter. Then, two well-known fairness reference models are introduced in the context of packet ring networks. Finally, we review the existing RPR fairness reference model.

Our assumptions in all discussions in this chapter are as follows:

- ▶ All traffic rates are considered as instantaneous fluid rates as we examine fairness of allocated rates of flows in the network at a time instance.
- ▶ We only consider traffic on one of the ringlets as fairness control on each ringlet is performed independent of the one on the other ringlet.
- ▶ For simplicity, in this chapter we assume that nodes are numbered in an increasing order with respect to the direction of traffic.
- ▶ For graphical presentation of each scenario, we only show one of the ringlets which carries traffic. Furthermore, we only show the part of the ring which has traffic.
- ▶ We denote the link between two adjacent nodes  $i$  and  $i + 1$  as  $l(i, i + 1)$ . Note that the direction of traffic on link  $l(i, i + 1)$  is from node  $i$  to node  $i + 1$ .

**Definition 3.1.1.** A *flow* is referred to all unidirectional traffic between a certain pair of nodes. We denote such traffic between node  $i$  and node  $j$  ( $i < j$ ) as  $f(i, j)$ : The allocated rate of  $f(i, j)$  is denoted by  $r_{ij}$ , and  $d_{ij}$  represents the offered (demand) rate of flow  $f(i, j)$ .

For graphical presentation of scenarios in this thesis, we show a traffic flow from source node  $i$  to destination node  $j$  by an arrow starting from source node  $i$  and pointing to the destination node  $j$ .

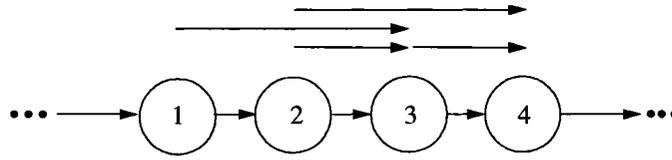


Figure 3.1: Illustration of max-min fairness

### 3.1.1 Max-Min Fairness

The max-min fairness model is one of the classical reference models for fairness in a network. In this fairness model, the rates of the flows are made as equal as possible subject to the constraints imposed by link capacities [56,57]. In other words, the goal in max-min fairness is to maximize the utilization of the links while allocating equal rates for competing flows. The following example clarifies the notion of max-min fairness.

**Example 3.1.1.** Consider the scenario depicted in Figure 3.1 in which all flows are infinite-demand and the capacity of all links is  $C$ . In this scenario, flows  $f(1,3)$ ,  $f(2,3)$ , and  $f(2,4)$  share bandwidth on link  $l(2,3)$ . To ensure fairness among these flows, the rate of each flow should be limited to  $\frac{C}{3}$ , i.e.  $r_{13} = r_{23} = r_{24} = \frac{C}{3}$ . On the other hand, flow  $f(3,4)$  can take over the unused bandwidth on link  $l(3,4)$  and its rate can be limited to  $\frac{2}{3} \cdot C$  as any lower limit would waste some of the capacity of link  $l(3,4)$ . Moreover, the rate of flow  $f(3,4)$  should not be higher than  $\frac{2}{3} \cdot C$  as it would limit flow  $f(2,4)$  more. ◀

Consider a network which consists of a set of nodes  $\mathbf{N}$  and a set of links  $\mathbf{L}$ . Assume that the capacity of each link  $l \in \mathbf{L}$  is denoted by  $C_l$  and let  $\mathbf{R} = \{r_{ij}\}$  be the set of allocated rates of all flows in the network.

**Definition 3.1.2.** (p. 526 in [56]) A set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is *feasible* if the following constraints are satisfied:

$$r_{ij} \geq 0, \quad \forall i, j \in \mathbf{N}, \quad (3.1.1)$$

$$\sum_{i,j} r_{ij} \leq C_l, \quad \forall l \in \mathbf{L} \text{ and all } f(i,j) \text{ crossing link } l. \quad (3.1.2)$$

In the following, we provide the formal definition of max-min fairness in a network.

**Definition 3.1.3** (p. 526 in [56]). A set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is said to be max-min fair if it is feasible, and if for each flow  $f(i,j)$ ,  $r_{ij}$  cannot be increased, while maintaining feasibility of  $\mathbf{R}$ , without decreasing  $r_{i'j'}$  for some flow  $f(i',j')$ , for which  $r_{i'j'} \leq r_{ij}$ .

It can be noted that according to the max-min fairness model, the allocated rate of a flow should be increased as long as it does not cause a decrease in the allocated rate of another flow that is already allocated a smaller or equal rate. In other words, flows with minimum allocated rates should not be further restricted according to this fairness model.

An iterative algorithm for computing the max-min allocated rates in an arbitrary network has been introduced in [56]. There has also been a great deal of work on distributed algorithms that maintain max-min fairness in various networks (see e.g. [58–63]). These algorithms are mostly based on the concept of either explicit rate calculations or end-to-end window flow control. However, there is always a trade-off between the implementation complexity and the resulting fairness of such algorithms.

### Single Resource Max-Min Fairness

Consider a special case of the max-min fair allocation in which all flows share the bandwidth of a single resource with capacity  $C_l$ . In this case, the max-min fair share of each flow is equal to the rate at which that flow would be served in a GPS<sup>1</sup> ([64]) server operating at rate  $C_l$ . In [6], a simple algorithm is proposed to calculate the allocated max-min fair rates of flows in a single resource scenario. The following example clarifies the concept of the single source max-min fairness.

---

<sup>1</sup>Generalized Processor Sharing.

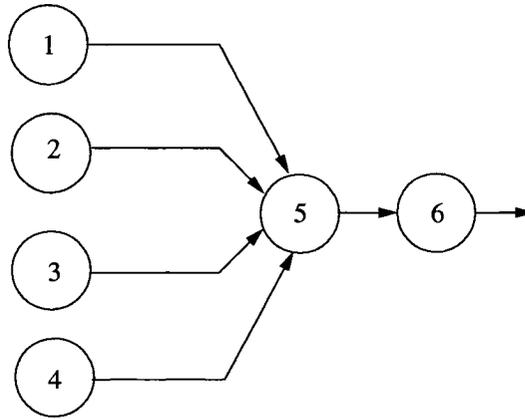


Figure 3.2: An example to illustrate the single resource max-min fairness

**Example 3.1.2.** Consider the scenario depicted in Figure 3.2 in which nodes 1, 2, 3, and 4 send traffic to node 6 sharing link  $l(5,6)$  with capacity  $C$ . Assume that flow  $f(1,6)$  is infinite-demand, and  $d_{26} = 0.3 \cdot C$ ,  $d_{36} = 0.1 \cdot C$ , and  $d_{46} = 0.2 \cdot C$ . The allocated max-min rates for this scenario are as follows:  $r_{16} = 0.4 \cdot C$ ,  $r_{26} = 0.3 \cdot C$ ,  $r_{36} = 0.1 \cdot C$ , and  $r_{46} = 0.2 \cdot C$ . Note that if all flows were infinite-demand, the max-min fair rate of all flows would be equal to  $0.25 \cdot C$ . However, in this scenario, the demand of flows  $f(3,6)$  and  $f(4,6)$  is less than  $0.25 \cdot C$  and, as a result, the unused bandwidth is allocated to flows  $f(1,6)$  and  $f(2,6)$  in order to maximize the link utilization. ◀

It is worth mentioning that in Example 3.1.2, the allocated bandwidth of the flows with the *minimum* allocation (i.e.  $f(3,6)$  and  $f(4,6)$ ) has been *maximized*, hence the name ‘max-min’ fairness.

### 3.1.2 Proportional Fairness

It is argued in [54] that most of flow control protocols such as TCP<sup>2</sup>, which are based on the additive-increase multiplicative-decrease congestion avoidance principle, do

---

<sup>2</sup>Transport Control Protocol.

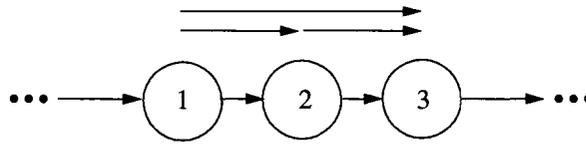


Figure 3.3: Illustration of proportional fairness

not tend to achieve max-min fairness. In particular, TCP favors short flows<sup>3</sup> in allocating bandwidth when compared to long flows. The proportional fairness model is an alternative fairness reference model, which aims at allocating more bandwidth to shorter flows (in most scenarios) and maximizing a logarithmic function of the overall allocated rates of the flows [53, 54, 65]. In the following, we provide the formal definition of proportional fairness.

**Definition 3.1.4** (see [54]). A set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is *proportionally fair* if it is feasible and if it maximizes the objective function  $\sum_{i,j} \log(r_{ij})$ .

**Example 3.1.3.** Consider the scenario depicted in Figure 3.3 in which flow  $f(1, 3)$  competes with flows  $f(1, 2)$  and  $f(2, 3)$  for the bandwidth on links  $l(1, 2)$  and  $l(2, 3)$ , respectively. Assume that all flows are infinite-demand and the capacity of all links is  $C$ . The proportionally fair solution for this example is  $r_{12} = r_{23} = \frac{2}{3} \cdot C$ , and  $r_{13} = \frac{C}{3}$ , whereas the max-min fair solution is  $r_{12} = r_{23} = r_{13} = \frac{C}{2}$ . It can be seen that in case of proportional fairness, the long flow (i.e.  $f(1, 3)$ ) is allocated proportionally less bandwidth compared to the short flows. ◀

### 3.1.3 Ring Ingress-Aggregated with Spatial Reuse (RIAS) Fairness

In this section, we briefly review the RIAS fairness model which has been proposed in [6] as a fairness reference model for RPR.

<sup>3</sup>A short flow is defined as a flow which passes through relatively less number of hops.

The objective of bandwidth allocation in RPR is to provide source-based fairness among the competing nodes on any segment of the ring. Therefore, per-flow fairness models such as proportional fairness and max-min fairness cannot be used as the reference fairness model in RPR. In these reference models, fairness is maintained among the individual source-destination flows and not among the source nodes. In other words, the granularity of flows for fairness determination in these reference models is a single source-destination flow. In order to provide source-based fairness, the *aggregate* rate of all flows originating from each ingress node (and not the rate of each individual flow originating from that node) should be considered for fairness determination at each link.

**Definition 3.1.5.** An ingress-aggregated (IA) flow at node  $n$  originated from node  $i \leq n$  is denoted by  $IA(i, n)$  and defined as the aggregate of all flows originating from source node  $i$  and destined to nodes beyond node  $n$ . The allocated rate of IA-flow  $IA(i, n)$  is denoted by  $r_i^n$  and obtained as

$$r_i^n = \sum_{j>n} r_{ij}. \quad (3.1.3)$$

In fact, an IA-flow  $IA(i, n)$  is the aggregate of flows originating from node  $i$  and transiting node  $n$ . The allocated rate of IA-flow  $IA(i, n)$  is measured on link  $l(i, i + 1)$ . In the RIAS fairness reference model, an IA-flow is considered as the level of granularity for fairness determination at each link. In fact, fairness should be maintained among IA-flows at each link in the RIAS fairness model. Further, spatial reuse should be maximized subject to the fairness constraint.

In the RIAS fairness model, it is assumed that each node can control the rate of each flow separately. Moreover, all flows are assumed to be infinite-demand. In the following, we review the formal definition of the RIAS fairness model.

**Definition 3.1.6** (Definition 1 in [6]). A set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is said to be RIAS-fair if it is feasible and if for each flow  $f(i, j)$ ,  $r_{ij}$  cannot be increased while maintaining feasibility without decreasing  $r_{i'j'}$  for some flow  $f(i', j')$  for which

$$r_{i'j'} \leq r_{ij}, \quad \text{when } i = i', \quad (3.1.4)$$

$$r_i^n \leq r_{i'}^n, \quad \text{at some common node } n \text{ otherwise.} \quad (3.1.5)$$

In the following, the necessary conditions in the definition of RIAS fairness are elaborated more.

- According to Inequality (3.1.4), in order to maintain feasibility, if the rate of flow  $f(i, j)$  increases, the rate of flow  $f(i', j')$  originated from the same source node ( $i = i'$ ) should be decreased. This inequality ensures the fairness among flows within an IA-flow. In other words, max-min fairness is maintained among the flows within an IA-flow.
- In the second case, flows  $f(i, j)$  and  $f(i', j')$  are originated from different source nodes ( $i \neq i'$ ). Increasing the rate of flow  $f(i, j)$  in this case results in decreasing the rate of  $f(i', j')$  where at some node  $n$ , IA-flow  $IA(i', n)$  has already received a lower bandwidth compared to IA-flow  $IA(i, n)$ . Inequality (3.1.5) ensures that fairness is maintained among different IA-flows. In this case, max-min fairness is maintained among IA-flows throughout the ring.

Therefore, a set of allocated rates is considered fair according to the RIAS fairness model, when max-min fairness is maintained among the IA-flows throughout the ring as well as among flows within an IA-flow. In the following, we provide an example to illustrate the RIAS fairness model.

**Example 3.1.4** (See [6]). Consider the scenario depicted in Figure 3.4 and assume that all flows are infinite-demand. The RIAS-fair solution for this scenario is  $\mathbf{R} =$

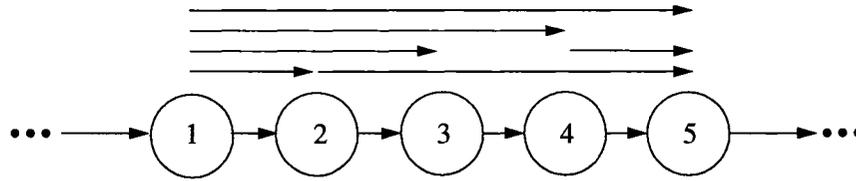


Figure 3.4: Illustration of RIAS fairness

$\{r_{12} = r_{25} = r_{45} = 0.4 \cdot C, r_{13} = r_{14} = r_{15} = 0.2 \cdot C\}$ . If we consider flow  $f(1,2)$ , its rate cannot be increased without decreasing the rate of flow  $f(1,3)$ ,  $f(1,4)$ , or  $f(1,5)$ , where  $r_{12}$  is greater than  $r_{13}$ ,  $r_{14}$ , and  $r_{15}$ . Moreover, the rate of flow  $f(2,5)$  (respectively,  $f(4,5)$ ) cannot be increased without decreasing the rate of flow  $f(4,5)$  (respectively,  $f(2,5)$ ) or flow  $f(1,5)$ , where  $r_2^4 \geq r_4^4$  (respectively,  $r_4^4 \geq r_2^4$ ) and  $r_2^4 \geq r_1^4$  (respectively,  $r_4^4 \geq r_1^4$ ). Finally, the rate of flows  $f(1,3)$ ,  $f(1,4)$ , or  $f(1,5)$  cannot be increased without decreasing the rate of flow  $f(2,5)$  where  $r_1^2 \geq r_2^2$ . ◀

It can be noted that according to the RIAS fairness reference model, max-min fairness should be maintained among the local flows at each node. However, this condition is not required by the RPR standard and each node is allowed to implement its own policy in allocating bandwidth to its local flows. Therefore, the RIAS fairness reference model is not a general fairness model for RPR as it is not valid for an arbitrary policy employed for bandwidth allocation among local flows of a node. We elaborate this issue in the following section by reviewing basic principles of fairness and bandwidth allocation in RPR.

## 3.2 Principles of Bandwidth Allocation and Fairness in RPR

In RPR, bandwidth is allocated to each flow based on a two-level process. At the first level, each congested node allocates bandwidth to competing IA-flows by calculating

a fair rate. At the second level, each node allocates bandwidth to its local flows based on the received fair rates from downstream congested nodes. In the following, we discuss the objective of bandwidth allocation at each level in more details.

### 3.2.1 Bandwidth Allocation at Ring Level

At the ring level, bandwidth on each link is distributed among competing nodes on that link. According to the RPR standard, the objective of bandwidth allocation at the ring level is to provide source-based (*inter-station*) fairness among competing nodes on each link. Therefore, at the ring level, the aggregate traffic originating from an ingress node (IA-flow) is considered for fairness determination on each link and not a source-destination flow. The bandwidth allocation algorithm employed at each RPR node is in charge of allocating bandwidth at the ring level to IA-flows and maintaining source-based fairness throughout the ring. At the ring level, the fair rate of a node is the maximum rate at which an IA-flow can transit that node.

### 3.2.2 Bandwidth Allocation at Source Level

At the ring level, bandwidth is allocated to all nodes competing at a congested node such that a source-based fairness is maintained. Then, at the source level, each node distributes its share of bandwidth among its local flows. In this case, each node should ensure that the aggregate rate of the its local flows transiting a downstream node does not exceed the advertised fair rate of that node. In fact, the bandwidth allocation at the source level should conform to the restrictions imposed by the bandwidth allocation at the ring level. Nonetheless, there are many policies that can be employed for bandwidth allocation at the source level. These policies may consider different objective functions in bandwidth allocation at the source level while conforming to the ring-level bandwidth allocation requirements.

The RPR standard does not specify the policy which should be employed for bandwidth allocation at the source level. This means that each node is allowed to employ any proprietary policy for bandwidth allocation at the source level. Therefore, various objective functions and fairness criteria may be considered for allocating bandwidth to the local flows at the source level. However, to avoid implementing complex algorithms, some nodes may choose simple schemes for bandwidth allocation at the source level. Such schemes are not necessarily very efficient in terms of fully utilizing the bandwidth share of each node; but they may be preferred for their simple implementation over other efficient complex schemes.

**Definition 3.2.1.** In this thesis, we refer to a source-level bandwidth allocation policy employed at an RPR node as *source behavior*.

The following example clarifies the concept of bandwidth allocation at ring level and source level in RPR.

**Example 3.2.1.** Consider the scenario depicted in Figure 3.5 in which all flows are infinite-demand and the capacity of all links is  $C$ . In this scenario, flow  $f(1,3)$  competes for the bandwidth on link  $l(2,3)$  with the flows originating from node 2, i.e.  $f(2,3)$  and  $f(2,5)$ . In order to maintain source-based fairness, the bandwidth on link  $l(2,3)$  should be equally divided between traffic of node 2 and node 3. That is, at the ring level, IA-flows  $IA(1,2)$  and  $IA(2,2)$  should receive equal bandwidth on link  $l(2,3)$  (i.e.,  $r_1^2 = r_2^2$ ). Hence,  $r_1^2 = r_{13} = 0.5 \cdot C$  and  $r_2^2 = r_{23} + r_{25} = 0.5 \cdot C$ . This means that at the ring level, the allocated bandwidth of node 2 and node 3 on link  $l(2,3)$  is equal to  $0.5 \cdot C$ . However, at the source level, node 2 may divide its bandwidth share to  $f(2,3)$  and  $f(2,5)$  according to different source behaviors. For instance, node 2 may equally divide its share of bandwidth to its flows, i.e.  $r_{23} = 0.25 \cdot C$  and  $r_{25} = 0.25 \cdot C$ . As long-flows usually face more congestion along their path to the destination, another policy for node 2 is, for example, to allocate proportionally more bandwidth to its long

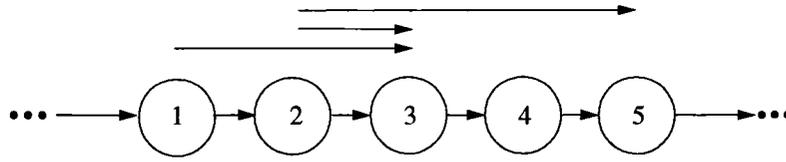


Figure 3.5: Illustration of ring-level and source-level bandwidth allocation in RPR

flow. For instance, if  $r_{25} = 0.375 \cdot C$  and  $r_{23} = 0.125 \cdot C$ , the aggregate rate of local traffic of node 2 is still  $0.5 \cdot C$  (which satisfies the fairness condition at the ring level) while the allocated rate of flow  $f(2, 5)$  is three times more than that of flow  $f(2, 3)$ .

In the RIAS fairness reference model, however, max-min fairness should be maintained among local flows at each node as well as among IA-flows. Thus, the only RIAS-fair solution for this example is  $r_{13} = 0.5 \cdot C$  and  $r_{23} = r_{25} = 0.25 \cdot C$ . ◀

It is evident from Example 3.2.1 that as the RPR standard allows different source behavior to be employed for bandwidth allocation at the source level, various fair solutions may exist. According to the RIAS fairness reference model, however, only one of the solutions is considered fair – the one in which bandwidth share of each node is allocated among local flows of that node in a max-min fashion. In other words, in the RIAS fairness model, nodes are not allowed to employ any arbitrary policy for distributing their share of bandwidth among their local flows. Therefore, the RIAS fairness model is not a general fairness reference model for RPR. In fact, a fairness reference model for RPR should be independent of the source behavior employed at each node. In the next section, we define a general fairness reference model for RPR.

### 3.3 Ring Ingress-Aggregated Max-Min Fairness

In this section, we propose the Ring Ingress-Aggregated Max-Min (RIAMM) fairness model as a new fairness reference model for RPR. This model is intended to determine the fairness of a set of rate allocations at the *ring level* for any arbitrary source

behavior.

In the RIAMM fairness model, the objective is to maintain source-based fairness among competing nodes in a max-min sense on any segment of the ring. The intuitive justification behind this objective is that when max-min fairness is maintained among traffic of competing nodes on any segment of the ring, source-based fairness is maintained while bandwidth utilization is maximized. In this model, the aggregate traffic originating from a single node (i.e. an IA-flow) is considered for fairness determination and bandwidth allocation on any segment of the ring.

In the following, we first consider a simple case where each node only sends traffic to a single destination. We then generalize the model for a multiple-destination case where each node may send traffic to multiple nodes. In all cases, we assume that the capacity (unreserved rate) of all links is equal to  $C$ .

### 3.3.1 Single-Destination Scenarios

In this section, we assume that each node sends traffic only to a single destination. Hence, given that flow  $f(i, j)$ ,  $i < j$ , is the only flow originating from node  $i$ , we have  $r_{ij'} = 0$ ,  $\forall j' \neq j$ . In this case, in order to achieve a source-based fairness among ring nodes, fairness should be maintained among source-destination flows throughout the ring. In other words, source-based fairness and per-flow fairness are essentially equivalent in this scenario as each node only has a single flow.

In addition to maintaining source-based fairness in allocating bandwidth in RPR, it is desirable to utilize the available bandwidth as much as possible. In a single-destination scenario, these goals can be achieved by maintaining max-min fairness among the competing flows on each link. The following example illustrates this point.

**Example 3.3.1.** *Consider a scenario depicted in Figure 3.6 in which all nodes 1, 2,*

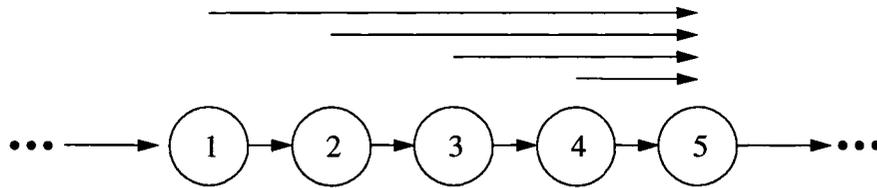


Figure 3.6: A single-destination scenario

3, and 4 send traffic only to node 5 sharing link  $l(4, 5)$  with capacity  $C$ .<sup>4</sup> Assume that flow  $f(1, 5)$  is infinite-demand and  $d_{25} = 0.3 \cdot C$ ,  $d_{35} = 0.1 \cdot C$ , and  $d_{45} = 0.2 \cdot C$ . This scenario is similar to the one in Example 3.1.2, but in the form of an RPR network. In order to maximize the utilization while maintaining fairness in bandwidth allocation, each node should receive its max-min fair share of the bandwidth on link  $l(4, 5)$ . Therefore,  $r_{15} = 0.4 \cdot C$ ,  $r_{25} = 0.3 \cdot C$ ,  $r_{35} = 0.1 \cdot C$ ,  $r_{45} = 0.2 \cdot C$ . In this case, the allocated rate of each flow is equal to the rate at which that flow is served along the other flows at a GPS server operating at rate  $C$ .

According to the RPR bandwidth allocation scheme, node 4 should advertise a single fair rate to its upstream nodes in order to achieve source-based fairness. One can notice that if node 4 advertises  $0.4 \cdot C$  as its fair rate to the upstream nodes, the max-min fairness among the flows is achieved. The reason is that once the fair rate is advertised, node 1 limits its rate to the advertised fair rate and we have  $r_{15} = 0.4 \cdot C$ . Moreover, nodes 2, 3, and 4 limit their rates to  $0.4 \cdot C$ . However, their demands are less than  $0.4 \cdot C$  and, as a result, their rates are limited to their demands and we have  $r_{25} = d_{25} = 0.3 \cdot C$ ,  $r_{35} = d_{35} = 0.1 \cdot C$ , and  $r_{45} = d_{45} = 0.2 \cdot C$ . Hence, the fair rate of node 4 should be equal to the maximum of the max-min allocated rates of the flows transiting node 4, which is equal to  $0.4 \cdot C$  in this example. Any fair rate less than  $0.4 \cdot C$  will result in a lower utilization at link  $l(4, 5)$ . ◀

<sup>4</sup>This scenario is known as the parking-lot scenario and considered as one of the benchmark scenarios in performance evaluation of packet rings. We will also study this scenario in Chapter 4 and Chapter 5.

Assume that the set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is given in a single-destination scenario. The aim is to determine whether this rate allocation is fair according to the RPR fairness objectives. In the following, we develop the criteria which are satisfied only when  $\mathbf{R} = \{r_{ij}\}$  is a fair solution. Therefore, if  $\mathbf{R}$  does not satisfy those criteria, it is not a RIAMM-fair solution.

Let  $F_n$  be the advertised fair rate of node  $n$ . We define  $\mathbf{R}_n = \{r_{ij}, i \leq n < j\}$  as the set of allocated rates of all flows passing through node  $n$ . Note that  $\mathbf{R}_n$  includes the allocated rate of the flow originating from node  $n$  as well.

In general, according to the RPR standard and to maintain the ring-level fairness, the rate of local traffic of node  $i$  transiting node  $n$ ,  $i \leq n < j$ , should not exceed the fair rate of node  $n$ . This means that the allocated rate of flow  $f(i, j)$  should not exceed  $F_n$ ,  $i \leq n < j$ , as we have assumed that flow  $f(i, j)$  is the only flow originated from node  $i$  in this single-destination case. On the other hand, the allocated rate of each flow is upper-bounded by its demand. Hence, the allocated rate of flow  $f(i, j)$  is given by

$$r_{ij} = \min \left\{ d_{ij}, \min_{i \leq n < j} \{F_n\} \right\}, \forall i, j. \quad (3.3.1)$$

Now, we need to illustrate the relation between the calculated fair rate of node  $n$  and the allocated rate of flows transiting node  $n$ . The goal is to maintain max-min fairness in allocating bandwidth to competing nodes on each link. This goal can be achieved according to Example 3.3.1) when the fair rate of node  $n$  is equal to the maximum of the max-min allocated rates of the flows transiting node  $n$ . Hence,

$$F_n = \Phi(C, \mathbf{R}_n), \forall n, \quad (3.3.2)$$

where  $\Phi(C, \mathbf{X})$  is the *maximum* of the max-min fair shares of the flows with the set of demands  $\mathbf{X} = \{x_i\}$  when served at a single server with capacity  $C$ .<sup>5</sup> We refer to

---

<sup>5</sup>Note that in Equation (3.3.2),  $\mathbf{R}_n$  is considered as the set of bandwidth demands of the flows transiting node  $n$ . It can be easily verified that if  $\mathbf{R}$  is a fair solution, using  $\mathbf{R}_n$  instead of the actual demands in Equation (3.3.2) results in the same fair rate.

$\Phi(C, \mathbf{X})$  as the *max-min fair rate* of flows with the set of demands  $\mathbf{X} = \{x_i\}$  served at a server with capacity  $C$ . The following proposition presents one of the properties of  $\Phi(C, \mathbf{X})$ .

**Proposition 3.3.1.** *The max-min fair rate of the flows with set of demands  $\mathbf{X} = \{x_i\}$  when served at a single server with capacity  $C$  satisfies the following equations:*

$$\Phi(C, \mathbf{X}) = \begin{cases} \max_i \{x_i\}, & \text{if } \sum_i x_i < C, \\ \frac{1}{|\mathbf{X}| - |\mathbf{Y}|} \cdot (C - \sum_{\forall x_i \in \mathbf{Y}} x_i), & \text{otherwise,} \end{cases} \quad (3.3.3)$$

where  $\mathbf{Y} = \{x_i : x_i < \Phi(C, \mathbf{X})\}$  and  $|\cdot|$  is the number of elements in a set.

*Proof.* When  $\sum_i x_i < C$ , the total demand is less than the capacity and, hence, the fair share of each flow is equal to its demand. In this case, if the fair rate is set to  $\max_i \{x_i\}$ , the allocated rate of each flow will be equal to its demand.

When  $\sum_i x_i > C$ , the total demand is more than the service capacity and, as a result, the allocated rate of some flows is limited to the fair rate which is less than their demands. Assume that  $\Phi(C, \mathbf{X})$  is the fair rate calculated at the server to maintain max-min fairness in bandwidth allocation among the flows with the set of demands  $\mathbf{X} = \{x_i\}$ . We also assume that  $\mathbf{Y}$  is the set of the demands of the flows which are not limited to the fair rate, i.e. their demands are less than the fair rate and we have  $\mathbf{Y} = \{x_i : x_i < \Phi(C, \mathbf{X})\}$ . Hence, the allocated rates of flows with set of demands  $\mathbf{Y}$  are equal to their demands. This means that the total allocated rate to the flows with the set of demands  $\mathbf{Y}$  is  $\sum_{\forall x_i \in \mathbf{Y}} x_i$ . The remaining capacity, i.e.  $(C - \sum_{\forall x_i \in \mathbf{Y}} x_i)$ , is equally distributed among the flows which are limited to the fair rate, i.e. their allocated rate is equal to the fair rate. Hence, for the fair rate (i.e.  $\Phi(C, \mathbf{X})$ ) we have  $\Phi(C, \mathbf{X}) = (C - \sum_{\forall x_i \in \mathbf{Y}} x_i) / (|\mathbf{X}| - |\mathbf{Y}|)$ .  $\square$

Note that  $\Phi(C, \mathbf{X})$  cannot be calculated easily from Equation (3.3.3) as it has a recursive form. A number of iterative algorithms have been introduced in order to calculate  $\Phi(C, \mathbf{X})$  which can be found for example in [6, 56, 66].

### Adjusting Max-Min Fair Rate

In Example 3.3.1, we showed that to maintain max-min fairness in bandwidth allocation at node  $n$ , the fair rate of node  $n$  should be set to the maximum of the max-min allocated rates of the flows transiting node  $n$ . We then defined  $\Phi(C, \mathbf{R}_n)$  as the fair rate of node  $n$ . Note that when unused bandwidth exists at node  $n$  (i.e.  $\sum_{r_{ij} \in \mathbf{R}_n} r_{ij} < C$ ), we have  $F_n = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\}$  according to Equation (3.3.3). In such cases, it is not guaranteed that the unused ring bandwidth can always be reclaimed. The following example clarifies this issue.

**Example 3.3.2.** Consider the scenario discussed in Example 3.3.1 where max-min fairness is achieved and we have  $r_{15} = 0.4 \cdot C$ ,  $r_{25} = 0.3 \cdot C$ ,  $r_{35} = 0.1 \cdot C$ , and  $r_{45} = 0.2 \cdot C$ , where the advertised fair rate of node 4 is  $F_4 = 0.4 \cdot C$  and the bandwidth on link  $l(4, 5)$  is fully utilized. Now, assume that node 3 stops sending traffic, i.e.  $r_{35} = 0$ . In this case, according to Equations (3.3.2) and (3.3.3), the fair rate of node 4 is calculated as  $F_4 = \Phi(C, \mathbf{R}_4 = \{r_{15}, r_{25}, r_{45}\}) = \max\{r_{15}, r_{25}, r_{45}\} = 0.4 \cdot C$ . Similarly, we have  $F_3 = F_2 = F_1 = 0.4 \cdot C$ . In this scenario, node 1 is the only node which can reclaim the unused bandwidth. However, the fair rates are equal to  $0.4 \cdot C$  which prevents node 1 from increasing its rate to reclaim the unused bandwidth on link  $l(4, 5)$ . Hence, the fair rates should be increased to allow node 4 to reclaim the unused bandwidth. ◀

Therefore, when unused bandwidth exists on link  $l(n, n+1)$ ,  $F_n$  cannot be obtained from Equation (3.3.2). In such cases, the allocated rate of a flow transiting node  $n$  should not be restricted by  $F_n$ . Hence,  $F_n$  should be greater than any of the allocated

rates of flows in  $\mathbf{R}_n$  to ensure that the unused bandwidth can be claimed. That is, when  $\sum_{r_{ij} \in \mathbf{R}_n} r_{ij} < C$ , we should have  $F_n > \Phi(C, \mathbf{R}_n) = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\}$ . In fact, when  $\mathbf{R}$  is a fair solution and bandwidth on link  $l(n, n+1)$  is not fully utilized, the fair rate of node  $n$  can be set to any value which is larger than the allocated rates in  $\mathbf{R}_n$ . The reason is that in such cases, all flows transiting node  $n$  are either limited to the fair rate of some other nodes (which have smaller fair rates compared to the fair rate of node  $n$ ), or do not have enough demand to utilize the unused bandwidth on link  $l(n, n+1)$ . The following example clarifies this issue.

**Example 3.3.3.** *Consider the scenario discussed in Example 3.3.1 and assume that a max-min fair solution is achieved and we have  $r_{15} = 0.4 \cdot C$ ,  $r_{25} = 0.3 \cdot C$ ,  $r_{35} = 0.1 \cdot C$ ,  $r_{45} = 0.2 \cdot C$ . In this case, the bandwidth on link  $l(4, 5)$  is fully utilized and we have  $F_4 = \Phi(C, \mathbf{R}_4 = \{r_{15}, r_{25}, r_{35}, r_{45}\}) = 0.4 \cdot C$ . However, the bandwidth on links  $l(1, 2)$ ,  $l(2, 3)$ , and  $l(3, 4)$  is not fully utilized. As  $\Phi(C, \mathbf{R}_n) = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\} = r_{15} = 0.4 \cdot C$ ,  $n = 1, 2, 3$ , the fair rate of nodes 1, 2, and 3 is set to a value greater than  $0.4 \cdot C$ . In this case, the allocated rate of flow  $f(1, 5)$  is only limited to the fair rate of node 4 as  $F_4 = 0.4 \cdot C$  and  $F_n > 0.4 \cdot C$ ,  $n = 1, 2, 3$ . Other flows do not have enough demand to claim the unused bandwidth on links  $l(1, 2)$ ,  $l(2, 3)$ , and  $l(3, 4)$ . In fact, as long as  $F_4 = 0.4 \cdot C$  and  $F_n > 0.4 \cdot C$ ,  $n = 1, 2, 3$ , the fair solution is achieved in this scenario. ◀*

As we stated before, when bandwidth is not fully utilized on link  $l(n, n+1)$ , the fair rate of node  $n$  can be set to any value greater than  $\Phi(C, \mathbf{R}_n) = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\}$ . However, we can find a maximum theoretical value for  $F_n$  in such cases. The unused bandwidth on link  $l(n, n+1)$  is equal to  $(C - \sum_{i \leq n < j} r_{ij})$  and the maximum of allocated rates of flows transiting node  $n$  is given by  $\Phi(C, \mathbf{R}_n) = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\}$ . In other words, at least one flow is sending traffic with rate  $\Phi(C, \mathbf{R}_n)$  transiting node  $n$ . If this flow could utilize the whole unused bandwidth on link  $l(n, n+1)$ , the fair rate

of node  $n$  would be  $\Phi(C, \mathbf{R}_n) + (C - \sum_{i \leq n < j} r_{ij})$ . For all other cases where the unused bandwidth could be distributed among the flows transiting node  $n$ , the fair rate of node  $n$  would be less than  $\Phi(C, \mathbf{R}_n) + (C - \sum_{i \leq n < j} r_{ij})$  and greater than  $\Phi(C, \mathbf{R}_n)$ . Therefore, the fair rate of node  $n$  in Equation (3.3.2) is adjusted as follows:

$$F_n = \Phi^+(C, \mathbf{R}_n, \eta) \equiv \Phi(C, \mathbf{R}_n) + \eta \cdot (C - \sum_{i \leq n < j} r_{ij})^+, \forall n, \quad (3.3.4)$$

where  $0 < \eta \leq 1$  and  $(x)^+ = \max\{x, 0\}$ . Note that in Equation (3.3.4), when bandwidth is fully utilized on link  $l(n, n+1)$  (i.e.  $\sum_{i \leq n < j} r_{ij} = C$ ), the fair rate of node  $n$  is still obtained as  $F_n = \Phi(C, \mathbf{R}_n)$  and fairness is maintained among the flows in a max-min sense at node  $n$ . On the other hand, when bandwidth is not fully utilized on link  $l(n, n+1)$ , the fair rate of node  $n$  is set to a value which is greater than  $\Phi(C, \mathbf{R}_n)$  for any  $\eta$ ,  $0 < \eta \leq 1$ . We refer to  $\Phi^+(C, \mathbf{R}_n, \eta)$  as the *adjusted* max-min fair rate with parameter  $\eta$  for the set of rates  $\mathbf{R}_n$  when served at a single server with capacity  $C$ .

In a single-destination scenario, a set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is RIAMM-fair, if it is feasible and if Equations (3.3.1) and (3.3.4) are simultaneously satisfied. In the following example, we show how fairness of a solution can be examined through the developed fairness reference model.

**Example 3.3.4.** Consider the scenario discussed in Example 3.3.1. Assume that  $\mathbf{R} = \{r_{15} = 0.3 \cdot C, r_{25} = 0.3 \cdot C, r_{35} = 0.1 \cdot C, r_{45} = 0.2 \cdot C\}$ . The goal is to see if  $\mathbf{R}$  is a RIAMM-fair solution. We have  $\mathbf{R}_1 = \{r_{15}\}$ ,  $\mathbf{R}_2 = \{r_{15}, r_{25}\}$ ,  $\mathbf{R}_3 = \{r_{15}, r_{25}, r_{35}\}$ , and  $\mathbf{R}_4 = \{r_{15}, r_{25}, r_{35}, r_{45}\}$ . Consequently,  $\Phi(C, \mathbf{R}_n) = \max\{r_{ij} : r_{ij} \in \mathbf{R}_n\} = 0.3 \cdot C$ ,  $n = 1, 2, 3, 4$ . We can now calculate the fair rates of all nodes. As all links have unused bandwidth, the fair rate of node  $n$ ,  $n = 1, 2, 3, 4$ , should be set to a value greater than  $\Phi(C, \mathbf{R}_n) = 0.3 \cdot C$ . We assume that  $\eta = 0.5$  and calculate the fair rate of all nodes from Equation (3.3.4). Thus,  $F_1 = 0.65 \cdot C$ ,  $F_2 = 0.5 \cdot C$ ,  $F_3 = 0.45 \cdot C$ ,

and  $F_4 = 0.35 \cdot C$ . Now, we should check if Equation (3.3.1) is satisfied as well. By substituting the fair rates in Equation (3.3.1) we have  $r_{15} = 0.35 \cdot C$ ,  $r_{25} = 0.3 \cdot C$ ,  $r_{35} = 0.1 \cdot C$ , and  $r_{45} = 0.2 \cdot C$ , which is different from the allocated rates in  $\mathbf{R}$  as  $r_{15}$  has changed. Therefore,  $\mathbf{R}$  is not a RIAMM-fair solution. It can be easily verified that  $\mathbf{R} = \{r_{15} = 0.4 \cdot C, r_{25} = 0.3 \cdot C, r_{35} = 0.1 \cdot C, r_{45} = 0.2 \cdot C\}$  is a RIAMM-fair solution as it satisfies Equations (3.3.1) and (3.3.4) simultaneously. Note that for each  $\eta$ ,  $0 < \eta \leq 1$ , we would obtain a different  $r_{15}$  which would be greater than  $0.3 \cdot C$  and conclude that  $\mathbf{R}$  is not a fair solution. ◀

In this section, we developed the RIAMM fairness criteria based on the assumption that each node sends traffic to a single destination. However, in a general case, each node may have several flows destined to multiple destinations on the ring. In the next section, the RIAMM fairness criteria is generalized for multiple-destination scenarios.

### 3.3.2 Multiple-Destination Scenarios

In this section, we define the RIAMM fairness reference model for a general case in which each node may send traffic to multiple destinations. According to the RPR fairness principles, the aggregate rate of traffic originating from a source (ingress) node is considered for bandwidth allocation and fairness determination on each link throughout the ring. In order to maintain fairness among different nodes in RPR, the allocated rate of an IA-flow transiting an arbitrary node  $n$  must be equal to or less than the calculated fair rate of node  $n$ . Moreover, as explained for single-destination scenarios, max-min fairness in bandwidth allocation should be maintained among IA-flows transiting node  $n$ .

Assume that  $\check{\mathbf{R}} = \{r_i^n, \forall i, n\}$  is the set of allocated rates of IA-flows, which can be obtained from the set of allocated rates  $\mathbf{R} = \{r_{ij}, \forall i, j\}$  using Equation (3.1.3), and let  $\check{\mathbf{R}}_n = \{r_i^n, \forall i \leq n\}$  be the set of allocated rates of all IA-flows transiting node

$n$ . (Note that  $\check{\mathbf{R}}_n$  includes the rate of the IA-flow originating from node  $n$  as well.) In the following, we formally define the RIAMM fairness reference model for a set of allocated rates.

**Definition 3.3.1.** A feasible set of allocated rates  $\mathbf{R} = \{r_{ij}, \forall i, j\}$  (or alternatively a set of IA allocated rates  $\check{\mathbf{R}} = \{r_i^n, \forall i, n\}$ ) is defined to be RIAMM-fair if the following criteria are met simultaneously:

$$r_i^n \leq F_n, \forall i, n, \quad (3.3.5)$$

$$F_n = \Phi^+(C, \check{\mathbf{R}}_n, \eta) \equiv \Phi(C, \check{\mathbf{R}}_n) + \eta \cdot (C - \sum_{i \leq n} r_i^n)^+, \forall n, \quad (3.3.6)$$

where  $0 < \eta \leq 1$ .

The following properties can be observed from the definition of the RIAMM fairness reference model:

- Based on Inequality (3.3.5), the aggregate allocated rate of traffic originating from an arbitrary node  $i$  and transiting node  $n$  cannot be larger than the fair rate of node  $n$ .
- At each node  $n$ , fairness is maintained among IA-flows from different nodes in a max-min sense according to Equation (3.3.6).
- When bandwidth is fully utilized on link  $l(n, n+1)$  (i.e.  $\sum_{i \leq n} r_i^n = C$ ), the fair rate of node  $n$  is obtained as  $\Phi(C, \check{\mathbf{R}}_n)$  regardless of the value of  $\eta$ . When unused bandwidth exists on link  $l(n, n+1)$  (i.e.  $\sum_{i \leq n} r_i^n < C$ ), the fair rate of node  $n$  is set to a value which is greater than the allocated rate of any IA-flow  $IA(i, n)$  for any  $\eta$ ,  $0 < \eta \leq 1$ . This ensures that the allocated rate of IA-flow  $IA(i, n)$  is not limited to  $F_n$  when unused capacity exists on link  $l(n, n+1)$ .

- In this fairness reference model, the rates of IA-flows are considered for fairness determination at each node and not the rate of each individual source-destination flow. Therefore, this fairness reference model does not depend on any particular source behavior at each node as the rate of each individual flow is not visible in Inequality (3.3.5) and Equation (3.3.6). Hence, each node can employ any policy to allocate its share of bandwidth among its local flows as long as Inequality (3.3.5) is satisfied.

*Remark 3.3.1.* Note that the fairness criteria in Inequality (3.3.5) and Equation (3.3.6) are only used to determine whether the set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is fair or not. When unused capacity exists on link  $l(n, n + 1)$ , we only need to make sure that the fair rate of node  $n$  is greater than the allocated rate of IA-flow  $IA(i, n)$ ,  $i \leq n$ , (i.e.  $F_n > r_i^n$ ) or, equivalently, the allocated rate of each IA-flow  $IA(i, n)$  is not restricted by the fair rate of node  $n$ . This can be achieved by choosing any  $\eta$ ,  $0 < \eta \leq 1$  in Equation (3.3.6). The reason is that when  $\sum_{i \leq n} r_i^n < C$ , from Equation (3.3.6) we obtain  $F_n > r_i^n$ ,  $i \leq n$ , for any  $\eta$ ,  $0 < \eta \leq 1$ . Therefore, in order to determine fairness of a set of allocated rates we can choose any  $\eta$ ,  $0 < \eta \leq 1$ , in this fairness model.

In order to determine whether a feasible set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is a RIAMM-fair solution, one should first calculate  $\check{\mathbf{R}}_n$  for each node  $n$ . Then, the fair rate of all nodes is calculated from Equation (3.3.6). Finally, it should be examined whether Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously at all nodes.

In the following, we study one of the properties of the RIAMM fairness model.

**Definition 3.3.2.** Given a feasible set of allocated rates  $\mathbf{R} = \{r_{ij}\}$ , node  $n$  (or alternatively link  $l(n, n + 1)$ ) is called a bottleneck for flow  $f(i, j)$ ,  $i \leq n < j$ , if  $r_i^n = F_n$  and  $r_{ij'} \leq r_{ij}$ ,  $\forall j' > n$ .

**Proposition 3.3.2.** *For a RIAMM-fair set of allocated rates  $\mathbf{R} = \{r_{ij}\}$ , node  $n$  is a bottleneck for at least one flow if and only if  $\sum_i r_i^n = C$ , that is, the bandwidth on link  $l(n, n+1)$  is fully utilized.*

*Proof.* To prove the necessity, let us assume that node  $n$  is a bottleneck for flow  $f(p, q)$ ,  $p \leq n < q$ . As  $\mathbf{R}$  is a RIAMM-fair solution, it is feasible and, as a result,  $\sum_i r_i^n \leq C$ . Assume that  $\sum_i r_i^n < C$ . Hence, it follows from Equation (3.3.6) that  $F_n > r_p^n$ . This means that node  $n$  is not a bottleneck for flow  $f(p, q)$  which contradicts the assumption and, hence,  $\sum_i r_i^n = C$ .

The proof of sufficiency is straightforward. As  $\sum_i r_i^n = C$ , it follows from Equation (3.3.6) that  $F_n = \Phi(C, \check{\mathbf{R}}_n)$  and, therefore, there exists an IA-flow  $IA(p, n)$ , for which  $r_p^n = F_n$ . Consequently, node  $n$  will be the bottleneck node for at least one of the flows within IA-flow  $IA(p, n)$ .  $\square$

In the following, we provide two examples to illustrate how we can examine fairness of a set of allocated rates according to the RIAMM fairness reference model.

**Example 3.3.5.** *Consider the scenario depicted in Figure 3.7 in which flows  $f(1, 3)$  and  $f(1, 4)$  compete for bandwidth on link  $l(2, 3)$  with flow  $f(2, 4)$ . Assume that  $d_{14} = d_{13} = d_{24} = C$ . The goal is to check if  $\mathbf{R} = \{r_{13} = 0.4 \cdot C, r_{14} = 0.4 \cdot C, r_{24} = 0.2 \cdot C\}$  is a RIAMM-fair solution. At node 2, we have  $\check{\mathbf{R}}_2 = \{r_1^2 = 0.8 \cdot C, r_2^2 = 0.2 \cdot C\}$ , and  $F_2 = \Phi(C, \check{\mathbf{R}}_2) = 0.8 \cdot C$  according to Equation (3.3.6). This means that node 2 is allowed to add traffic to the ring with the rate  $r_2^2 = r_{24} = \min\{d_{24} = C, F_2 = 0.8 \cdot C\} = 0.8 \cdot C$ . However, the rate of transit traffic at node 2 is  $r_1^2 = 0.8 \cdot C$  and, as a result, node 2 is unable to add traffic to the ring with the rate  $r_2^2 = 0.8 \cdot C$ . In fact,  $\check{\mathbf{R}}_2 = \{r_1^2 = 0.8 \cdot C, r_2^2 = 0.8 \cdot C\}$  is not a feasible solution and, consequently, the fair rate at node 2 is re-calculated as  $F_2 = \Phi(C, \{r_1^2 = 0.8 \cdot C, r_2^2 = 0.8 \cdot C\}) = 0.5 \cdot C$ . This means that node 1 is sending traffic more than the fair rate of node 2 and Equation (3.3.6) is not satisfied. Therefore,  $\mathbf{R}$  is not a RIAMM-fair solution.*

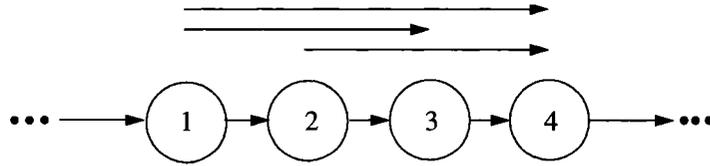


Figure 3.7: A multiple-destination scenario

Note that  $\{r_{14} + r_{13} = 0.5 \cdot C, r_{24} = 0.5 \cdot C\}$  is a RIAMM-fair solution in this scenario. Assuming that  $\eta = 1$ , from Equation (3.3.6) we have  $F_1 = C$ ,  $F_2 = 0.5 \cdot C$ ,  $F_3 = 0.75 \cdot C$ , and it can be verified that Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously at all nodes. Recall that the RIAMM fairness reference model does not depend on any particular source behavior and as long as  $r_{14} + r_{13} = r_{24} = 0.5 \cdot C$ , this allocation is considered a RIAMM-fair solution. Therefore, node 1 can distribute its fair share of bandwidth among its local flows (i.e.  $f(1,4)$  and  $f(1,3)$ ) in many different ways in order to achieve various objectives. Furthermore, one can note that for the RIAMM-fair solution  $r_{14} = r_{13} = 0.25 \cdot C$  and  $r_{24} = 0.5 \cdot C$ , node 2 is a bottleneck for all flows and we have  $r_1^2 + r_2^2 = C$ . ◀

**Example 3.3.6.** Consider the scenario discussed in Example 3.3.5 and assume that  $d_{24} = 0.2 \cdot C$ . In this case,  $\mathbf{R} = \{r_{13} = 0.4 \cdot C, r_{14} = 0.4 \cdot C, r_{24} = 0.2 \cdot C\}$  is a RIAMM-fair solution. The reason is that we have  $F_2 = 0.8 \cdot C$  and  $r_2^2 = r_{24} = \min\{d_{24} = 0.2 \cdot C, F_2 = 0.8 \cdot C\} = 0.2 \cdot C$  and, hence, Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously at node 2. It can be also verified that for any  $\eta$ ,  $0 < \eta \leq 1$ , the fairness conditions are maintained at nodes 1 and 3 as well. ◀

Note that in order to calculate the fair rate of node  $n$ , the rate of all IA-flows  $IA(i,n), i \leq n$ , should be calculated. The rate of the IA-flow  $IA(i,n)$ ,  $i < n$ , is simply the rate of traffic originated from node  $i$  transiting node  $n$ . However, the rate of the IA-flow  $IA(n,n)$  may be more than the service rate of local traffic of node  $n$ . (This can be observed in Example 3.3.5, where  $r_2^2 = 0.8 \cdot C$  while the service rate

of traffic of node 2 was  $0.2 \cdot C$ .) This usually happens when the fair rates are not properly adjusted and the transit traffic of node  $n$  blocks the local traffic of that node. Therefore, the rate of  $IA(n, n)$  should be calculated as the total allocated rate of local traffic of node  $n$  even though it is not fully serviced (i.e. partially blocked by the transit traffic of node  $n$ ). In general, we have  $r_n^n = \sum_{j>n} r_{nj}$ , and the allocated rate of each flow  $f(n, j)$ ,  $j > n$ , is obtained as follows:

$$r_{nj} = \min \{d_{nj}, \Theta_{nj}\}, \quad (3.3.7)$$

where  $\Theta_{nj}$  represents the rate limiter value of  $f(n, j)$  which is set according to the source behavior at node  $n$  and the advertised fair rates of downstream nodes of node  $n$ .

In single-destination scenarios discussed in Section 3.3.1, each node sends traffic to a single destination. As a result, each node fully allocates its share of bandwidth to its single local flow. Further, in a single-destination scenario, the allocated rate of each flow is limited to the minimum of the fair rates of the nodes along its path to the destination. Hence, for flow  $f(i, j)$  we have  $\Theta_{ij} = \min_{i \leq n < j} \{F_n\}$ . However, in multiple-destination scenarios,  $\Theta_{ij}$  depends on the advertised fair rates of downstream nodes as well-as the source behavior of node  $i$ .

Before comparing the RIAMM and RIAS fairness reference models, we study a number of source behaviors, which may be employed at an RPR node.

### 3.4 RPR Source Behaviors

The RPR standard does not provide any guidelines for the buffer management at each node. It only requires the local traffic at each node to be differentiated according to class of service; the number of queues and queue managers are a matter of choice. In addition, nodes may employ different source behaviors to achieve different objectives.

**Definition 3.4.1.** A source behavior at node  $i$  resulting in a set of allocated rates  $\mathbf{R} = \{r_{ij}, \forall j > i\}$  is called *feasible* if the following conditions are satisfied:

$$\sum_{j>i} r_{ij} \leq C, \quad (3.4.1)$$

$$\sum_{j>n} r_{ij} \leq F_n, \forall n \geq i, \quad (3.4.2)$$

where  $C$  is the capacity of each link on the ring.

A source behavior employed at an RPR node must be feasible in order to comply with the RPR standard requirements. In the following, three feasible source behaviors are introduced and compared in terms of the achieved throughput.

### 3.4.1 Simple Single-Rate (SSR)

Consider the simplest structure for the local queues of an RPR node in which there is only one queue for each traffic class. As a result, the rate of local traffic of a node to different destinations cannot be controlled separately. Therefore, the *aggregate* rate of FE local traffic added by each node should be limited to the fair rate of the most congested node of the congestion span which has the minimum fair rate. Assume that node  $m$  is the most congested node in a congestion span and its fair rate is equal to  $F_m$ . Any node  $i \leq m$  in this congestion domain should limit the aggregate rate of its FE local traffic to  $F_m$ . That is,

$$\sum_{j>i} r_{ij} \leq F_m. \quad (3.4.3)$$

In general, the SSR source behavior is not efficient in terms of bandwidth utilization throughout the ring as nodes are always limited to the fair rate of the most congested node while their local traffic may not transit that congested node (see Example 3.4.1). However, the RPR standard allows nodes to use such a source behavior and, as a

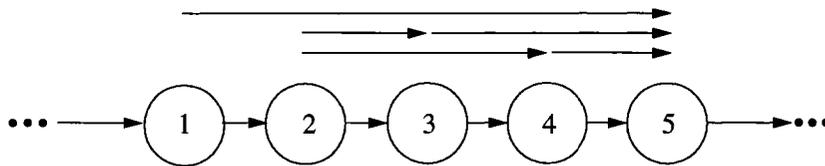


Figure 3.8: An example for the SSR source behavior

result, some nodes may choose to implement this simple source behavior instead of more complex ones.

**Example 3.4.1.** Consider the scenario depicted in Figure 3.8, in which nodes 1, 3, and 4 send traffic to node 5 and node 2 has two flows destined to nodes 3 and 4. Assume that all flows are infinite-demand and the SSR source behavior is employed at all nodes. In this congestion span, we have  $F_4 = \frac{C}{3}$  and, hence  $r_{15} = r_{35} = r_{45} = \frac{C}{3}$ . However, node 2 is unable to adjust the rate of its local traffic to each destination separately as it employs the SSR source behavior. Therefore, the aggregate rate of its local traffic added to the ring is limited to  $F_4 = \frac{C}{3}$  and, as a result,  $r_{23} = r_{24} = \frac{C}{6}$ . It can be verified that for  $\mathbf{R} = \{r_{15} = r_{35} = r_{45} = \frac{C}{3}, r_{23} = r_{24} = \frac{C}{6}\}$  and for any  $\eta$ ,  $0 < \eta \leq 1$ , the fairness criteria in Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously. Hence,  $\mathbf{R}$  is a RIAMM-fair solution when the SSR source behavior is employed. ◀

Note that in Example 3.4.1, the allocated rate of each local flow of node 2 could be equal to  $\frac{C}{3}$ , if node 2 had the capability of controlling the rate of each flow separately. In other words, using the SSR source behavior in this scenario results in a RIAMM-fair solution which is not efficient in terms of bandwidth utilization.

### 3.4.2 Equal Partitioning (EP)

In this source behavior, which is introduced in [6], it is assumed that each node supports the virtual destination queuing, i.e. local flows destined to different nodes

are buffered in separate queues. As a result, each node can regulate the rate of each flow separately. When this source behavior is used at node  $i$ , the rate limiter value of flow  $f(i, j)$ , i.e.  $\Theta_{ij}$ , is set as

$$\Theta_{ij} = \min_{i \leq n < j} \left\{ \frac{F_n}{\Lambda_i^n} \right\}, \quad (3.4.4)$$

where  $\Lambda_i^n$  denotes the number of flows originating from node  $i$  and transiting node  $n$ . In fact,  $\Lambda_i^n$  is the number of flows within IA-flow  $IA(i, n)$ . According to this source behavior, the bandwidth share of node  $i$  at each downstream node  $n$  is *equally* divided among the local flows of node  $i$  transiting node  $n$ , hence the name ‘equal partitioning’. The EP source behavior is particularly attractive for its remarkably simple implementation while exhibiting high performance.

**Example 3.4.2.** Consider the scenario depicted in Figure 3.9 and assume that the EP source behavior is employed at each node. Node 1 has two flows destined to node 3 and node 4, respectively, and node 2 has a flow destined to node 3. Assume that  $d_{14} = 0.1 \cdot C$  and  $d_{13} = d_{23} = C$ . Consider the set of allocated rates  $\mathbf{R} = \{r_{13} = 0.3 \cdot C, r_{14} = 0.1 \cdot C, r_{23} = 0.6 \cdot C\}$ , which is a feasible solution for this scenario. From Equation (3.3.6) and for  $\eta = 1$  we have  $F_1 = C$ ,  $F_2 = 0.6 \cdot C$ , and  $F_3 = C$ . Based on these fair rates, the rate limiter of each flow is calculated from Equation (3.4.4) as  $\Theta_{13} = \min\{\frac{F_1}{2}, \frac{F_2}{2}\} = 0.3 \cdot C$ ,  $\Theta_{14} = \min\{\frac{F_1}{2}, \frac{F_2}{2}, \frac{F_3}{1}\} = 0.3 \cdot C$ , and  $\Theta_{23} = \min\{\frac{F_2}{1}\} = 0.6 \cdot C$ . The allocated rate of each flow is then obtained from Equation (3.3.7) as  $r_{13} = 0.3 \cdot C$ ,  $r_{14} = 0.1 \cdot C$ , and  $r_{23} = 0.6 \cdot C$ , which are equal to the allocated rates in  $\mathbf{R}$ . Hence, the fairness criteria in Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously which means that  $\mathbf{R}$  is a RIAMM-fair solution in this scenario. Note that node 2 is allocated more bandwidth on link  $l(2, 3)$  when compared to node 1. The reason is that node 1 equally divides its share of bandwidth among its local flows regardless of their demands. However, flow  $f(1, 4)$  cannot fully utilize its share of bandwidth as  $d_{14} < \Theta_{14}$ . Consequently, node 2 reclaims

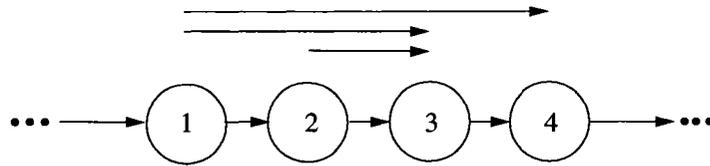


Figure 3.9: An example for the EP source behavior

the unused bandwidth of node 1.

Now, assume that the demand of flow  $f(1, 4)$  is increased to  $d_{14} = C$ . As the rate limiter of flow  $f(1, 4)$  is set to  $\Theta_{14} = 0.3 \cdot C$ , we have  $r_{14} = 0.3 \cdot C$ . Hence, the rate of IA-flows at node 2 will be  $r_1^2 = 0.6 \cdot C$  and  $r_2^2 = 0.6 \cdot C$ , which is not a feasible solution. Consequently, the fair rate of node 2 is calculated as  $F_2 = 0.5 \cdot C$  according to Equation (3.3.6). Based on the new fair rate of node 2, the rate limiters of flows  $f(1, 3)$  and  $f(1, 4)$  are set as  $\Theta_{13} = \Theta_{14} = 0.25 \cdot C$ . As a result, the RIAMM-fair solution is  $\mathbf{R} = \{r_{13} = r_{14} = 0.25 \cdot C, r_{23} = C\}$ . In this case, the allocated rates of node 1 and node 2 on link  $l(2, 3)$  are equal. ◀

Note that when the EP source behavior is used, each node simply divides its fair share of bandwidth at a given node among its local flows passing through that node. This is performed regardless of demands of local flows. As a result, when a flow does not fully utilize its share of bandwidth, the unused bandwidth cannot be reclaimed by other flows originating from the same node. In this case, other nodes will have a chance to reclaim the unused bandwidth.

### 3.4.3 Max-Min Partitioning (MMP)

In this source behavior, each node allocates its share of bandwidth to its local flows in a max-min fashion. In other words, the rate of each local flow is regulated such that max-min fairness is maintained among local flows. As the rate of each local flow is regulated separately, each node should support the virtual destination queuing.

Assume that the MMP source behavior is implemented at node  $i$  and  $F_n$ ,  $n \geq i$ , is the fair rate of downstream node  $n$ . In this case, the rate of IA-flow  $IA(i, n)$ ,  $r_i^n$ , should not exceed the fair rate of node  $n$ . In other words, the available capacity (service rate) for traffic of node  $i$  at node  $n$  is equal to  $F_n$ . Node  $i$  distributes its fair share of bandwidth at node  $n$  among its local flows, which transit node  $n$ , in a max-min fashion. Assume that  $L_n$  is the *max-min* fair rate (defined in Section 3.3.1) for the set of local flows  $\{r_{im}, m > n\}$  of node  $i$  served at node  $n$  with service rate  $F_n$ . Therefore, based on the discussions in Section 3.3.1, we have

$$L_n = \Phi^+(F_n, \{r_{im}, m > n\}, \eta), \forall n \geq i, \quad (3.4.5)$$

where  $0 < \eta \leq 1$ . As each flow  $f(i, j)$  passes through multiple nodes along its path from node  $i$  to node  $j$ , there is a max-min fair rate associated with this flow at each node  $n$ ,  $i \leq n < j$ . Therefore, the rate limiter of flow  $f(i, j)$  should be set to the minimum of the fair rates of this flow at each downstream node  $n$ ,  $i \leq n < j$ . Furthermore, the allocated rate of each flow  $f(i, j)$  is upper-bounded by its demand, i.e.  $d_{ij}$ . Hence, the allocated rate of flow  $f(i, j)$  is obtained as

$$r_{ij} = \min \left\{ d_{ij}, \min_{i \leq n < j} \{L_n\} \right\}, \forall j > i. \quad (3.4.6)$$

In fact, when the MMP source behavior is used at node  $i$ , Equations (3.4.5) and (3.4.6) are maintained simultaneously for the allocated rates of all local flows at node  $i$ . Note that calculating the allocated rate of each local flow with the MMP source behavior is much more complex than the one with SSR and EP.

In the following, we describe one of the properties associated with the MMP source behavior.

**Definition 3.4.2.** A flow  $f(i, j)$  is called *backlogged*, when its allocated rate is less than its offered (demand) rate, that is,  $r_{ij} < d_{ij}$ .

**Proposition 3.4.1.** *For a RIAMM-fair solution  $\mathbf{R} = \{r_{ij}\}$  and with the MMP source behavior at all nodes, every backlogged flow  $f(i, j)$  has a bottleneck node.*

*Proof.* Assume that flow  $f(i, j)$  with allocated rate  $r_{ij}$  and demand  $d_{ij}$  is backlogged. As a result,  $r_{ij} < d_{ij}$  and it follows from Equation (3.4.6) that  $r_{ij} = \min_{i \leq n < j} \{L_n\}$ . Assume that the minimum fair rate for flow  $f(i, j)$  is the one calculated at node  $m$  ( $i \leq m < j$ ), i.e.  $r_{ij} = L_m$ . Consequently, we have  $r_{ij} \geq r_{ij'}, \forall j' > m$ , as the rate of each flow originating from node  $i$  and transiting node  $m$  cannot be more than  $L_m$ . In order to prove that node  $m$  is a bottleneck node for flow  $f(i, j)$ , we need to show that  $r_i^m = F_m$ . As  $\mathbf{R}$  is a RIAMM-fair solution, we have  $r_i^m \leq F_m$ . Assume that  $r_i^m < F_m$ . As  $r_i^m = \sum_{j' > m} r_{ij'} < F_m$ , it follows from Equation (3.4.6) that  $L_m > r_{ij'}, \forall j' > m$ . This means that  $L_m > r_{ij}$  which contradicts the above assumption for flow  $f(i, j)$  and, hence,  $r_i^m = F_m$ . As  $r_{ij} \geq r_{ij'}, j' > m$ , and  $r_i^m = F_m$ , node  $m$  is a bottleneck for flow  $f(i, j)$ .  $\square$

According to Proposition 3.4.1, when the MMP source behavior is used, there is a bottleneck node for each backlogged flow. This ensures that the utilized bandwidth is maximized while max-min fairness is maintained among local flows at each node. In other words, the MMP source behavior outperforms the other source behaviors in terms of the bandwidth utilization.

**Example 3.4.3.** *Consider the scenario discussed in Example 3.4.2 and assume that the MMP source behavior is used at each node. For  $d_{14} = 0.1 \cdot C$ , we show that  $\mathbf{R} = \{r_{13} = 0.4 \cdot C, r_{14} = 0.1 \cdot C, r_{23} = 0.5 \cdot C\}$  is the fair solution with the MMP source behavior. For  $\eta = 1$ , we have  $F_1 = C$ ,  $F_2 = 0.5 \cdot C$ , and  $F_3 = C$ . It then follows from Equation (3.4.5) that  $L_1 = 0.9 \cdot C$ ,  $L_2 = 0.4 \cdot C$ , and  $L_3 = C$ . Consequently, it can be easily verified that Equation (3.4.6) is satisfied. Similarly, we can show that Equations (3.4.5) and (3.4.6) are also maintained for the flow of node 2. Moreover,*

*Inequality (3.3.5) and Equation (3.3.6) are satisfied and, therefore,  $\mathbf{R}$  is a RIAMM-fair solution with the MMP source behavior. Note that in this scenario, node 2 is a bottleneck for flows  $f(1, 3)$  and  $f(2, 3)$ .*

*For the second case, where  $d_{14} = C$ , it can be shown that  $\mathbf{R} = \{r_{13} = 0.25 \cdot C, r_{14} = 0.25 \cdot C, r_{23} = 0.5 \cdot C\}$  is the fair solution with the MMP source behavior which is same as the one with the EP source behavior.*

### 3.4.4 Performance Comparison of SSR, EP, and MMP Source Behaviors

In this section, we study the impact of using SSR and EP source behaviors on the achieved throughput in a number of scenarios and compare their steady-state performance with the MMP source behavior. In each scenario, we assume that fairness is achieved and  $\mathbf{R} = \{r_{ij}\}$  is the RIAMM-fair solution. The performance metric is throughput denoted by  $\rho$  and defined as the aggregate allocated rates of all flows throughout the ring. That is,

$$\rho = \sum_{\forall i,j} r_{ij}. \quad (3.4.7)$$

In the following, we compare the performance of MMP, EP, and SSR source behaviors in three different scenarios.

#### Parallel Parking-Lot Scenario

In this scenario, nodes  $1, 2, \dots, K$  send traffic to node  $K + 1$ . Further, node 1 sends traffic to node  $H + 1$  ( $1 \leq H < K$ ) which is  $H$  hops away. It is assumed that all flows in this scenario are infinite-demand. The parallel parking-lot scenario is also one of the benchmark scenarios which is considered in literature for performance evaluation

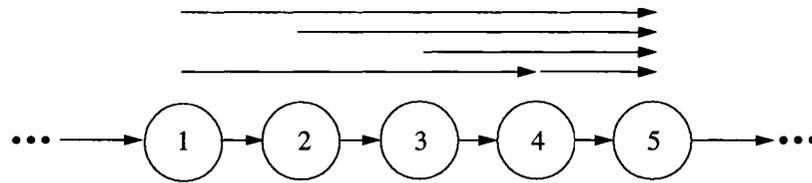


Figure 3.10: Parallel parking-lot scenario for  $K = 4$  and  $H = 3$

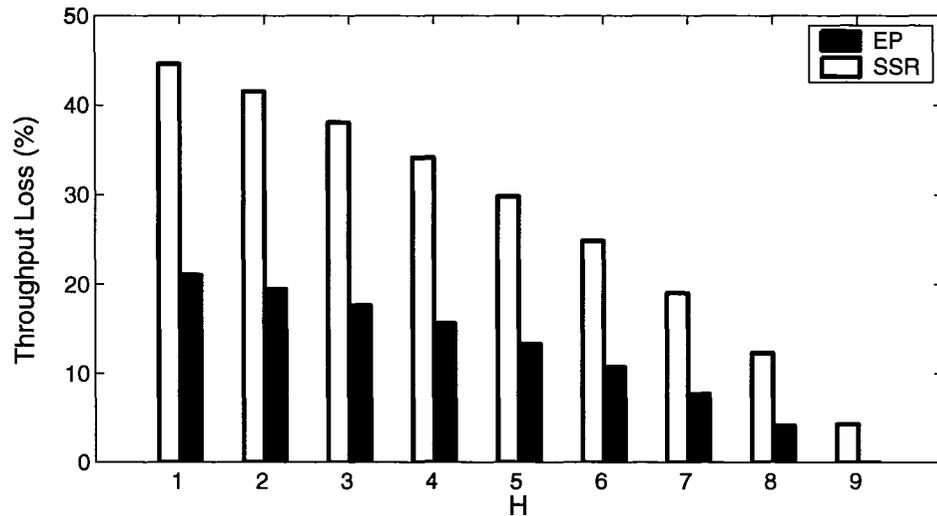


Figure 3.11: Throughput loss versus  $H$  for the SSR and EP source behaviors in a parallel parking-lot scenario with  $K = 10$

of packet rings [6]. In Figure 3.10, a parallel parking-lot scenario with  $K = 4$  and  $H = 3$  is presented.

Figure 3.11 presents the relative throughput loss for SSR and EP compared to the achieved throughput with the MMP source behavior.<sup>6</sup> In this scenario, it is assumed that  $K = 10$  and the throughput loss is obtained for various  $H$ . One can first notice that the throughput loss of the SSR source behavior is much larger when compared to the one for EP. The reason is that when the SSR source behavior is used, node 1 cannot control the rate of its local flows separately and, hence, cannot

<sup>6</sup>For example, the relative throughput loss of the SSR source behavior is obtained as  $100 \cdot (1 - \rho_{\text{SSR}}/\rho_{\text{MMP}})$ .

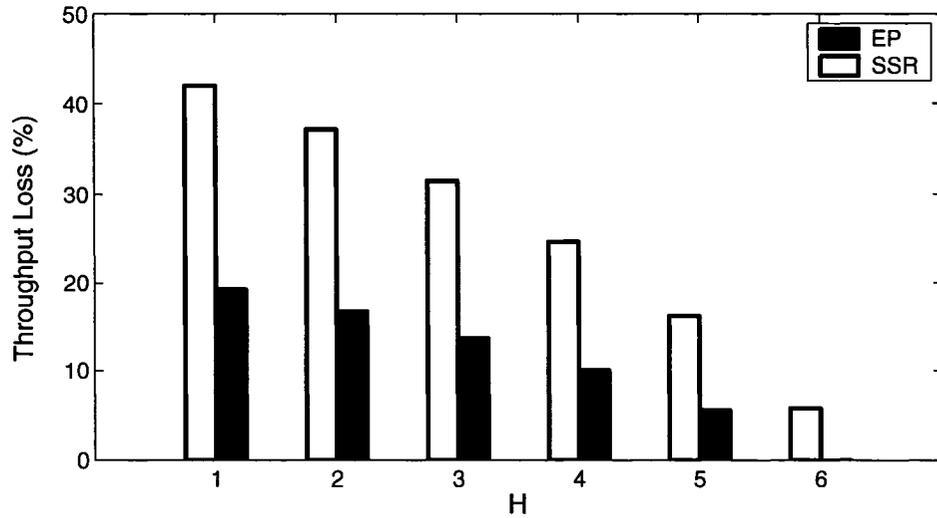


Figure 3.12: Throughput loss versus  $H$  for the SSR and EP source behaviors in a parallel parking-lot scenario with  $K = 7$

efficiently allocate bandwidth to its local flows. On the other hand, when the EP source behavior is employed, node 1 controls the rate of its local flows separately and, as a result, the throughput loss is lower when compared to the one with SSR.

In general, when the difference between the available bandwidth for the short-flow and the long-flow of node 1 increases, the throughput loss with EP and SSR source behaviors increases. The reason is that the bandwidth is not efficiently allocated to the short-flows when EP and SSR source behaviors are employed. It can be seen in Figure 3.11 that the throughput loss of EP and SSR source behaviors decreases with  $H$ . The reason is that, as  $H$  increases, the available bandwidth for the short-flow of node 1 decreases. This means that the difference between the available bandwidth of the local flows of node 1 decreases with  $H$ . Therefore, the throughput loss of SSR and EP decreases as  $H$  increases.

In Figure 3.12, the relative throughput loss versus  $H$  is shown for EP and SSR source behaviors in a parallel parking-lot scenario with  $K = 7$ . One can see that for a constant  $H$ , the throughput loss for SSR and EP source behaviors for  $K = 7$

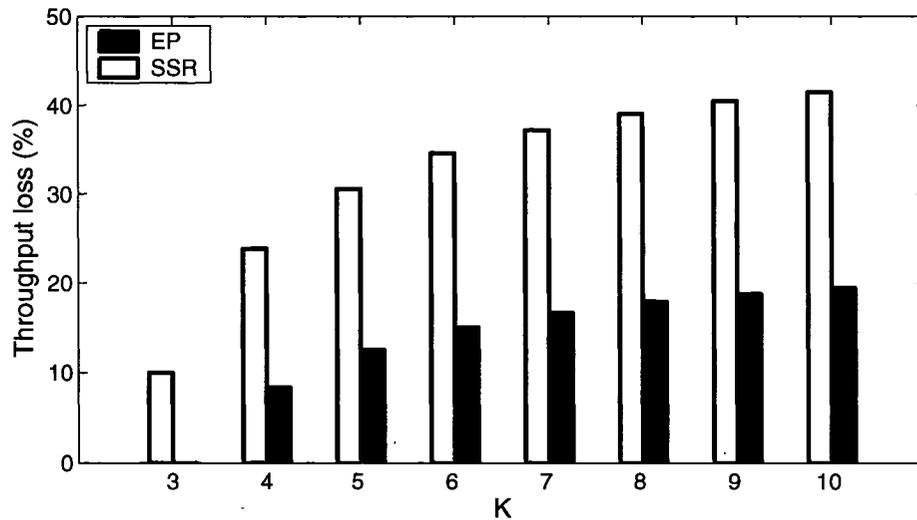


Figure 3.13: Throughput loss versus  $K$  for the SSR and EP source behaviors in a parallel parking-lot scenario with  $H = 2$

is slightly less than the one for  $K = 10$  (Figure 3.11). The reason is that, in this scenario, the short-flow of node 1 always competes for the bandwidth with  $H$  flows regardless of the number of nodes  $K$ . On the other hand, the allocated rate of each of the flows destined to node  $K + 1$  decreases with  $K$ . This means that for a fixed  $H$ , the available bandwidth for the short-flow of node 1 increases with  $K$ . In other words, the difference between the available bandwidth for the short flow and the long flow of node 1 for a given  $H$  increases with  $K$ . As a result, the throughput loss grows with  $K$  for a given  $H$ . This can be seen in Figure 3.13 where the throughput loss of the SSR and EP source behaviors increases with  $K$  for  $H = 2$ .

### Mixed Parking-Lot Scenario

In this scenario, nodes 1, 2, ...,  $K$  send traffic to node  $K + 1$ . Moreover, each node  $i < K - H$  ( $1 \leq H \leq K$ ) has a flow destined to node  $i + H$ . It is also assumed that all flows in this scenario are infinite-demand. We consider this scenario to study the effect of EP and SSR source behavior on the throughput loss when all nodes send

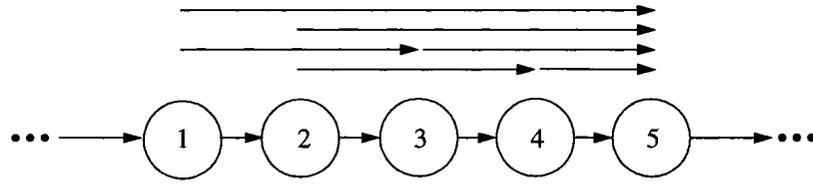


Figure 3.14: Mixed parking-lot scenario for  $K = 4$  and  $H = 2$

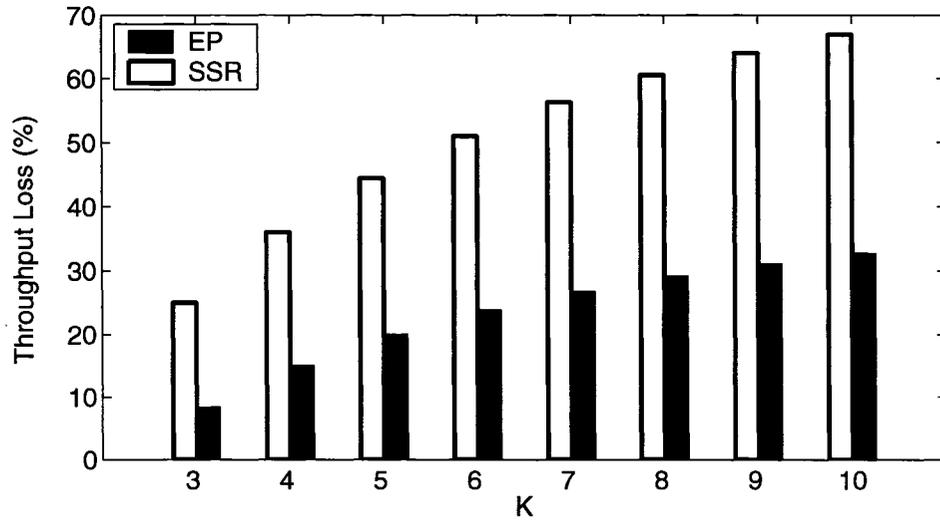


Figure 3.15: Throughput loss versus  $K$  for the SSR and EP source behaviors in a mixed parking-lot scenario with  $H = 1$

traffic to multiple destinations. Figure 3.14 presents a mixed parking-lot scenario with  $K = 4$  and  $H = 2$ .

As mentioned before, when the difference between the available bandwidth for the short-flow and for the long-flow of each node increases, the throughput loss of EP and SSR source behaviors increases. Figure 3.15 presents the relative throughput loss versus  $K$  in the mixed parking-lot scenario with  $H = 1$  for EP and SSR source behaviors. One can see that as  $K$  increases, the throughput loss grows for both source behaviors. The reason is that as  $K$  increases the available bandwidth for the short flow of each node increases while the available bandwidth for the long flow decreases. In other words, the disparity between the allocated rate of the flows of

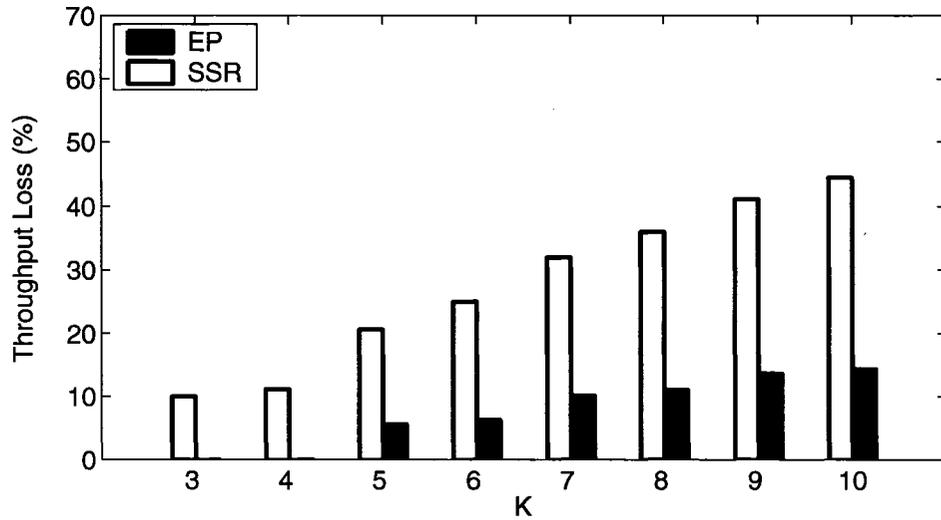


Figure 3.16: Throughput loss versus  $K$  for the SSR and EP source behaviors in a mixed parking-lot scenario with  $H = 2$

each node increases with  $K$  and, as a result, the throughput loss of EP and SSR source behaviors increases.

In Figure 3.16, the relative throughput loss versus  $K$  is depicted for EP and SSR source behaviors in a mixed parking-lot scenario with  $H = 2$ . The comparison between Figure 3.15 and Figure 3.16 shows that for a fixed  $K$ , the throughput loss with  $H = 1$  is greater than the one with  $H = 2$  for both source behaviors. The reason is that the difference between the available bandwidth for the short flow and for the long flow of each node in the scenario with  $H = 1$  is larger than the one with  $H = 2$ . Note that in Figure 3.16 the throughput loss with EP is less than 14% in all cases. This means that the achieved throughput with EP is not less than 86% of the one with MMP. (Note that for  $K = 3$  and  $K = 4$  the throughput loss is zero when EP is employed which means that the achieved throughput with EP is same as the one with MMP in these cases.) Therefore, the EP source behavior may be preferred over MMP as its implementation complexity is much less.

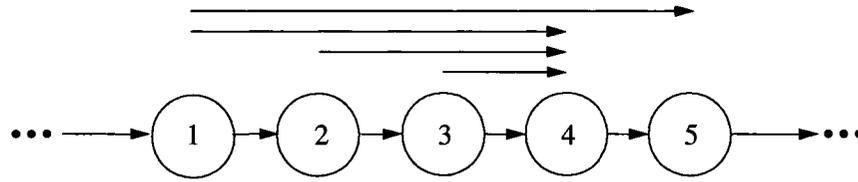


Figure 3.17: Two-exit parking-lot scenario for  $K = 3$

### Two-Exit Parking-Lot Scenario

Recall that with the EP source behavior, each node equally distributes its share of bandwidth among its local flows regardless of their demands. As it was shown in Example 3.4.2, when demands of the local flows of a node are different, the unused bandwidth of that node may be reclaimed by other nodes. To study this issue, we consider the two-exit parking-lot scenario which is one of the benchmark scenarios used in literature for performance evaluation of the packet rings [6]. In this scenario, nodes  $1, 2, \dots, K$  send traffic to node  $K + 1$ . Moreover, node 1 has a flow destined to node  $K + 2$ . Figure 3.17 presents a two-exit parking-lot scenario with  $K = 3$ . We assume that the demand of flow  $f(1, K + 2)$  is equal to  $d$  and all other flows are infinite-demand.

When the MMP source behavior is used, the bandwidth share of each node is equal to  $\frac{C}{K}$ . In this case, node 1 divides its share of bandwidth among its local flows in a max-min sense and, consequently, regardless of the demand of flow  $f(1, K + 2)$ , the total bandwidth consumed by node 1 will be equal to  $\frac{C}{K}$ . However, when the EP source behavior is employed, node 1 equally divides its share of bandwidth among its local flows. If flow  $f(1, K + 2)$  with the limited demand  $d$  does not fully utilize its allocated bandwidth, the unused bandwidth will be reclaimed by the other nodes. As a result, the bandwidth share of other nodes will be larger than that of node 1. Figure 3.18 presents the relative difference between the throughput of node 1 and

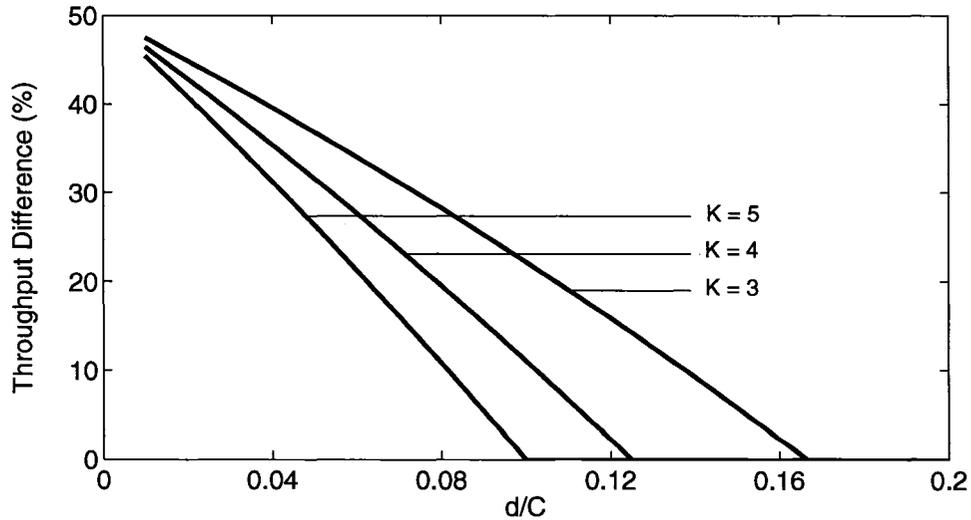


Figure 3.18: The relative throughput difference between node 1 and other nodes versus  $d$  in a two-exit parking-lot scenario for  $K = 3, 4, 5$

that of node  $i$ ,  $1 < i \leq K$ , versus  $d$  when the EP source behavior used in a two-exit parking lot scenario for  $K = 3, 4, 5$ . One can see that as the demand of flow  $f(1, K + 2)$  increases, the difference between the allocated of node 1 and the other nodes decreases. The reason is that when  $d$  is large, the unused bandwidth of flow  $f(1, K + 2)$  decreases and, hence, the throughput difference becomes smaller. In general, the throughput difference between node 1 and the other nodes is zero in this scenario when  $d \geq 0.5 \cdot \frac{C}{K}$ .

### 3.5 Comparison of RIAMM and RIAS Fairness Reference Models

In this section, we compare the RIAS and RIAMM fairness reference models and discuss their relation. We first state the following proposition without proof, which describes one of the properties of the RIAS fairness model.

**Proposition 3.5.1** (Proposition 1 in [6]). *A feasible set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is RIAS-fair if and only if each backlogged flow  $f(i, j)$  has a bottleneck node with respect to  $\mathbf{R}$ .*

Note that in the RIAS-fair solution discussed in Example 3.1.4, node 1 is a bottleneck node for flow  $f(1, 2)$ ; node 2 is the bottleneck of flows  $f(1, 3)$ ,  $f(1, 4)$ , and  $f(1, 5)$ ; and node 4 is a bottleneck node for flows  $f(2, 5)$  and  $f(4, 5)$ .

In the following propositions, we first prove that a RIAS-fair rate allocation is also RIAMM-fair. We then show that with the MMP source behavior employed at all nodes, a RIAMM-fair allocation is also a RIAS-fair solution.

**Proposition 3.5.2.** *A RIAS-fair set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  is also a RIAMM-fair solution.*

*Proof.* As  $\mathbf{R} = \{r_{ij}\}$  is a RIAS-fair solution, according to Proposition 3.5.1, every backlogged flow has a bottleneck node. The rate of a backlogged flow cannot be increased while maintaining feasibility as it violates the RIAS fairness conditions. One of the following cases occurs for an arbitrary node  $n$ :

- If node  $n$  is a bottleneck node for at least one flow, we have  $\sum_j r_j^n = C$ . It then follows from Equation (3.3.6) that  $F_n = \Phi(C, \check{\mathbf{R}}_n)$  and, consequently,  $r_i^n \leq F_n$ . The equality is maintained when node  $n$  is a bottleneck for at least one of the flows within  $IA(i, n)$ . When  $r_i^n < F_n$ , node  $n$  is not a bottleneck node for any flow within  $IA(i, n)$ . In this case, the rate of  $IA(i, n)$  cannot be increased although it is less than the fair rate of node  $n$  (i.e.  $r_i^n < F_n$ ). The reason is that in order to increase the rate of  $IA(i, n)$ , the rate of the backlogged flows within  $IA(i, n)$  should be increased which violates the RIAS fairness conditions. Hence, the Inequality (3.3.5) and Equation (3.3.6) are simultaneously satisfied at an arbitrary node  $n$  which is a bottleneck for at least one backlogged flow.

- If node  $n$  is not a bottleneck for any flow, we have  $\sum_j r_j^n < C$  and according to Equation (3.3.6),  $F_n = \Phi^+(C, \check{\mathbf{R}}_n, \eta) > r_i^n$  for any  $\eta$ ,  $0 < \eta \leq 1$ . Similar to the previous case,  $r_i^n$  cannot be increased as it violates the RIAS fairness conditions. In other words, Inequality (3.3.5) and Equation (3.3.6) are maintained simultaneously for an arbitrary node  $n$  which is not a bottleneck for any flow.

Therefore, given that the set of allocated rates  $\mathbf{R}$  is a RIAS-fair solution, the RIAMM fairness conditions in Inequality (3.3.5) and Equation (3.3.6) are satisfied simultaneously. Hence,  $\mathbf{R}$  is a RIAMM-fair solution.  $\square$

**Proposition 3.5.3.** *A RIAMM-fair set of allocated rates  $\mathbf{R} = \{r_{ij}\}$  with the MMP source behavior is a RIAS-fair solution.*

*Proof.* As  $\mathbf{R}$  is a RIAMM-fair solution with the MMP source behavior, according to Proposition 3.4.1 every backlogged flow has a bottleneck node. Therefore,  $\mathbf{R}$  is a RIAS-fair solution according to Proposition 3.5.1.  $\square$

It can be easily shown by counter examples that a RIAMM-fair rate allocations achieved by an arbitrary source behaviors is not necessarily a RIAS-fair solution. In the RIAS fairness model, it is assumed that the MMP source behavior is employed at all nodes. Therefore, when an arbitrary feasible source behavior is employed, RIAS fairness is not necessarily maintained. The RIAMM fairness model can be viewed as the generalized form of the RIAS fairness model which does not depend on any particular source behavior. In fact, the RIAMM fairness criteria can be met by any feasible source behavior. Nonetheless, in order to evaluate the fairness of a set of allocated rates, the source behavior should be known in the RIAMM fairness model.

Another advantage of the RIAMM fairness reference model is that, Equation (3.3.6) of the RIAMM fairness criteria suggests how fair rate should be calculated in order to achieve RIAMM fairness. However, in the RIAS fairness model, the fair rate is

not incorporated into the fairness criteria and, therefore, it is not clear how fair rates should be calculated to achieve RIAS fairness.

### 3.5.1 Equal Opportunity Source Behavior: A Case Study on Comparing RIAS and RIAMM Fairness Models

As mentioned earlier in Section 3.2.2, different source behaviors may be employed at each node in RPR to achieve different objectives. In the RIAS-fairness, however, it is assumed that the MMP source behavior is deployed at each node.

In Section 3.4 we studied SSR, EP, and MMP source behaviors. We argued that SSR and EP source behaviors may be preferred over the MMP source behavior because of their low implementation complexity. In this section, we study a scenario in which the aim is to provide equal transmission opportunity (throughput) for the local flows of each node when the network conditions permit. This source behavior is introduced in [11] as the Equal Opportunity (EO) source behavior. We show that the average throughput of local flows can not be equal in our scenario when RIAS fairness is considered. However, such a solution can exist under the RIAMM fairness conditions by employing the EO source behavior.

Consider the parallel parking-lot scenario (see Section 3.4.4) with  $K = 4$  and  $H = 1$  depicted in Figure 3.19, and assume that flows  $f(1, 2)$  and  $f(1, 5)$  are infinite-demand. We consider the following states for the network:

- **High-State:** In this state we have  $d_{25} = d_{35} = d_{45} = 0.25 \cdot C$  and, hence, the RIAS-fair solution for this state is  $\mathbf{R} = \{r_{15} = r_{25} = r_{35} = r_{45} = 0.25 \cdot C, r_{12} = 0.75 \cdot C\}$ .
- **Low-State:** In this state we have  $d_{25} = d_{35} = d_{45} = \frac{0.25}{3} \cdot C$  and, as a result, the RIAS-fair solution for this state is  $\mathbf{R} = \{r_{12} = r_{15} = 0.375 \cdot C, r_{25} = r_{35} =$

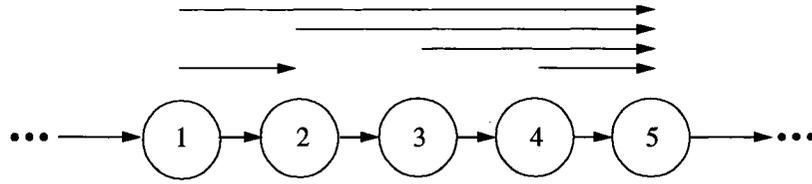


Figure 3.19: An example to illustrate the EO source behavior

$$r_{45} = \frac{0.25}{3} \cdot C\}.$$

Note that according to the RIAS fairness model, the throughput of the short-flow of node 1 (i.e.  $f(1, 2)$ ) is equal to or greater than that of its long-flow (i.e.  $f(1, 5)$ ) in the above cases although both flows have equal demands.

Now, consider a scenario where the network state is alternating between two states; for a period of  $t$  seconds the network is in the high-state and we have  $d_{25} = d_{35} = d_{45} = 0.25 \cdot C$ ; this follows with another period of  $t$  seconds where the network is in the low-state and we have  $d_{25} = d_{35} = d_{45} = \frac{0.25}{3} \cdot C$ . We assume that  $t$  is much larger than the convergence time of the network and, therefore, the network reaches its steady state shortly after switching to the new state. We define a *cycle* as a period of time which consists of a high-state followed by a low-state.

In this scenario, the average throughput of flow  $f(1, 2)$  over a cycle is  $0.5625 \cdot C$  when the RIAS-fair allocated rates are considered. Meanwhile, the average throughput of flow  $f(1, 5)$  is  $0.3125 \cdot C$  which is much less than that of flow  $f(1, 2)$ . Note that in the low-state, node 1 can allocate more bandwidth to its long-flow to increase the average throughput of this flow over a cycle. In other words, if the allocated rates of local flows at node 1 in the low-state were  $r_{15} = 0.75 \cdot C$  and  $r_{12} = 0.25 \cdot C$ , the average throughput of both flows over a cycle would be equal to  $0.5 \cdot C$ . Note that when the allocated rates of local flows at node 1 in the low-state are  $r_{15} = 0.75 \cdot C$  and  $r_{12} = 0.25 \cdot C$ , the fairness at the ring level is maintained as we still have  $r_{25} = r_{35} = r_{45} = \frac{0.25}{3} \cdot C$ . However, according to the RIAS fairness model, the rate of flow  $f(1, 5)$  cannot be more than the allocated rate of flow  $f(1, 2)$  when the

network is in the low-state in this scenario. This means that based on the RIAS fairness model, the average throughput of flow  $f(1,2)$  and flow  $f(1,5)$  cannot be equal. However, by employing the EO source behavior, a RIAMM-fair solution can be achieved in which node 1 provides equal throughput for its local flows.

The EO source behavior is intended to provide equal transmission opportunity for local flows of an RPR node in the framework of the RIAMM fairness model [11]. In this source behavior, each node keeps track of the allocated rates of its local flows. When network conditions permit, each node tries to allocate more bandwidth to those local flows which have received less bandwidth in the past due to congestion periods. Clearly, such a source behavior does not result in a RIAS-fair solution in general.

### 3.6 Conclusion

We proposed a new fairness reference model for RPR. The proposed model is a general model which does not depend on any particular source behavior. We showed that the RIAMM fairness model is a generalized form of another existing fairness model proposed for RPR. We also studied different source behaviors which can be employed at each RPR node and evaluated the performance of these source behaviors in terms of the ring throughput. We also reviewed the EO source behavior which aims at providing equal opportunity transmission for local flows in an RPR node. It was shown that this source behavior does not generally result in a RIAS-fair solution. However, based on the RIAMM fairness model, such a solution is considered fair as long as the ring level fairness is maintained.

## Chapter 4

# Analysis of RPR Standard Bandwidth Allocation Algorithms

In RPR, a bandwidth allocation algorithm is employed at each node in order to provide a fair ring access to all nodes and control the amount of traffic each node adds to the ring. The objective of the bandwidth allocation algorithm in RPR is to distribute fairly the available bandwidth on any link among the local traffic of all competing nodes on that link. We define the *true fair rate* as the rate at which every node should add its local traffic to the ring without starving its downstream nodes. (Note that  $F_n$ , which was defined in the previous chapter, is equal to the true fair rate of node  $n$  only if the allocated rates are RIAMM-fair.) When a node is starved and cannot add its fair share of traffic to the ring, it becomes congested. In this case, the congested node calculates a first approximation to the true fair rate. (Note that in case of congestion, some nodes are starved and cannot add traffic to the ring and, as a result, the exact value of the true fair rate is not usually known for the congested node.) The calculated fair rate is then advertised through a control message to upstream nodes, which contribute to congestion. As upstream nodes receive the control message, they adjust their add-rates according to the received fair rate. Notice that as the calculated fair rate is an approximation to the true fair rate, congestion may not be cleared at the congested node. In this case, the congested

node calculates another approximation of the true fair rate and advertises it to the upstream nodes. This process may repeat until the calculated fair rate converges to the true fair rate or, equivalently, source-based fairness is achieved among competing nodes. The time needed for the bandwidth allocation algorithm to converge to the true fair rate is commonly referred to as *convergence time*, which is an important criterion in evaluating performance and efficiency of an RPR bandwidth allocation algorithm.

The performance of the RPR standard bandwidth allocation algorithms (RPR-AM and RPR-CM) is investigated based on simulation results in [4–9]. The performance of the RPR standard bandwidth allocation algorithms can also be evaluated through analytical results by modeling the fair rate calculation process in these algorithms. Such analytical models can also provide insights on dynamics of these algorithms. However, an exact analysis of a complex dynamic system such as RPR seems extremely difficult. There have been a few analytical works on performance evaluation of RPR [46,67,68]. In [46], analytical results are presented to approximate the steady-state average packet transfer delay in RPR for two different structures of the transit path. In [67], an analytical bound for the ring access delay of nodes in the steady-state is developed. However, the effect of the bandwidth allocation algorithm has not been studied in these works as only the steady-state conditions were investigated. In [68], an analytical bound for stability and convergence of RPR-AM for a dual-queue transit path is derived. In this analysis, however, the effect of the advertised fair rate of the congested node on the add-rate of the upstream nodes has not been considered.

In this chapter, we propose analytical models for fair rate calculations in RPR-AM and RPR-CM. That is, we derive the fair rate equations in each algorithm as a function of system parameters. We then study the impact of different system parameters on the performance of RPR-AM and RPR-CM in terms of convergence and stability of the fair rate through analytical results. The difference between our

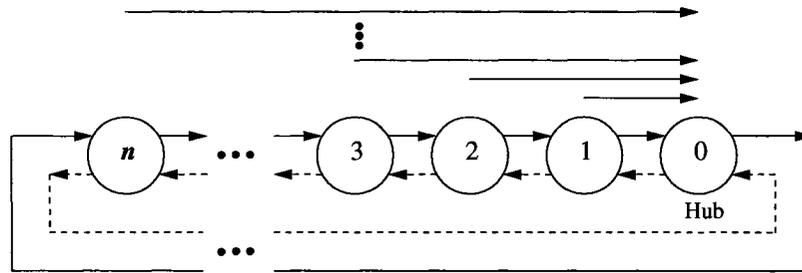


Figure 4.1: Parking-lot scenario

work and what is done in [68] is that in this work, we consider the effect of the feedback between fair rate advertisement and fair rate calculation in RPR in details. Further, we provide a closed form for the calculated fair rate of the congested node as a function of system parameters for RPR-AM and RPR-CM. Moreover, this work is the first analytical work on RPR-CM as far as we know.

The analytical models are developed for the parking-lot scenario depicted in Figure 4.1 in which all nodes  $1, 2, \dots,$  and  $n$  send traffic to node 0 (hub).<sup>1</sup> This scenario is the most popular benchmark scenario in RPR and has been widely studied in the standardization process of RPR. The parking-lot scenario is a traditional and challenging scenario which is considered in [4–9, 68, 69] to compare the performance of different RPR bandwidth allocation algorithms. One of the main reasons is that the parking-lot scenario represents a class of scenarios in which a number of nodes compete for bandwidth on a link. In fact, many other scenarios can be viewed as a parking-lot scenario (see Section 4.3). The performance of RPR bandwidth allocation algorithms can also be evaluated in arbitrary scenarios where each node may have traffic to send to multiple destinations. However, the performance evaluation of such scenarios through analytical results is extremely difficult due to the complexity of the RPR fairness control mechanism.

This chapter is organized as follows. We model the RPR-AM fair rate calculation

<sup>1</sup>Note that unlike the assumption in Chapter 4, nodes are numbered in a decreasing order with respect to the direction of traffic in Figure 4.1.

process in the parking-lot scenario in Section 4.1 and, then, study dynamics of RPR-AM through analytical results. In Section 4.2, we model the fair rate calculation process for RPR-CM in the parking-lot scenario. Finally, Section 4.4 concludes this chapter.

## 4.1 Analysis of RPR-AM Fair Rate Calculations in Parking-lot Scenario

In this section, we model the RPR-AM fair rate calculation process in the parking-lot scenario and, then, study the effect of different system parameters on the calculated fair rate through analytical results. We start by the problem description and, then, we formulate the problem in two cases (with and without the link propagation delay). Numerical results are also provided at the end of this section.

### 4.1.1 Problem Description

Consider the parking-lot scenario depicted in Figure 4.1. The dynamics of an RPR bandwidth allocation algorithm should be studied in an overloaded network; otherwise, congestion does not occur and there is no need for a rate control mechanism. In this analysis, we consider an extreme case where in Figure 4.1 nodes 1, 2,  $\dots$ , and  $n$  have theoretically infinite amount of traffic in their local buffers destined to node 0 (hub). In addition, we make the following assumptions:

- ▶ We only consider FE traffic as the bandwidth allocation algorithm only controls the rate of FE traffic throughout the ring. Therefore, the unreserved rate is equal to the link rate in our model.
- ▶ All nodes have equal weights. As all nodes compete for bandwidth on the link

between node 1 and node 0, this link becomes the most congested one. Hence, the fair share of each node is  $\frac{1}{n}$  of the link bandwidth.

- ▶ RPR-AM is employed at all nodes.
- ▶ The transit queue at each node is implemented in the single-queue mode. Note that in RPR, the transit queue mode is independent of the bandwidth allocation algorithm and only specifies how congestion is triggered [44].
- ▶ A node detects congestion only when the low-pass filtered version of the link usage at the output of its scheduler exceeds  $\tau_\ell$ . In fact, we ignore the access delay factor that may also trigger congestion (see Section 2.2.3).
- ▶ For simplicity, we assume that all rates are normalized to  $C$  (unreserved-rate) in this analysis (or we can assume that  $C = 1$ ).

In the following, we describe two methods for control message transmissions in RPR. In the first method, each RPR node checks its congestion status at the end of every control interval. In case of congestion, the congested node advertises its fair rate to its upstream node through a control message. If the upstream node is also congested, it advertises the minimum of its own fair rate and the latest fair rate received from its downstream node at the end of every control interval. If the upstream node is not congested, it forwards the fair rate of its downstream congested node at the end of every control interval. In this case, the fair rate of the most congested node (which is also called the head node of the congestion domain and has the minimum fair rate) propagates one hop in each control interval until it reaches the furthest node contributing to congestion [44].

In the second method, which is introduced in [6], there is a *single* control message circulating along the ring within each control interval. It contains separate fields for the calculated fair rate of all nodes. At the end of each control interval, one of the

nodes initiates transmission of the control message. This node inserts its advertised fair rate in a designated field in the control message and sends the control message to its upstream node. Upon receiving the control message, each node:

- sets its *allowedRateCongested* to the advertised fair rate of the head node of the congestion domain;
- updates the corresponding field in the control message with its most recently calculated fair rate and forwards the control message to its upstream node.

This method results in fewer control message transmissions and faster fair rate propagation throughout the ring compared to the first method. In this analysis, we consider the second scheme for transmitting the control message. In Figure 4.1, the solid-line ringlet is the traffic path to node 0. The control message is transmitted backward on the dashed-line ringlet.

In the following, we formulate this problem in two cases. First, we assume that the link propagation delay is zero and investigate this problem in an ideal case. Then, we incorporate a deterministic link propagation delay into the model and analyze the problem in a more realistic case.

#### 4.1.2 Analysis of Model without Link Propagation Delay

In this section, we assume that the link propagation delay is zero and model the fair rate calculations in RPR-AM. We then determine the conditions with respect to which the convergence time of RPR-AM is finite.

As we only consider FE traffic in this analysis,  $r_i^u(k)$  (i.e. the average aggregate rate of the unreserved traffic (all traffic except traffic of subclass A0), added and/or forwarded by node  $i$  in control interval  $k$ ) is equal to the average usage rate of the

output link of node  $i$  in control interval  $k$  and we have

$$r_i^u(k) = r_i^a(k) + r_i^f(k) \leq 1, \quad (4.1.1)$$

where  $r_i^a(k)$  and  $r_i^f(k)$  are the add-rate and forward-rate (see Section 2.2.4) of node  $i$  in control interval  $k$ , respectively .

As mentioned in Section 2.2.4, the add-rate of node  $i > 0$  in control interval  $k$ , i.e.  $r_i^a(k)$ , is limited to  $\ell_i(k)$  (*allowedRate*) and  $\ell_i^c(k)$  (*allowedRateCongested*). However, in RPR-AM, we have  $\ell_i(k) = 1, \forall k$ . In addition,  $r_i^a(k)$  is bounded by  $1 - r_i^f(k)$  as the transit buffer in all nodes is implemented in single-queue mode and the transit traffic has service priority over the local traffic at each node. Hence,

$$r_i^a(k) = \min \{1 - r_i^f(k), \ell_i(k) = 1, \ell_i^c(k)\} = \min \{1 - r_i^f(k), \ell_i^c(k)\}. \quad (4.1.2)$$

In the following, we first calculate the congestion detection time and, then, derive the fair rate equations. Finally, the stability condition of the fair rate is investigated.

### Congestion Detection Time

Starting at time 0 (or during control interval  $k = 1$ ), each node has traffic at the link rate to transmit to node 0. The initial value of *allowedRateCongested* of all nodes is equal to the link rate, i.e.  $\ell_i^c(1) = 1, i = 1, 2, \dots, n$ . As the transit traffic at each node has service priority over the local traffic of that node, only node  $n$  which is the most upstream node, can transmit at the link rate, i.e.  $r_n^a(k) = 1$ ; other nodes will not be able to add their local traffic to the ring before congestion detection and we have  $r_i^a(k) = 0$ , and  $r_i^f(k) = 1, \forall i < n$ . Note that as node  $n$  sends traffic at the link rate to node 0, before the congestion detection time we have  $r_i^u(k) = r_i^a(k) + r_i^f(k) = 1, i = 1, \dots, n$ .

Node  $i$  detects congestion when  $\tilde{r}_i^u(k)$  (i.e. low-pass filtered value of  $r_i^u(k)$ ) exceeds  $\tau_\ell$ . Note that as congestion is detected through a low-pass filtering approach, it is

not immediately detected when the link usage exceeds  $\tau_\ell$ . Assume that congestion is detected at the end of control interval  $\hat{k}$ . For node  $i > 0$  we have  $\tilde{r}_i^u(0) = 0$  and  $r_i^u(k) = 1$ , for all  $k \leq \hat{k}$ . Therefore, using Equation (2.2.2), one can derive  $\tilde{r}_i^u(k)$  as

$$\tilde{r}_i^u(k) = 1 - (1 - \alpha)^k, k \leq \hat{k}. \quad (4.1.3)$$

Consequently, using Equation (4.1.3), the congestion detection time for node  $i > 0$ , i.e.  $\hat{k} = \min \{k : \tilde{r}_i^u(k) > \tau_\ell\}$ , can be obtained as follows:

$$\hat{k} = \left\lfloor \frac{\ln(1 - \tau_\ell)}{\ln(1 - \alpha)} + 1 \right\rfloor, \quad (4.1.4)$$

where  $\lfloor \cdot \rfloor$  denotes the floor function. Note that all nodes 1, 2, ..., and  $n$ , detect congestion at the same time.

### Advertised Fair Rate

Assume that node 1 initiates the control message transmission at the end of each control interval. As the size of the control message is relatively short, we can ignore the transmission and the processing time of the control message throughout the ring. Moreover, we have assumed that the link propagation delay is zero in this ideal model. As a result, the time needed for a control message to reach the furthest upstream node in the ring (i.e. node  $n$ ) is zero. This means that all upstream nodes of node 1 receive the control message right after it is transmitted. For  $k \leq \hat{k}$ , only node  $n$  can add traffic to the ring. Therefore, we have

$$\tilde{r}_i^a(\hat{k}) = 0, \quad i = 1, 2, \dots, n - 1. \quad (4.1.5)$$

When congestion is detected, node 1, as the head of the congestion domain, advertises its calculated fair rate, i.e.  $F_1(\hat{k}) = \tilde{r}_1^a(\hat{k}) = 0$ , to its upstream nodes. When upstream node  $i > 1$  receives the control message, it sets its *allowedRateCongested* to the received fair rate, i.e.  $\ell_i^c(\hat{k} + 1) = F_1(\hat{k}) = 0$ . Note that in RPR the

*allowedRateCongested* of a node only changes when that node contributes to the downstream congestion and receives a fair rate from one of its downstream congested nodes. Therefore,  $\ell_1^c(k) = 1, \forall k$ , as there is no congested node in the downstream of node 1. Consequently, from Equation (4.1.2) we have

$$r_1^a(k) = \min \{1 - r_1^f(k), \ell_1^c(k) = 1\} = 1 - r_1^f(k), \forall k. \quad (4.1.6)$$

As  $r_i^u(k) = r_i^a(k) + r_i^f(k), \forall i$ , it follows from Equation (4.1.6) that  $r_1^u(k) = 1, \forall k$ . Therefore, mimicking the reasoning used to derive Equation (4.1.3) we have  $\tilde{r}_1^u(k) = 1 - (1 - \alpha)^k, \forall k$ , which is an increasing function of  $k$ . Consequently,  $\tilde{r}_1^u(k) > \tau_\ell, \forall k \geq \hat{k}$  as  $\tilde{r}_1^u(\hat{k}) > \tau_\ell$ . This means that node 1 is congested for all  $k \geq \hat{k}$ . As a result, node 1 calculates and advertises a fair rate to its upstream nodes at the end of every control interval after the congestion detection.

Now, we investigate the fair rate calculations in each control interval after the congestion detection.

- In control interval  $k = \hat{k} + 1$ , as  $\ell_i^c(\hat{k} + 1) = F_1(\hat{k}) = 0, i > 1$ , it follows from Equation (4.1.2) that  $r_i^a(\hat{k} + 1) = 0, i > 1$ . In other words, node  $i > 1$  does not add any traffic to the ring. Consequently,  $r_1^f(\hat{k} + 1) = 0$ , and using Equation (4.1.6) we have  $r_1^a(\hat{k} + 1) = 1$ . Therefore, according to Equation (2.2.1) the new calculated fair rate of node 1 is equal to  $F_1(\hat{k} + 1) = \tilde{r}_1^a(\hat{k} + 1) = (1 - \alpha) \cdot 0 + \alpha \cdot 1 = \alpha$ .

Note that node  $i > 1$  is not congested in control interval  $\hat{k} + 1$  as  $r_i^u(\hat{k} + 1) = 0, i > 1$ . Node 1 is the head node in the congestion domain and all upstream nodes adjust their add-rates according to the advertised fair rate of node 1. At the end of control interval  $\hat{k} + 1$ , node 1 advertises  $F_1(\hat{k} + 1) = \tilde{r}_1^a(\hat{k} + 1)$  as the fair rate to its upstream nodes.

- In control interval  $k = \hat{k} + 2$ , we have  $\ell_i^c(\hat{k} + 2) = F_1(\hat{k} + 1) = \alpha, i > 1$ . This means that each node  $i > 1$  can add traffic to the ring up to the rate  $\alpha$ . It

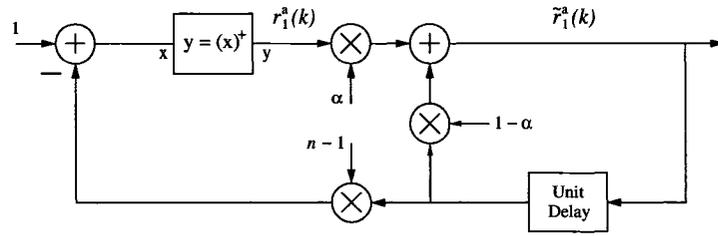


Figure 4.2: A low-pass filter model for fair rate calculations

follows from Equation (4.1.2) that  $r_i^a(\hat{k}+2) = \min \{1 - r_i^f(\hat{k}+2), \alpha\}$ ,  $i > 1$ . For the transit traffic of node 1, which is the aggregated traffic from nodes 2, 3, ...,  $n$ , we can write  $r_1^f(\hat{k}+2) = \min \{1, (n-1) \cdot \alpha\}$ . Hence, the add-rate of node 1 is  $r_1^a(\hat{k}+2) = 1 - r_1^f(\hat{k}+2) = \max \{0, 1 - (n-1) \cdot \alpha\}$ . Then, node 1 calculates its new fair rate as  $F_1(\hat{k}+2) = \tilde{r}_1^a(\hat{k}+2) = (1-\alpha) \cdot \alpha + \alpha \cdot r_1^a(\hat{k}+2)$ , which is advertised to its upstream nodes.

- In general, in control interval  $k > \hat{k}$ ,  $\ell_i^c(k) = F_1(k-1) = \tilde{r}_1^a(k-1)$ ,  $i > 1$ ,  $r_1^f(k) = \min \{1, (n-1) \cdot \tilde{r}_1^a(k-1)\}$ , and the add-rate of node 1 is obtained as

$$r_1^a(k) = (1 - (n-1) \cdot \tilde{r}_1^a(k-1))^+, k > \hat{k}, \quad (4.1.7)$$

where  $(x)^+ = \max \{0, x\}$ . Consequently, the fair rate of node 1 at the end of control interval  $k > \hat{k}$  can be calculated as follows:

$$F_1(k) = \tilde{r}_1^a(k) = (1-\alpha) \cdot \tilde{r}_1^a(k-1) + \alpha \cdot (1 - (n-1) \cdot \tilde{r}_1^a(k-1))^+. \quad (4.1.8)$$

It can be seen that  $\tilde{r}_1^a(k)$  depends on  $\alpha$ ,  $n$ , and the previous calculated fair rate,  $F_1(k-1) = \tilde{r}_1^a(k-1)$ . Therefore, we can model the fair rate calculation process at node 1 using the *first-order* nonlinear discrete-time filter depicted in Figure 4.2, where the unit delay is equal to one control interval. The advertised fair rate of node 1 is the response of this filter to a unit step function applied at  $k = \hat{k} + 1$  with initial condition  $\tilde{r}_1^a(\hat{k}) = 0$ .

### Condition for Finite Convergence Time of Fair Rate

Consider the filter in Figure 4.2 and assume that its nonlinear block is bypassed (i.e.,  $y = (x)^+$  replaced by  $y = x$ ). In this case, the system is a first-order *linear* discrete-time filter with transfer function

$$H(z) = \frac{\alpha}{1 + (\alpha \cdot n - 1) \cdot z^{-1}}. \quad (4.1.9)$$

We refer to the above linear system as the *corresponding linear system* of the nonlinear model which is obtained by bypassing the nonlinear block of the nonlinear model. This corresponding linear system is stable when its pole is inside the unit circle, which yields

$$\alpha < \frac{2}{n}. \quad (4.1.10)$$

In Appendix A, Proposition A.1.1 provides the stability condition of the nonlinear system depicted in Figure 4.2. It is proved that the stability condition of the nonlinear system presented in Figure 4.2 is given by Inequality (4.1.10) as well. That is, if  $\alpha < \frac{2}{n}$ , the convergence time of the calculated fair rate is finite.

Note that when the fair rate converges, we have  $\lim_{k \rightarrow \infty} r_i^a(k) = \frac{1}{n}$ ,  $\forall i$ . In this case,  $\lim_{k \rightarrow \infty} \tilde{r}_i^u(k) = 1 - \frac{i-1}{n}$ , where  $\tilde{r}_1^u(k) > \tilde{r}_2^u(k) > \dots > \tilde{r}_n^u(k)$ . In this analysis, we assume that the congestion threshold,  $\tau_\ell$ , is set to a value greater than  $1 - \frac{1}{n}$  and close to 1. This means that  $\tilde{r}_1^u(k) > \tau_\ell > \tilde{r}_2^u(k) > \dots > \tilde{r}_n^u(k)$  and, as a result, node 1 remains the most congested node after the congestion detection and all other nodes adjust their local rates according to the advertised fair rate of node 1.

#### 4.1.3 Analysis of Model with Link Propagation Delay

We now consider the effect of the link propagation delay on fair rate calculations. Up to the congestion detection time, the system with the propagation delay works similar to the ideal model and the congestion detection time can still be calculated

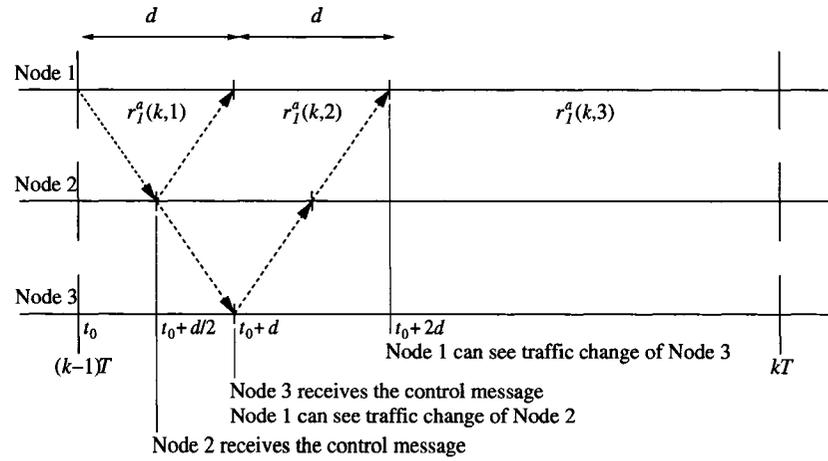


Figure 4.3: Effect of propagation delay on advertising the fair rate in RPR-AM

from Equation (4.1.4). We define  $d$  as the one-hop *round-trip* propagation delay. In other words, the propagation delay of each link is equal to  $\frac{d}{2}$ . Figure 4.3 illustrates the effect of the link propagation delay on the add-rate of node 1 in a scenario with  $n = 3$ . It presents a typical control interval  $k > \hat{k}$  at which node 1 sends a control message to its upstream nodes. It also shows the time when the upstream nodes receive the control message.

At the same time ( $t_0 + d$  seconds), node 3 receives the control message forwarded by node 2. This node also sets its *allowedRateCongested* to the advertised fair rate. Node 1 observes the effect of the rate adjustment of node 3 at ( $t_0 + 2 \cdot d$ ) seconds. In general, node 1 can see the traffic change of node  $j > 1$  due to the most recently advertised fair rate at ( $t_0 + (j - 1) \cdot d$ ) seconds. In order to simplify the problem, we assume that the time needed for the congested node 1 to see the traffic change of node  $n$ , which is the furthest node contributing to congestion, is less than the length of control interval. That is,

$$(n - 1) \cdot d < T. \quad (4.1.11)$$

Therefore, after advertising a new fair rate at the beginning of each control interval,

congested node 1 can see the effect of traffic change of all nodes contributing to congestion in the same control interval. As a result, it can make a more solid estimation of the true fair rate. Note that because of the link propagation delay, node 1 gradually sees the effect of traffic change of its upstream nodes. Therefore, its add-rate changes during a control interval. In this case, the average add-rate of node 1 during a control interval should be measured in order to calculate the fair rate. Let us define the following notations:

- ▶ Subinterval  $j$ ,  $1 \leq j \leq n - 1$ , of control interval  $k > \hat{k}$  is the time interval  $((k - 1) \cdot T + (j - 1) \cdot d, (k - 1) \cdot T + j \cdot d]$ . At the beginning of subinterval  $j$  of control interval  $k$ , node 1 can see the effect of the add-rate adjustment of node  $j$  due to the advertised fair rate, i.e.  $F_1(k - 1)$ .
- ▶  $r_1^a(k, j)$  is the add-rate of node 1 in subinterval  $j$  of control interval  $k$ . In other words,  $r_1^a(k, j)$  corresponds to the rate adjustment of node  $j$  due to the advertised fair rate  $F_1(k - 1)$ .
- ▶  $r_1^a(k, n)$  is the add-rate of node 1 in time interval  $((k - 1) \cdot T + (n - 1) \cdot d, k \cdot T]$ . This represents the add-rate of node 1 when the effect of the rate adjustment of all upstream nodes (corresponding to the advertised fair rate  $F_1(k - 1)$ ) can be observed by node 1.

In Figure 4.3, the subintervals of control interval  $k$  with the corresponding add-rates of node 1 are shown for  $n = 3$ . Now, consider node 1 at the beginning of control interval  $k > \hat{k}$ , when fair rate  $F_1(k - 1)$  is just advertised to the upstream nodes. We can make the following observations on the add-rate of node 1 during control interval  $k$ :

- During subinterval  $j = 1$ , node 1 cannot see the effect of the latest advertised rate, i.e.  $F_1(k - 1) = \tilde{r}_1^a(k - 1)$ . The transit traffic of node 1 in this subinterval

corresponds to the previous advertised fair rate, i.e.  $F_1(k-2) = \tilde{r}_1^a(k-2)$ .

Using Equation (4.1.7), the add-rate of node 1 in this subinterval is given by

$$r_1^a(k, 1) = (1 - (n-1) \cdot F_1(k-2))^+ = (1 - (n-1) \cdot \tilde{r}_1^a(k-2))^+.$$

- In subinterval  $j = 2$ , the traffic change of node 2 can be seen as  $d$  seconds has elapsed since the latest fair rate advertisement. However, the traffic of nodes 3, 4, ..., and  $n$  (observed at node 1) still corresponds to the previous advertised fair rate. In this case,  $r_1^a(k, 2) = (1 - (n-2) \cdot F_1(k-2) - F_1(k-1))^+$ .

- In general, for subinterval  $j$  ( $1 \leq j \leq n-1$ ) we have

$$r_1^a(k, j) = (1 - (n-j) \cdot F_1(k-2) - (j-1) \cdot F_1(k-1))^+. \quad (4.1.12)$$

- During time interval  $((k-1) \cdot T + (n-1) \cdot d, k \cdot T]$ , the forward-rate of node 1 only corresponds to the latest advertised rate. Hence, the add-rate of node 1 in this time interval is obtained as,

$$r_1^a(k, n) = (1 - (n-1) \cdot F_1(k-1))^+. \quad (4.1.13)$$

Using Equations (4.1.12) and (4.1.13), we can now calculate the add-rate of node 1 in control interval  $k > \hat{k}$  by obtaining time-average of its add-rate over this control interval as follows:

$$r_1^a(k) = \left(1 - (n-1) \cdot \frac{d}{T}\right) \cdot r_1^a(k, n) + \frac{d}{T} \cdot \sum_{j=1}^{n-1} r_1^a(k, j), \quad (4.1.14)$$

where  $r_1^a(k, j) = (1 - (n-j) \cdot F_1(k-2) - (j-1) \cdot F_1(k-1))^+$ ,  $1 \leq j \leq n$ .

Note that calculation of the add-rate of node 1 in Equation (4.1.14) is a recursive function as its value in control interval  $k$  depends on  $F_1(k-1) = \tilde{r}_1^a(k-1)$  and  $F_1(k-2) = \tilde{r}_1^a(k-2)$ . The fair rate of node 1 in control interval  $k > \hat{k}$ , i.e.  $F_1(k) = \tilde{r}_1^a(k)$ , is obtained by substituting Equation (4.1.14) in Equation (2.2.1) as

$$F_1(k) = \alpha \cdot r_1^a(k) + (1 - \alpha) \cdot F_1(k-1). \quad (4.1.15)$$

One can see that  $F_1(k)$ ,  $k > \hat{k}$ , depends on  $\alpha$ ,  $n$ ,  $F_1(k-2)$ , and  $F_1(k-1)$ . Therefore, similar to the method used in the ideal case ( $d = 0$ ), we can model the fair rate calculations at node 1 by a *second-order* nonlinear discrete-time filter. In this case, the fair rate of node 1 will be the response of the second-order nonlinear discrete-time filter to a unit step function applied at  $k = \hat{k} + 1$ .

In order to calculate  $F_1(k)$ ,  $k > \hat{k}$ , we need to know  $F_1(k-1)$  and  $F_1(k-2)$ . Specifically, to calculate  $F_1(\hat{k} + 1)$ , two initial values of the fair rate, i.e.  $F_1(\hat{k} - 1)$ , and  $F_1(\hat{k})$  are required. In the following, we calculate  $F_1(\hat{k} - 1)$  and  $F_1(\hat{k})$ .

Recall that at the end of control interval  $k \geq \hat{k}$ , node 1 advertises its fair rate, i.e.  $F_1(k) = \tilde{r}_1^a(k)$  to its upstream nodes. The upstream nodes then set their *allowedRateCongested* to  $F_1(k)$ . Before congestion detection, however, node 1 does not advertise any fair rate and the *allowedRateCongested* value of all upstream nodes is equal to 1. This resembles a situation where the advertised fair rate of node 1 before congestion detection ( $k < \hat{k}$ ) equals to 1. In other words, we can assume that  $F_1(k) = 1$ ,  $k < \hat{k}$  and, hence,  $F_1(\hat{k} - 1) \equiv 1$ .

We can also calculate the exact value of the first advertised fair rate,  $r_1^a(\hat{k})$ . During control intervals 2, 3, ..., and  $\hat{k}$  node 1 could not send any traffic. Hence,

$$r_1^a(k) = 0, \quad k = 2, 3, \dots, \hat{k}. \quad (4.1.16)$$

However, in the first control interval, node 1 can send traffic at the link rate for a short period of time before traffic of node 2 reaches its transit buffer. This time is equal to the one-hop link propagation delay, i.e.  $\frac{d}{2}$ . Similar to node 1, nodes 2, 3, ..., and  $n-1$  can only send traffic at the link rate for  $\frac{d}{2}$  seconds in the first control interval. After that, only node  $n$  sends traffic until congestion is detected. The average add-rate of node 1 in the first control interval is

$$r_1^a(1) = \frac{d}{2T}. \quad (4.1.17)$$

Using Equations (2.2.1), (4.1.16) and (4.1.17), we can calculate the fair rate at the time of congestion detection as follows:

$$F_1(\hat{k}) = \tilde{r}_1^a(\hat{k}) = \alpha \cdot (1 - \alpha)^{\hat{k}-1} \cdot \frac{d}{2T}. \quad (4.1.18)$$

Thus, the fair rate of node 1 in an arbitrary control interval  $k > \hat{k}$  can be recursively obtained using Equations (4.1.14), (4.1.15), (4.1.18), and  $F_1(\hat{k} - 1) \equiv 1$ .

We implemented this analytical model in MATLAB to study the effect of the propagation delay and the low-pass filtering coefficient on the calculated fair rate. We observed similar properties between the model with the propagation delay and its corresponding linear system, which is obtained by bypassing all non-linear blocks in Equation (4.1.14). For instance, we observed that the stability domain of the second-order nonlinear system (with respect to  $\alpha$  for a fixed  $n$ ) is similar to its corresponding linear system. Further, Proposition A.2.1 in Appendix A compares the stability domain of the corresponding linear system of the model with the propagation delay with the one for the ideal model without the propagation delay. It is proved that the corresponding linear system of the model with the propagation delay is stable for a wider range of  $\alpha$  compared to the model without the propagation delay. It is also shown in Proposition A.2.2 in Appendix A that the maximum range of  $\alpha$  with respect to which the corresponding linear system of the model with the propagation delay remains stable, is achieved when  $(n - 1) \cdot d = 0.5 \cdot T$ .

Note that in the analysis of the model with the link propagation delay, we assumed that Inequality (4.1.11) is maintained. However, for a large ring we may have  $(n - 1) \cdot d > T$ . In this case, the calculated fair rate in control interval  $k$  would depend on  $F_1(k - 1)$ ,  $F_1(k - 2)$ ,  $F_1(k - 3)$ , and etc., which makes the analysis more complex.

In addition, in this analysis we used a simple scheme for transmitting the control message where a single control message circulates throughout the ring and each node can forward the control message right after receiving it. If we considered the first

method for control message transmissions, which was explained in Section 4.1.1, each node could only transmit a control message at the end of each control interval. In this case, the fair rate of the head node of the congestion domain would propagate only one hop in each control interval. This means that the fair rate of node 1 calculated at the end of control interval  $k$  is received by node  $i > 1$  during control interval  $k + i - 1$ . Therefore, the add-rate of node 1 in control interval  $k$  would depend on  $F_1(k - 1)$ ,  $F_1(k - 2)$ ,  $\dots$ , and  $F_1(k - n + 1)$ . In this case, the fair rate calculations can be modeled as a non-linear system where the order of such a system linearly increases with the number of nodes. In this analysis, however, the employed scheme for control message transmissions results in a second-order nonlinear low-pass filter model as long as Inequality (4.1.11) is maintained.

#### 4.1.4 Numerical Results and Discussions

In this section, we compare the analytical results with the ones from simulations to examine the accuracy of the model. We also study the impact of various parameters on the calculated fair rate. The simulation model is implemented in OPNET and we considered 100Mbps links, 512 bits packet size, and  $T = 1\text{msec}$ .

##### Effect of $\alpha$ on Fair Rate Oscillations

The oscillations of the fair rate (before converging to the true fair rate) result from overestimation or underestimation of the rates. As the oscillations contribute to the convergence time, they are part of the performance of the RPR bandwidth allocation algorithms. Figure 4.4 presents both analytical and simulation results for the calculated fair rate in a scenario with  $n = 8$ ,  $d = 0.1 \cdot T$ ,  $\tau_\ell = 0.95$ , for  $\alpha = 0.05$ ,  $0.15$ , and  $0.25$ . One can first note that analytical and simulation results are quite close. The developed analytical model is a very accurate approximation of the real

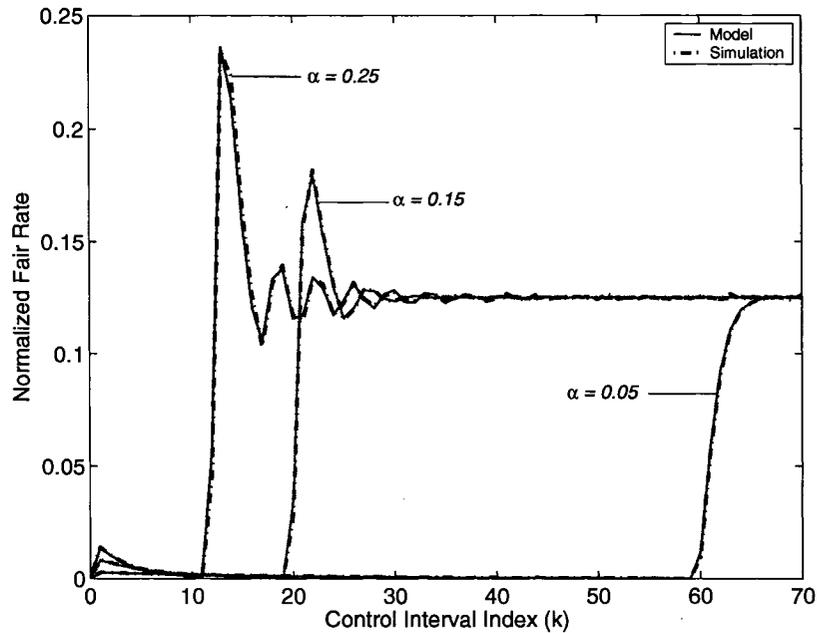


Figure 4.4: Normalized calculated fair rate for  $n = 8$ ,  $\tau_\ell = 0.95$ , and  $d = 0.1 \cdot T$

system and, therefore, it can be used to evaluate the performance of RPR-AM. The observed error between the analytical and simulation results is due to the fact that the add-rate of each node is controlled by a token-bucket in the simulation model. As a result, the add-rate of each node may not be instantly adjusted to the received fair rate due to existing tokens in the corresponding token-bucket. Nonetheless, the error rate between analytical and simulation results is less than 2% in all cases. In Figure 4.4, congestion is detected at control intervals 59, 19, and 11 for  $\alpha = 0.05$ , 0.15, and 0.25, respectively. One can see that as  $\alpha$  increases, the congestion detection time decreases. However, the calculated fair rate converges to the true fair rate with a larger number of oscillations. The reason is that when  $\alpha$  is large, the latest measured data has more weight in the calculation of the next fair rate according to Equation (2.2.1), and the true fair rate may not be estimated accurately. Thus, the fair rate may be overestimated, then underestimated, and so on.

Note that in control interval  $k = 1$ , node 1 is able to send traffic for  $\frac{d}{2}$  seconds

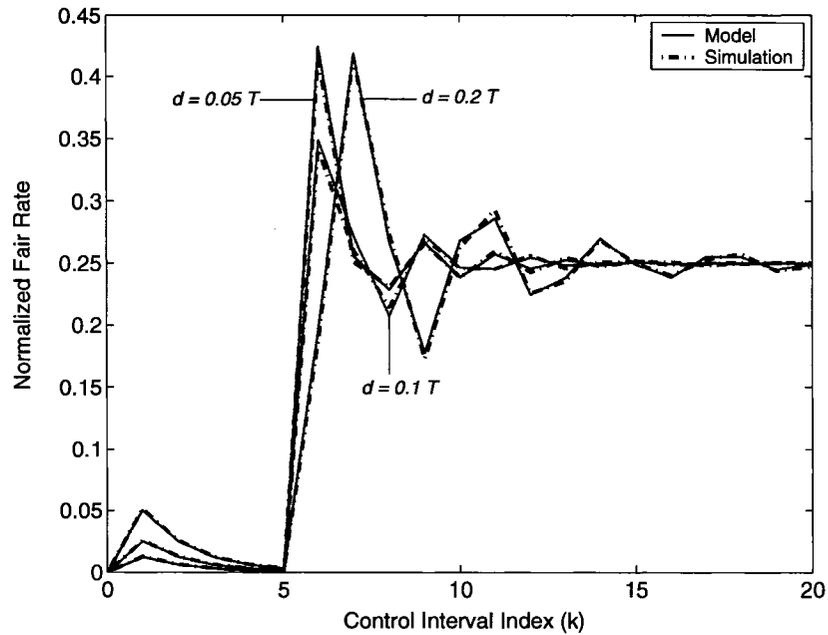


Figure 4.5: Normalized calculated fair rate for  $n = 4$ ,  $\tau_\ell = 0.95$ , and  $\alpha = 0.5$ .

according to Equation (4.1.17) and, as a result,  $r_1^a(1) > 0$ . Node 1 is then unable to send any traffic until congestion is detected. Hence, the low-pass filtered version of its add-rate decreases prior to the congestion detection time. This can be observed in Figure 4.4 where the fair rate (i.e. the low-pass filtered of add-rate of node 1) decreases for  $1 < k \leq \hat{k}$ .

### Effect of Propagation Delay on Fair Rate Oscillations

Figure 4.5 illustrates the effect of the propagation delay on the calculated fair rate for  $n = 4$  and  $\alpha = 0.5$ . Both analytical and simulation results are shown. A close match between simulation and analytical results is observed. In this case, congestion is detected at the end of the 5<sup>th</sup> control interval ( $\hat{k} = 5$ ). As the propagation delay increases, the congested node observes the effect of traffic changes due to the last advertised rate in a smaller portion of the control interval, which changes the value

of the calculated rate. For example for  $d = 0.2 \cdot T$ , the first overshoot of the fair rate occurs with one control interval delay when compared to the other cases due to the propagation delay.

### Effect of Propagation Delay on Stability of Fair Rate Calculations

We define  $\alpha_n^*(d)$  as the supremum value of  $\alpha$  such that  $\forall \alpha < \alpha_n^*(d)$ , the calculated fair rate does converge to  $\frac{1}{n}$  in an  $n$ -node scenario with  $d$  one-hop round-trip propagation delay. In the ideal model ( $d = 0$ ), (i) the calculated fair rate in control interval  $k$  depends on the calculated fair rate in control interval  $k - 1$  according to Equation (4.1.8); and (ii)  $\alpha_n^*(0) = \frac{2}{n}$  (see Inequality (4.1.10)). In the realistic model ( $d > 0$ ), the calculated fair rate in control interval  $k$  depends on the fair rate calculated in the control intervals  $k - 1$  and  $k - 2$  (see Equation (4.1.14)). As it was mentioned in Section 4.1.3, we observed through analytical results that the system with the propagation delay is stable for a wider range of  $\alpha$  compared to the system without the propagation delay. Therefore,  $\alpha_n^*(d > 0) > \alpha_n^*(0)$ . In Proposition A.2.1 in Appendix A, we have proved this property of the model for a particular case. Figure 4.6 shows  $\alpha_n^*(d)$  versus the normalized propagation delay (i.e.  $\frac{d}{T}$ ) for different number of nodes. These results are obtained using the analytical model. It can be seen in Figure 4.6 that  $\alpha_n^*(d > 0)$  is greater than  $\alpha_n^*(0) = \frac{2}{n}$ . As  $d \rightarrow 0$ , the model behaves similar to the ideal model ( $d = 0$ ) and  $\alpha_n^*(d) \rightarrow \frac{2}{n}$ . In Appendix A, we have proved that the corresponding linear system of the model with the propagation delay, which is obtained by bypassing all nonlinear blocks in Equation (4.1.14), exhibits a unique maximum value of  $\alpha_n^*(d)$  for  $\frac{d}{T} = \frac{1}{2 \cdot (n-1)}$ . This property can also be observed for the actual model (which includes the nonlinear blocks) in Figure 4.6, where for a given  $n$ , the maximum value of  $\alpha_n^*(d)$  is achieved approximately for  $\frac{d}{T} = \frac{1}{2 \cdot (n-1)}$  (or equivalently  $(n - 1) \cdot d = 0.5 \cdot T$ ).

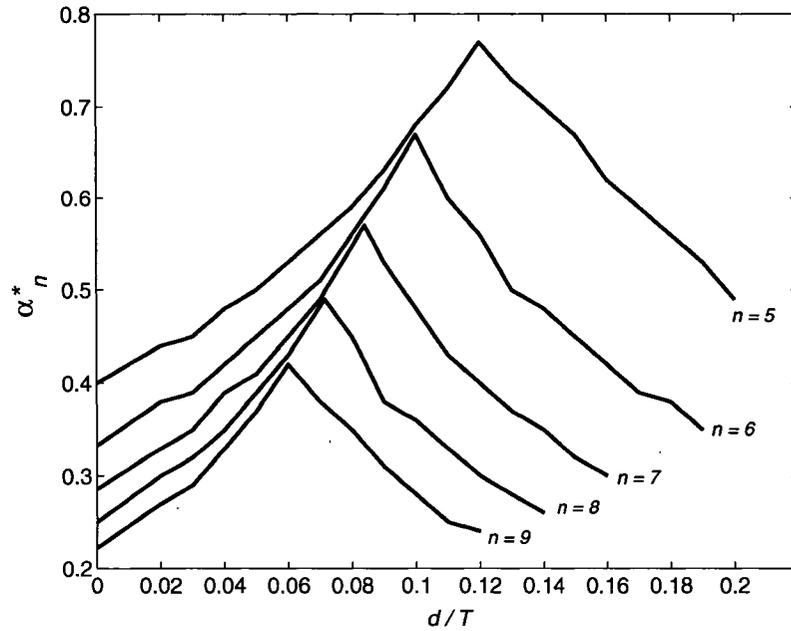


Figure 4.6: Variations of  $\alpha_n^*$  versus the normalized propagation delay

### Effect of $\alpha$ on Fair Rate Convergence Time

We define  $k_{c,n}$  as the convergence time of the system with  $n$  nodes, i.e.  $k_{c,n} = \min \{k : |r_i^a(k) - \frac{1}{n}| \leq \frac{\varepsilon}{n}, i = 1, 2, \dots, n\}$ , for a given  $\varepsilon$  ( $0 < \varepsilon < 1$ ). Figure 4.7 shows the convergence time for different number of nodes versus  $\alpha$  for  $\tau_\ell = 0.95$  and  $d = 0.1 \cdot T$  obtained from the analytical model. The convergence time consists of two components: (i) congestion detection time; and (ii) *search-time* which is defined as the time between the congestion detection and convergence of the fair rate, when nodes search for the true fair rate. The congestion detection time is independent of the number of nodes contributing to the congestion, and is a decreasing function of  $\alpha$  for a fixed congestion threshold  $\tau_\ell$ , according to Equation (4.1.4). As  $\alpha$  increases, the congestion detection time decreases, however, the calculated fair rate experiences more oscillations and, hence the search-time increases.

One can notice in Figure 4.7 that for a given value of  $n$  and small values of  $\alpha$

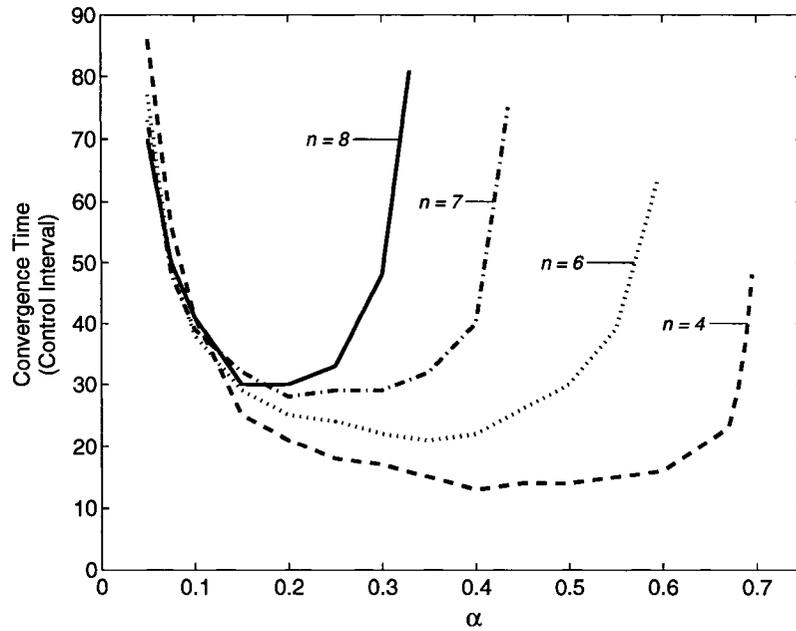


Figure 4.7: Convergence time for various values of  $n$  and  $\alpha$  with  $\varepsilon = 0.01$  and  $d = 0.1 \cdot T$

(close to 0), the convergence time is a decreasing function of  $\alpha$ . The reason is that when  $\alpha$  is close to 0, the congestion detection time (which is a decreasing function of  $\alpha$ ) is the dominant part in the convergence time.

As  $\alpha$  increases, the congestion detection time drops while the search-time rises. Therefore, the convergence time does not change significantly and remains in the same range in some extent. For example, it can be seen in Figure 4.7 that the convergence time for  $n = 4$  does not change significantly for  $\alpha$  such that  $0.2 < \alpha < 0.6$ . As  $\alpha$  gets closer to  $\alpha_n^*(0.1 \cdot T)$  (which is, for example, around 0.7 for  $n = 4$  in Figure 4.7), the search-time is the main source of the delay in the convergence time and its drastic increase.

We also observe that when  $\alpha$  is relatively small (e.g.  $\alpha < 0.05$ ), for  $n_1 > n_2$ , we have  $k_{c,n_1} < k_{c,n_2}$ . The reason is that when  $\alpha$  is small, the calculated fair rate does not oscillate and instead increasingly converges to the true fair rate (this is shown in

Figure 4.4 for  $\alpha = 0.05$ ). In this case, for the true fair rates we have  $\frac{1}{n_1} < \frac{1}{n_2}$  and, hence,  $k_{c,n_1} < k_{c,n_2}$ .

It can also be seen that as  $\alpha$  increases and estimated fair rate oscillates, the system with more nodes faces more delay in converging to the true fair rate. The reason is that for  $n_1 > n_2$ , we have  $\alpha_{n_1}^*(d) < \alpha_{n_2}^*(d)$  and if  $\alpha$  is within the stable domain of both systems, it is closer to  $\alpha_{n_1}^*(d)$  than  $\alpha_{n_2}^*(d)$ . In this case, we can conclude that the convergence time of the system with more node is longer.

Figure 4.8 presents convergence times versus  $\alpha$  for a scenario with  $n = 6$ ,  $\tau_\ell = 0.95$ ,  $\varepsilon = 0.01$ , and for  $d = 0.02 \cdot T, 0.04 \cdot T, 0.06 \cdot T$ , and  $0.08 \cdot T$ . One can see that convergence times for different values of  $d$  follow a similar pattern. That is, the convergence time for a given  $d$  decreases with  $\alpha$  when  $\alpha$  is relatively small. It then increases as  $\alpha$  approaches  $\alpha_n^*(d)$ . It can also be noted that for small values of  $\alpha$  (e.g.  $\alpha < 0.1$ ), convergence times are almost the same for different  $d$ . The reason is that when  $\alpha$  is relatively small, the congestion detection time is the dominant part of the convergence time. However, the congestion detection time is independent of  $d$  according to Equation (4.1.4). Hence, for a given  $n$ , the convergence times for different  $d$  are almost equal when  $\alpha$  is small.

Note that according to Figure 4.6,  $\alpha_n^*(d)$  is an increasing function of  $d$  for  $n = 6$  and  $d < 0.1 \cdot T$ . This can be observed in Figure 4.8 where the convergence time is finite for a wider range of  $\alpha$  as  $d$  increases.

It is very desirable to find the optimum value of  $\alpha$  which minimizes the convergence time for a given  $n$  and  $d$ . However, deriving a formula for the optimum value of  $\alpha$  as a function of  $n$  and  $d$  is very complicated in this scenario. In the following, we try to specify the values of  $\alpha$  for which the convergence time for a given  $n$  is finite and relatively short.

When  $n$  and  $d$  are known, the optimum value of  $\alpha$  can be obtained from the graphs similar to Figure 4.8 where convergence times versus  $\alpha$  are given for various values of

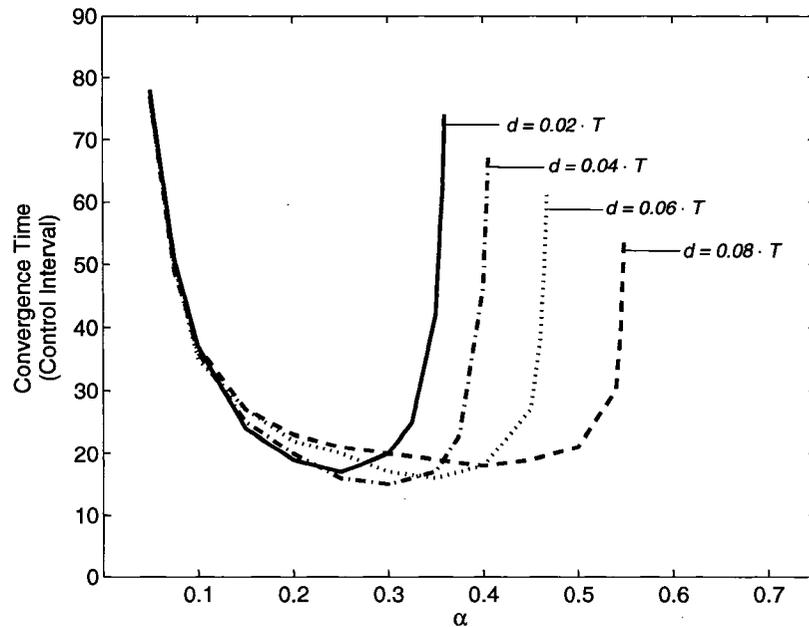


Figure 4.8: Convergence time versus  $\alpha$  for  $n = 6$  and various values of  $d$  with  $\varepsilon = 0.01$

$d$ . When  $d$  is not known,  $\alpha$  should not be set to a value close to  $\frac{2}{n}$ . The reason is that  $\alpha_n^*(d)$  could be close to  $\frac{2}{n}$  if  $d$  is relatively small. In this case, the convergence time becomes very long if  $\alpha$  is set to a value close to  $\frac{2}{n}$ . On the other hand,  $\alpha$  should not be very small as the congestion detection time and, consequently, the convergence time becomes very long. Based on observations from Figs. 4.7 and 4.8, one can say that the convergence time of a system with  $n$  nodes is relatively long when  $\alpha < \frac{1}{n}$ . Therefore, if  $d$  is not known,  $\alpha$  should be set to a value which is greater than  $\frac{1}{n}$  and not very close to  $\frac{2}{n}$  to ensure that the convergence time is relatively short. A heuristic choice for  $\alpha$  can be the middle point between  $\frac{1}{n}$  and  $\frac{2}{n}$ , that is,  $\frac{1.5}{n}$ . It can be seen in Figure 4.7 that for  $\alpha = \frac{1.5}{n}$ , the convergence time of the system with  $n$  node is very close to its minimum value for different values of  $n$ . Further, one can also see that in Figure 4.8 the convergence time with  $\alpha = \frac{1.5}{n}$  is still relatively short for different values of  $d$ . Note that as  $n$  increases,  $\alpha_n^*(d)$  decreases for a given  $d$  (see Figure 4.6). In this case, the heuristic choice for  $\alpha$  becomes a better approximation of its optimum

value as it can be seen in Figure 4.7.

## 4.2 Analysis of RPR-CM Fair Rate Calculations in Parking-lot Scenario

In the previous section, we modeled the fair rate calculation process for RPR-AM in the parking-lot scenario and studied the effect of different system parameters on the calculated fair rate through analytical results. In this section, we perform similar analysis for RPR-CM. In the following, we first describe the problem and, then, formulate the problem in two cases (i.e. with and without the propagation delay). We also provide analytical results to study the effect of different system parameters on fair rate calculations in RPR-CM.

### 4.2.1 Problem Description

In order to model the fair rate calculation process in RPR-CM, we consider exactly the same scenario and assumptions described for RPR-AM in Section 4.1.1. The only difference is that we assume RPR-CM is employed at all nodes.

In the following, we analyze the problem in two cases. First, we consider an ideal scenario where the link propagation delay is zero. We then consider the link propagation delay and analyze the fair rate calculation process in a more realistic scenario.

### 4.2.2 Analysis of Model without Link Propagation Delay

Recall from Remark 2.2.3 that the add-rate of node  $i$  in control interval  $k$ , i.e.  $r_i^a(k)$ , is limited to  $\ell_i(k)$  (*allowedRate*) and  $\ell_i^c(k)$  (*allowedRateCongested*). In addition,

$r_i^a(k)$  is bounded by  $1 - r_i^f(k)$  as the transit buffer in all nodes is implemented in single-queue mode and the transit traffic has service priority over the local traffic at each node. Hence,

$$r_i^a(k) = \min \{1 - r_i^f(k), \ell_i(k), \ell_i^c(k)\}. \quad (4.2.1)$$

Note that in RPR-AM, we have  $\ell_i(k) = 1, \forall k$ . However, in RPR-CM, node  $i$ , which is in the congested state, sets its *allowedRate* to its calculated fair rate. In the following, we compute the congestion detection time and, then, calculate the FRTT and the fair rate of the head-node of congestion span, i.e. node 1 (see Figure 4.1).

### Congestion Detection Time

Starting at time 0 (or beginning of control interval  $k = 1$ ), each node has traffic to send to node 0. The initial values of *allowedRate* and *allowedRateCongested* of all nodes are equal to the link rate, i.e.  $\ell_i(1) = \ell_i^c(1) = 1, i = 1, 2, \dots, n$ . As a result, similar to the scenario with RPR-AM (see Section 4.2.2), only node  $n$  which is the most upstream node in this scenario can send traffic at the link rate; other nodes are not able to add their local traffic to the ring before congestion detection. Consequently, the congestion detection time (i.e.  $\hat{k}$ ) can be obtained from Equation (4.1.4). When node  $i > 0$  detects the congestion at  $k = \hat{k}$ , it enters the congested state.

Note that in this scenario, node 1 is the most congested node (or head-node of the congestion span) and each node  $i > 1$  sets  $\ell_i^c(k), k > \hat{k}$ , to the advertised fair rate of node 1.

### Fairness Round-Trip Time (FRTT)

As the size of the control message is relatively short, we can ignore its transmission and processing time. We have also assumed that the link propagation delay is zero in this ideal case. As a result, the time needed for the control message to reach the

furthest upstream node in the ring (i.e. node  $n$ ) is negligible and we can assume that all nodes receive and apply the fair rate of node 1 with no delay after it is advertised. Therefore, the FRTT of node 1 (i.e. the time needed for node 1 to observe the effect of its latest advertised fair rate on the add-traffic of the furthest node (i.e. node  $n$ )) is approximately zero. Recall that in RPR-CM, the congested node calculates a new fair rate at the end of a control interval when at least an FRTT has passed since the last fair rate adjustment (see Section 2.2.4). As in this scenario we have  $T > \text{FRTT} \simeq 0$ , after congestion detection a new fair rate is calculated at the end of every control interval.

### Advertised Fair Rate

As all nodes detect congestion at the end of control interval  $\hat{k}$ , they enter the congested state and calculate their fair rates. As mentioned in Section 2.2.4, when node  $i < n$  detects the congestion for the first time in control interval  $\hat{k}$ , its initial fair rate is set to  $1/\Delta_i(\hat{k})$ , where  $\Delta_i(\hat{k}) = 2$ . The reason is that, node  $n$  is the only node sending traffic through node  $i < n$ . In addition, each node  $i < n$  considers itself as an active node when calculating  $\Delta_i(\hat{k})$ . Hence, the initial calculated fair rate of node  $i < n$  is

$$F_i(\hat{k}) = 0.5, \quad i = 1, 2, \dots, n-1. \quad (4.2.2)$$

Note that for node  $n$  we have  $\Delta_n(\hat{k}) = 1$ . This means that  $F_n(\hat{k}) = 1$  and, as a result, node  $n$  is considered uncongested. When node  $i$  calculates a fair rate, it sets its *allowedRate* to its own calculated fair rate. Therefore,

$$\ell_i(k) = F_i(k-1), \quad \forall k > \hat{k}, \forall i. \quad (4.2.3)$$

Assume that node 1 initiates the control message transmission at the end of each control interval. When congestion is detected, node 1 transmits the control message to its upstream node 2 advertising its fair rate, i.e.  $F_1(\hat{k}) = 0.5$ . Node 2 receives the

control message and sets its *allowedRateCongested* to the fair rate of node 1, i.e.  $\ell_2^c(\hat{k}+1) = F_1(\hat{k}) = 0.5$ . Node 2 then forwards the message to its upstream node 3. This process repeats until all nodes  $i > 1$  adjust their *allowedRateCongested* to  $F_1(\hat{k})$ . Recall that all nodes receive and apply the fair rate with no delay after it is advertised by node 1 in this ideal case. Hence,  $\ell_i^c(\hat{k}+1) = F_1(\hat{k}) = 0.5, \forall i > 1$ . We also have  $\ell_i(\hat{k}+1) = F_i(\hat{k}) = 0.5, \forall i < n$ , and  $\ell_n(\hat{k}+1) = F_n(\hat{k}) = 1$ . Therefore, in control interval  $k = \hat{k}+1$ , according to Equation (4.2.1) we have  $r_n^a(\hat{k}+1) = r_{n-1}^a(\hat{k}+1) = 0.5$ , and  $r_i^a(\hat{k}+1) = 0, \forall i < n-1$ . Moreover,  $r_i^u(\hat{k}+1) = 1, \forall i < n$ . At the end of control interval  $k = \hat{k}+1$ , each node  $i < n$  calculates a new fair rate. Since node  $i < n$  is already in the congested state, its new fair rate, i.e.  $F_i(\hat{k}+1)$ , is obtained by ramping  $F_i(\hat{k})$ . As  $r_i^u(\hat{k}+1) = 1 > \tau_h$ , it follows from Equation (2.2.5) that  $F_i(\hat{k}+1) = 0.5 \cdot (1 - \beta), \forall i < n$ . In general, at the end of control interval  $k > \hat{k}$ , node  $i$  ramps up or ramps down its fair rate based on the value of  $r_i^u(k) = r_i^a(k) + r_i^f(k)$ .

Note that node 1 is the head-node of the congestion span in this scenario and we have

$$F_1(k) \leq F_i(k), \forall k \geq \hat{k}, i = 2, 3, \dots, n. \quad (4.2.4)$$

Therefore, all nodes set their *allowedRateCongested* to the advertised fair rate of node 1. In other words,

$$\ell_i^c(k) = F_1(k-1), \forall k > \hat{k}, i = 2, 3, \dots, n. \quad (4.2.5)$$

One can combine Equations (4.2.3) and (4.2.5) and Inequality (4.2.4), to conclude that

$$\ell_i^c(k) \leq \ell_i(k), \forall k > \hat{k}, i = 2, 3, \dots, n. \quad (4.2.6)$$

It then follows from Equations (4.2.1), (4.2.5), and Inequality (4.2.6) that

$$r_i^a(k) = \min \{1 - r_i^f(k), F_1(k-1)\}, \forall k > \hat{k}, \forall i > 1. \quad (4.2.7)$$

Note that the *allowedRateCongested* of a node only changes when that node receives a fair rate from a congested downstream node. Hence, for node 1 we have  $\ell_1^c(k) = 1$ ,

$\forall k$ , as there is no congested node on downstream of node 1. Moreover, it follows from Equation (4.2.3) that  $\ell_1(k) = F_1(k-1)$ ,  $\forall k > \hat{k}$ . As a result, according to Equation (4.2.1), the add-rate of node 1 is given by Equation (4.2.7) as well. Note that according to Equation (4.2.7), each node  $i > 0$  in control interval  $k > \hat{k}$  is allowed to add traffic to the ring up to the rate  $F_1(k-1)$ , i.e.  $r_i^a(k) \leq F_1(k-1)$ . Therefore,

$$r_1^u(k) = \sum_{i=1}^n r_i^a(k) = \min \{1, n \cdot F_1(k-1)\}, \forall k > \hat{k}. \quad (4.2.8)$$

Thus, using Equations (2.2.5), (2.2.6), (2.2.7), and (4.2.8), the fair rate of node 1 at the end of control interval  $k > \hat{k}$  is given by

$$F_1(k) = \begin{cases} (1 - \beta) \cdot F_1(k-1), & \text{if } n \cdot F_1(k-1) > \tau_h, \\ \min \{1, F_1(k-1) \\ + \beta \cdot (1 - n \cdot F_1(k-1))\}, & \text{if } n \cdot F_1(k-1) < \tau_\ell, \\ F_1(k-1), & \text{otherwise,} \end{cases} \quad (4.2.9)$$

where  $F_1(\hat{k}) = 0.5$ .

Note that the calculated fair rate of node 1, i.e.  $F_1(k)$ , depends on  $F_1(k-1)$ ,  $n$ , and  $\beta$  and it can be recursively calculated in each control interval. Once the fair rate of node 1 is known, the add-rate of each node  $i > 1$  can be obtained from Equation (4.2.7).

*Remark 4.2.1.* It can be noticed that in an  $n$ -node scenario, the fair rate of node 1 converges in control interval  $k_{c,n}$  when  $\tau_\ell \leq r_1^u(k_{c,n}) = n \cdot F_1(k_{c,n} - 1) \leq \tau_h$  and, consequently,  $F_1(k_{c,n}) = F_1(k_{c,n} - 1)$ . In other words, when convergence is achieved, we have  $\frac{1}{n} \cdot \tau_\ell \leq r_i^a(k_{c,n}) = F_1(k_{c,n} - 1) \leq \frac{1}{n} \cdot \tau_h$ . This means that the fair rate and, consequently, the add-rate of each node may converge to a value less than  $\frac{1}{n}$ .

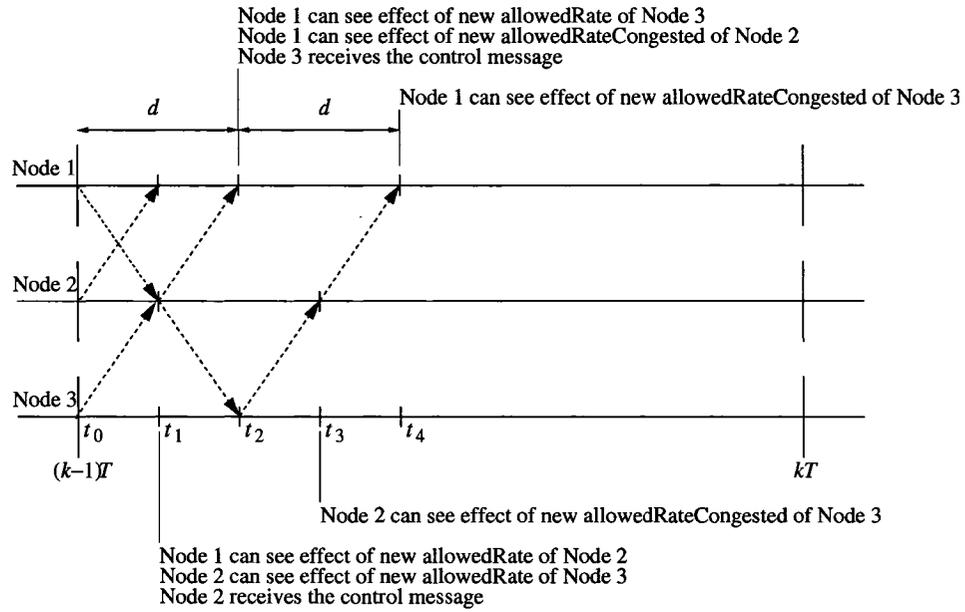


Figure 4.9: Effect of the propagation delay on advertising the fair rate in RPR-CM

### 4.2.3 Analysis of Model with Link Propagation Delay

We now consider the impact of the link propagation delay on the fair rate calculation process. Up to the congestion detection time, the system with the propagation delay works similar to the ideal model and the congestion detection time can still be computed from Equation (4.1.4). The initial fair rate of the nodes can also be obtained from Equation (4.2.2). Similar to the scenario with RPR-AM, we assume that  $d$  is the one-hop *round-trip* propagation delay where the propagation delay of each link is equal to  $\frac{d}{2}$ . Figure 4.9 presents a typical control interval  $k > \hat{k}$  in which node 1 sends the control message to its upstream nodes advertising  $F_1(k-1)$ . It illustrates how the link propagation delay affects the time when the upstream nodes receive the control message. It also shows the time when node 1 observes the effect of traffic change of each upstream node. During control interval  $k$ , one can see that

- At  $t_0 = (k-1) \cdot T$  seconds:

- Node  $i$ ,  $i = 1,2,3$ , calculates its fair rate, i.e.  $F_i(k - 1)$ , and sets its *allowedRate* to  $F_i(k - 1)$ . Hence, we have  $\ell_i(k) = F_i(k - 1)$ ,  $i = 1,2,3$ .
- At  $t_1 = t_0 + \frac{d}{2}$  seconds:
  - Node 1 can observe the effect of traffic change of node 2 due to the new value of  $\ell_2(k)$  which was set at  $t_0$ .
  - Node 2 receives the control message and sets its *allowedRateCongested* to the advertised fair rate and then forwards the control message to node 3. In addition, node 2 observes the effect of traffic change of node 3 due to the new value of  $\ell_3(k) = F_3(k - 1)$ .
- At  $t_2 = t_0 + d$  seconds:
  - Node 1 can see the effect of traffic change of node 2 due to the advertised fair rate. Moreover, node 1 observes the traffic change of node 3 due to the new value of  $\ell_3(k) = F_3(k - 1)$ .
  - Node 3 receives the control message forwarded by node 2 and sets its *allowedRateCongested* to  $F_1(k - 1)$ .
- At  $t_3 = t_0 + \frac{3}{2} \cdot d$  seconds:
  - Node 2 can see the effect of traffic change of node 3 due to the advertised fair rate.
- At  $t_4 = t_0 + 2 \cdot d$  seconds:
  - Node 1 observes the effect of the rate adjustment of node 3 due to the advertised fair rate.

In general, for node  $m > 1$  and its downstream node  $i < m$  we have

- At  $t_0$  seconds, node  $m$  sets its *allowedRate* to its calculated fair rate and we have  $\ell_m(k) = F_m(k - 1)$ .
- At  $t_0 + (m - i) \cdot \frac{d}{2}$  seconds, node  $i < m$  observes the traffic change of node  $m$  due to the new value of  $\ell_m(k)$ .
- At  $t_0 + (m - 1) \cdot \frac{d}{2}$  seconds, node  $m$  receives the control message and sets its *allowedRateCongested* to the received fair rate from node 1, i.e.  $F_1(k - 1)$ .
- At  $t_0 + (2 \cdot m - i - 1) \cdot \frac{d}{2}$  seconds, node  $i < m$  can see the effect of traffic change of node  $m$  due to the received fair rate from node 1.

In this scenario, node  $n$ , which is the furthest node contributing to congestion, sets its *allowedRateCongested* to the advertised fair rate at  $t_0 + (n - 1) \cdot \frac{d}{2}$ . Then, node 1 observes the effect of traffic change of node  $n$  due to the advertised fair rate at  $t_0 + (n - 1) \cdot d$  seconds. In other words,  $\text{FRTT} = (n - 1) \cdot d$ . To simplify the problem, we assume that

$$(n - 1) \cdot d < T. \quad (4.2.10)$$

Therefore, after congestion detection, node 1 calculates and advertises a new fair rate at the end of every control interval (since  $\text{FRTT} < T$ ) and observes the effect of the advertised fair rate from all nodes in the same control interval.

Note that at the beginning of control interval  $k$ , node  $i > 1$  sets  $\ell_i(k)$  to its calculated fair rate, i.e.  $F_i(k - 1)$ . This may change the add-rate of node  $i$  as it is always restricted to  $\ell_i(k)$ . On the other hand, as each link has propagation delay, node  $i$  receives the control message with delay within control interval  $k$  and sets its *allowedRateCongested* to the advertised fair rate of node 1. The add-rate of node  $i$  may change after receiving the control message as it is also limited to the *allowedRateCongested* of node  $i$ . Note that before receiving the control message, the *allowedRateCongested* of node  $i$  was set to the previous fair rate, i.e.  $F_1(k - 2)$ .

Therefore, during control interval  $k$ , the add-rate of node  $i$  depends on  $\ell_i(k) = F_i(k-1)$ ,  $F_1(k-1)$ , and  $F_1(k-2)$ . Let us define the following notations:

- ▶ Subinterval  $j$ ,  $1 \leq j \leq 2 \cdot (n-1)$ , of control interval  $k > \hat{k}$  is the time interval  $\left((k-1) \cdot T + (j-1) \cdot \frac{d}{2}, (k-1) \cdot T + j \cdot \frac{d}{2}\right]$  with the duration of  $\frac{d}{2}$  seconds.
- ▶ Subinterval  $j = 2 \cdot (n-1) + 1$  of control interval  $k$  is the time interval  $\left((k-1) \cdot T + (n-1) \cdot d, k \cdot T\right]$  with the duration of  $T - (n-1) \cdot d$  seconds.
- ▶  $r_{m,i}^a(k, j)$  is the add-rate of node  $m$  observed by node  $i$  ( $1 \leq i \leq m$ ) in subinterval  $j$  of control interval  $k$ .
- ▶  $r_i^u(k, j)$  is the usage rate of the output link of node  $i$  in subinterval  $j$  of control interval  $k$ .

In order to calculate the fair rate of node  $i > 0$  in control interval  $k > \hat{k}$ ,  $r_i^u(k)$  should be computed. However,  $r_i^u(k)$  depends on the add-rate of node  $i$  and the add-rates of its upstream nodes which change during control interval  $k$  due to the propagation delay. Consider node  $i$  and its upstream node  $m$  ( $m \geq i$ ):

- At the beginning of control interval  $k$ , node  $m$  sets its *allowedRate* to its local calculated fair rate, i.e.  $\ell_m(k) = F_m(k-1)$ . This can change the add-rate of node  $m$  as it is restricted to  $\ell_m(k)$ . Node  $i$  observes the effect of the new value of  $\ell_m(k)$  on the add-rate of node  $m$  after  $(m-i) \cdot \frac{d}{2}$  seconds. Therefore, during subinterval  $j \leq m-i$ , the add-rate of node  $m$  observed by node  $i$  depends on  $\ell_m(k-1) = F_m(k-2)$  and the *allowedRateCongested* of node  $m$  which is still equal to  $F_1(k-2)$ . Note that as node 1 is the most congested node, we have  $F_1(k-2) \leq F_i(k-2)$ ,  $i = 2, 3, \dots, n$ . Hence, during subinterval  $j \leq m-i$ , the add-rate of node  $m$  observed by node  $i$  is limited to  $F_1(k-2)$ . In other words,

$$r_{m,i}^a(k, j) = F_1(k-2), \quad j \leq m-i. \quad (4.2.11)$$

- Node  $m$  receives the advertised fair rate of node 1, i.e.  $F_1(k-1)$  at  $t_0 + (m-1) \cdot \frac{d}{2}$  and sets its *allowedRateCongested* to  $F_1(k-1)$ . Node  $i$  observes the effect of the new fair rate on the add-rate of node  $m$  at  $t_0 + (2 \cdot m - i - 1) \cdot \frac{d}{2}$  seconds. Therefore, during subinterval  $j$ ,  $m-i < j \leq 2 \cdot m - i - 1$ , the add-rate of node  $m$  observed by node  $i$  depends on  $\ell_m(k) = F_m(k-1)$  and the previous advertised fair rate of node 1, i.e.  $F_1(k-2)$ . In other words,

$$r_{m,i}^a(k, j) = \min\{F_1(k-2), F_m(k-1)\}, \quad m-i < j \leq 2 \cdot m - i - 1. \quad (4.2.12)$$

- In subinterval  $j$ ,  $j > 2 \cdot m - i - 1$ , node  $i$  observes the effect of the new advertised fair rate of node 1 on the add-rate of node  $m$ . Note that when node  $m$  receives the fair rate, it sets its *allowedRateCongested* to  $F_1(k-1)$  while  $\ell_m(k) = F_m(k-1)$ . However, node 1 is the most congested node and we have  $F_1(k-1) \leq F_m(k-1)$ . Therefore, after receiving the new fair rate of node 1, the add-rate of node  $m$  is only limited to  $F_1(k-1)$ . Thus,

$$r_{m,i}^a(k, j) = F_1(k-1), \quad j > 2 \cdot m - i - 1. \quad (4.2.13)$$

Therefore, the add-rate of node  $m$  observed at node  $i$  in subinterval  $j$  of control interval  $k$  is obtained as

$$r_{m,i}^a(k, j) = \begin{cases} F_1(k-2), & \text{if } j \leq m-i, \\ \min\{F_1(k-2), F_m(k-1)\}, & \text{if } m-i < j \leq 2 \cdot m - i - 1, \\ F_1(k-1), & \text{otherwise.} \end{cases} \quad (4.2.14)$$

We can now calculate  $r_i^u(k, j)$ , which is the aggregate add-rate of all nodes  $m \geq i$  observed by node  $i$  on its output link in control interval  $j$ , as follows:

$$r_i^u(k, j) = \min \left\{ \sum_{m=i}^n r_{m,i}^a(k, j), 1 \right\}. \quad (4.2.15)$$

Note that in Equation (4.2.15),  $r_i^u(k, j)$  is limited to the normalized link rate, i.e. 1. Now, we can compute  $r_i^u(k)$  by taking time-average of the usage rate of the output link of node  $i$  over control interval  $k$  as follows:

$$r_i^u(k) = \left(1 - (n-1) \cdot \frac{d}{T}\right) \cdot r_i^u(k, 2 \cdot (n-1) + 1) + \frac{d}{2 \cdot T} \sum_{j=1}^{2 \cdot (n-1)} r_i^u(k, j). \quad (4.2.16)$$

Note that at  $t_0 + (n-1) \cdot d$  seconds, the effect of the new advertised fair rate on the add-rate of node  $n$  (which is the furthest node in the congestion span) can be observed by node 1. Therefore, the link usage rate of all nodes  $i > 0$  does not change during subinterval  $j = 2 \cdot (n-1) + 1$  (time interval  $[(k-1) \cdot T + (n-1) \cdot d, k \cdot T]$ ). As the duration of subinterval  $2 \cdot (n-1) + 1$  is equal to  $T - (n-1) \cdot d$  seconds, we considered  $r_i^u(k, 2 \cdot (n-1) + 1)$  in Equation (4.2.16) with weight  $\frac{1}{T} \cdot (T - (n-1) \cdot d)$ . The duration of each control interval  $j$ ,  $j \leq 2 \cdot (n-1)$ , is equal to  $\frac{d}{2}$  seconds and, as a result, the weight of  $r_i^u(k, j)$ ,  $j \leq 2 \cdot (n-1)$ , is equal to  $\frac{1}{T} \cdot \frac{d}{2}$  in Equation (4.2.16).

The fair rate of node  $i$  is then obtained according to the value of  $r_i^u(k)$  (obtained from Equation (4.2.16)) with respect to  $\tau_h$  and  $\tau_\ell$  from (Equations 2.2.5), (2.2.6), (2.2.7). Note that  $r_i^u(k)$  depends on  $F_1(k-1)$ ,  $F_1(k-2)$  and  $F_m(k-1)$ ,  $m \geq i$ . For the initial fair rates, we define  $F_1(\hat{k}-1) \equiv 1$  and we have  $F_i(\hat{k}) = 0.5$ ,  $1 \leq i < n$ , and  $F_n(\hat{k}) = 1$ .

#### 4.2.4 Numerical Results and Discussions

This section presents our analytical and simulation results. We compare the analytical results with the ones obtained from the simulation to verify the accuracy of our model. The simulation model is implemented in OPNET. We consider 100Mbps links, 512 bits packet size, and  $T = 1\text{msec}$ .

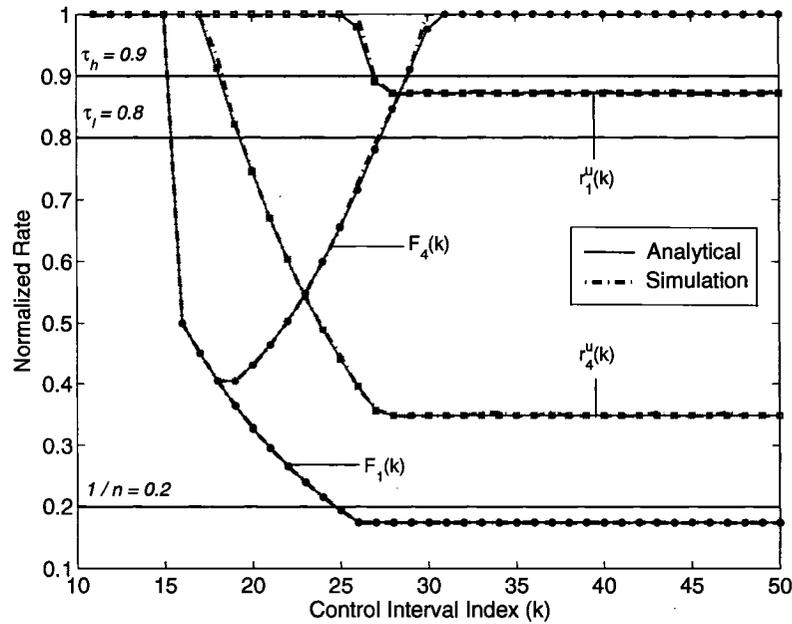


Figure 4.10: An example to illustrate dynamics of RPR-CM

### Dynamics of Fair Rate Calculations in Conservative Mode

Figure 4.10 presents both analytical and simulation results for a scenario with  $n = 5$ ,  $\alpha = 0.1$ ,  $\beta = 0.1$ ,  $d = 0.1 \cdot T$ ,  $\tau_\ell = 0.8$ , and  $\tau_h = 0.9$ . The calculated fair rates for node 1 and one of its upstream nodes (i.e. node 4) as well as their corresponding link usage rates are shown. One can first notice that analytical and simulation results are very close. In fact, our analytical model is very accurate and can be used to evaluate the performance of RPR-CM. In this scenario, congestion is detected at  $\hat{k} = 16$ . Before congestion detection, the fair rates do not change and we have  $F_1(k) = F_4(k) = 1$ . As explained before, at the time of congestion detection, the fair rate of the congested nodes is set to 0.5. This can be seen in Figure 4.10 that when congestion is detected at  $\hat{k} = 16$ ,  $F_1(\hat{k})$  (respectively,  $F_4(\hat{k})$ ) is set to 0.5. The fair rates are then ramped down in the following control intervals as the link usage at both nodes is greater than  $\tau_h$ . One can see that at  $k = 20$ , we have  $r_4^u(k) < \tau_\ell$  and, as a result, the fair rate of node 4

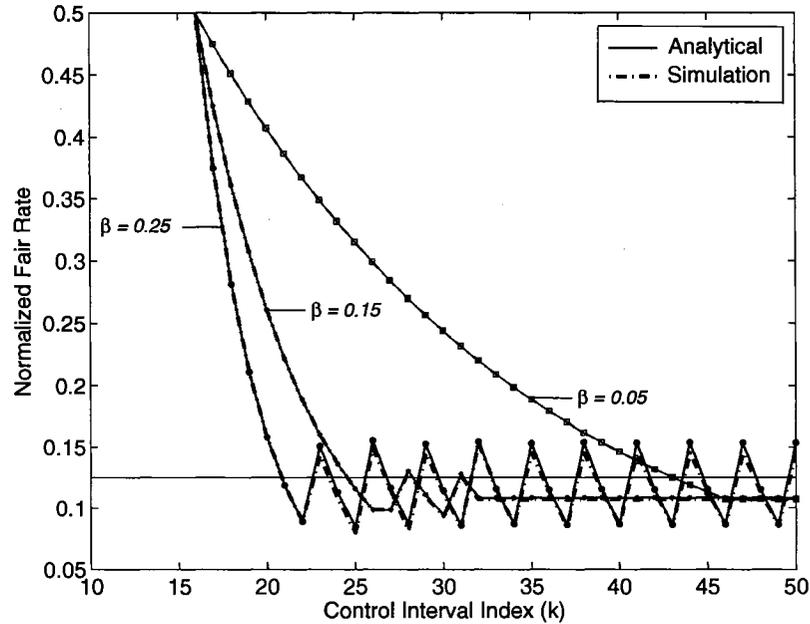


Figure 4.11: Effect of  $\beta$  on convergence of fair rate

is ramped up. Meanwhile,  $F_1(k)$  is still ramped down as  $r_1^u(k) > \tau_h$ . Note that for  $k \geq 27$ , we have  $\tau_\ell < r_1^u(k) < \tau_h$  and the fair rate of node 1 converges. One can notice that at  $k = 30$ ,  $F_4(k)$  is fully ramped up and node 4 enters the uncongested state. However, node 1 will remain congested and advertises its fair rate to all upstream nodes.

### Effect of $\beta$ on Fair Rate Oscillations

Figure 4.11 illustrates the effect of the ramping coefficient on convergence of the fair rate in a scenario with  $n = 8$ ,  $\alpha = 0.1$ ,  $d = 0.05 \cdot T$ ,  $\tau_\ell = 0.8$ ,  $\tau_h = 0.9$ , for  $\beta = 0.05$ , 0.15, and 0.25. It can be seen that analytical and simulation results are very close. When congestion is detected at  $\hat{k} = 16$ , the fair rate of node 1 is initially set to 0.5. The fair rate is then ramped down until  $r_1^u(k)$  becomes less than  $\tau_h$ . For  $\beta = 0.05$ , the fair rate is ramped down with a low rate until it converges. As  $\beta$  increases, the

fair rate decreases with a higher rate and, consequently,  $r_1^u(k)$  may fall below  $\tau_\ell$ . In this case, the fair rate is ramped up. However, when  $\beta$  is relatively large, the fair rate is ramped up with a high rate and, as a result,  $r_1^u(k)$  may hit  $\tau_h$  again. In fact, when  $\beta$  is large, the fair rate may be overestimated and underestimated frequently. This can be observed for  $\beta = 0.25$ , where the fair rate never converges and instead oscillates around  $\frac{1}{n}$ . In general, as  $\beta$  decreases, the fair rate exhibits less oscillations at the expense of longer convergence times.

### Effect of Propagation Delay on Stability of Fair Rate

Figure 4.12 illustrates the effect of the link propagation delay on convergence of the fair rate in a scenario with  $n = 5$ ,  $\alpha = 0.1$ ,  $\beta = 0.3$ ,  $\tau_\ell = 0.8$ , and  $\tau_h = 0.9$  for  $d = 0.02 \cdot T$  and  $0.1 \cdot T$ . These results are obtained from our analytical model. One can see that for  $d = 0.02 \cdot T$ , the fair rate converges after being ramped down in several control intervals. However, the fair rate exhibits permanent oscillations for  $d = 0.1 \cdot T$ . The reason is that when the link propagation delay exists, the upstream nodes of node 1 receive the control message with delay and do not adjust their add-rates at the same time. As a result, the link usage rate at node 1, which is the aggregate add-rate of all nodes, gradually changes during a control interval. In order to compute the new fair rate, node 1 calculates  $r_1^u(k)$  which is the average usage rate of its output link during control interval  $k$ . According to Inequality (4.2.10), only in time interval  $((k-1) \cdot T + (n-1) \cdot d, k \cdot T]$  of control interval  $k$ , the aggregate add-rate of all nodes observed at node 1 corresponds to the latest advertised fair rate. In other words, as  $d$  increases, the weight of the latest advertised fair rate in calculation of  $r_1^u(k)$  decreases; hence,  $r_1^u(k)$  does not accurately estimate the aggregate add-rate of all nodes corresponding to the latest advertised fair rate. As a result, the fair rate may not be adjusted properly when  $d$  is relatively large. Note that fair rate oscillations can be removed by decreasing  $\beta$  when  $d$  is relatively large. It can be seen in Figure 4.12

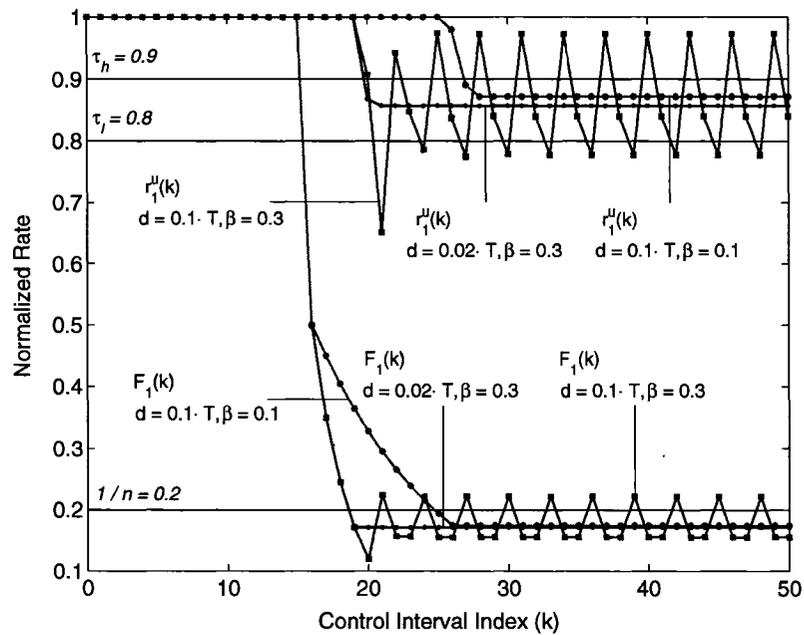


Figure 4.12: Effect of  $d$  on convergence of fair rate

that the fair rate does not oscillate when  $\beta$  is decreased to 0.1 while  $d = 0.1 \cdot T$ . The reason is that when the ramping coefficient is decreased, the fair rate is ramped down or ramped up with a low rate which means that the latest advertised fair rate is close to the previous one. Therefore,  $r_1^u(k)$  is a better estimation of the aggregate traffic of all nodes corresponding to the advertised fair rate and the fair rate is adjusted more accurately. It seems that for a fixed  $d$ , there is a maximum value of  $\beta$  for which the fair rate does converge. This can be a very interesting problem to explore the relation between other system parameters and the maximum value of  $\beta$  for which the fair rate does not oscillate.

### Effect of $\tau_\ell$ and $\tau_h$ on Stability of Fair Rate

We observed in previous sections that when  $\beta$  is relatively large, the fair rate may never converge. The oscillations result from ramping the fair rate with high rates such

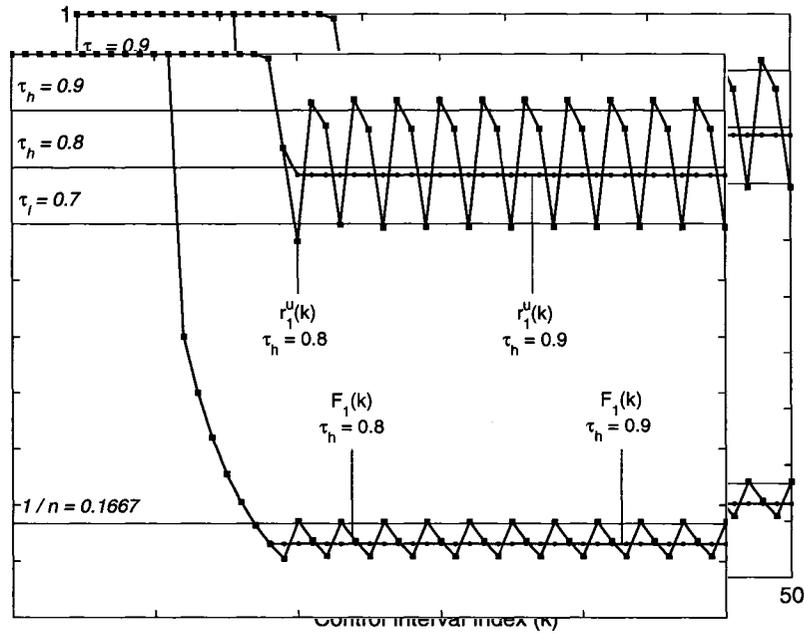


Figure 4.13: Effect of  $\tau_h - \tau_\ell$  on convergence of fair rate

that  $r_1^u(k)$  cannot be held between  $\tau_\ell$  and  $\tau_h$ . Figure 4.13 presents  $F_1(k)$  and  $r_1^u(k)$  obtained from the analytical model in a scenario with  $n = 6$ ,  $d = 0.1 \cdot T$ ,  $\alpha = 0.1$ ,  $\beta = 0.3$ ,  $\tau_\ell = 0.7$ , for  $\tau_h = 0.8$  and  $0.9$ . In this scenario  $\beta$  is chosen relatively large and, as a result, the fair rate changes with a high rate when ramped up or ramped down. For  $\tau_h = 0.8$ , the fair rate has permanent oscillations and  $r_1^u(k)$  oscillates around  $\tau_h$  and  $\tau_\ell$ . However, when  $\tau_h$  is increased to  $0.9$ , the fair rate converges without oscillations. In general, when  $\tau_h - \tau_\ell$  increases, the chances that  $r_1^u(k)$  could be held between  $\tau_h$  and  $\tau_\ell$  increase, specially when the ramping coefficient is relatively large. The oscillations can also be removed by decreasing  $\tau_\ell$ . However, recall that the fair rate may converge to any value in interval  $[\tau_\ell/n, \tau_h/n]$  (when convergence does exist) and, as a result,  $r_1^u(k)$  can be as low as  $\tau_\ell$  at the steady-state condition. In other words, reducing  $\tau_\ell$  can decrease the achieved throughput although it may remove the fair rate oscillations.

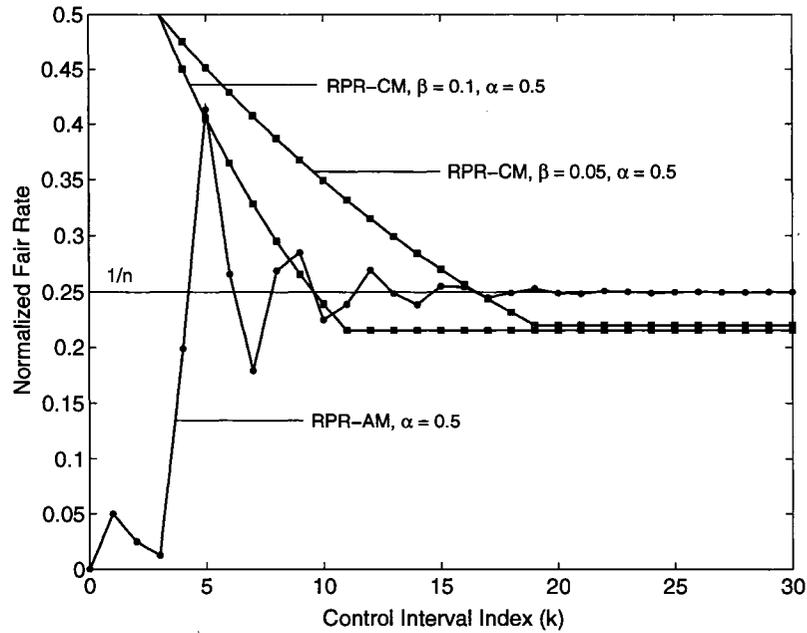


Figure 4.14: Comparison between the RPR-AM and the RPR-CM

### Comparison with Aggressive Mode

In this section, we compare the fair rate calculations in RPR-CM with the one in RPR-AM. Figure 4.14 presents the calculated fair rate of node 1 with RPR-AM and RPR-CM for  $n = 4$ ,  $d = 0.2 \cdot T$ ,  $\alpha = 0.5$ ,  $\tau_h = 0.9$ ,  $\tau_\ell = 0.8$ , and  $\beta = 0.05$ , and  $0.1$ . The calculated fair rate of both algorithms is obtained from analytical results presented in this chapter.

As  $\tau_\ell = 0.8$  and  $\alpha = 0.5$ , congestion is detected at  $\hat{k} = 3$  according to Equation (4.1.4). In RPR-CM, the initial fair rate of the congested node 1 is set to 0.5 and, then, gradually ramped down until converges. However, in RPR-AM, the fair rate is set to the low-pass filtered of the add-rate of node 1 and exhibits some oscillations before converging to  $\frac{1}{n}$ . In RPR-CM, however, the fair rate is set to 0.5 at the time of congestion detection. It is then gradually ramped down and converges without oscillations for both values of  $\beta$ .

A difference between RPR-AM and RPR-CM in this scenario is in the achieved throughput when the fair rate does converge. When convergence exists, the fair rate converges to  $\frac{1}{n}$  in RPR-AM and, as a result, the output link of the congested node 1 is fully utilized. However, the fair rate in RPR-CM may converge to any value between  $\tau_\ell/n = 0.2$  and  $\tau_h/n = 0.225$  according to Remark 4.2.1. Therefore, the output link of the congested node 1 is not fully utilized in RPR-CM. This can be seen in Fig 4.14 where the fair rate in RPR-CM converges to a value less than  $\frac{1}{n} = 0.25$  for both values of  $\beta$  whereas in RPR-AM the fair rate converges to 0.25.

### 4.3 Discussion on Generalization of Analyses

In this chapter, we analyzed the fair rate calculation process in the RPR standard bandwidth allocation algorithms for the parking-lot scenario, which is the most popular benchmark scenario considered for performance evaluation of RPR bandwidth allocation algorithms (see [4–9, 68, 69]). Although these analyses were performed for the parking-lot scenario, they can apply to the scenarios which are viewed as a parking-lot scenario. In fact, the parking-lot scenario represents a class of scenarios in which a set of nodes compete for bandwidth on a link. (This is also one of the main reasons that the parking-lot scenario is considered as a benchmark scenario in RPR.) For instance, consider the parking-lot scenario depicted in Figure 4.1 and assume that the destination node for some of the flows is beyond node 0. This scenario would be equivalent to the parking-lot scenario and analysis of such a scenario would be the same as the one for the parking-lot scenario. Another example is a scenario in which some of the nodes have multiple flows passing through node 1 and destined to nodes beyond node 0. This scenario is also equivalent to the parking-lot scenario. A special case of such a scenario is referred to as the two-exit parking-lot scenario is introduced in Section 3.4.4.

Further, consider the parking-lot scenario depicted in Figure 4.1 and assume that node  $n$  has another flow destined to node  $n - 1$ . This is, in fact, a parallel parking-lot scenario which was introduced in Section 3.4.4 and also considered for performance evaluation of RPR fairness algorithms in [4, 6]. The developed model for RPR-AM in the parking-lot scenario can apply to the parallel parking-lot scenario as well. The reason is that the short-flow of node  $n - 1$  would not compete with other flows for the bandwidth on the link between nodes 1 and 0; hence, it does not have any effect on the calculated fair rate of node 1.

## 4.4 Conclusion

In this chapter, we modeled the fair rate calculation process in the RPR standard bandwidth allocation algorithms (i.e. RPR-AM and RPR-CM) in the parking-lot scenario. The fair rate equations were derived as a function of system parameters such as the number of nodes, the low-pass filtering coefficient, and the link propagation delay for both algorithms. These models can apply to a class of scenarios which are viewed as a parking-lot scenario. The developed models are accurate analytical models which can be used to study the dynamics of the RPR standard bandwidth allocation algorithms. For RPR-AM, we studied the effect of different system parameters on fair rate calculations through the developed model. We observed that the model with propagation delay is stable (i.e. the fair rate converges) for a wider range of the low-pass filtering coefficient when compared to the ideal system without propagation delay. We also observed that the range of the low-pass filtering coefficient for which the fair rate converges, is maximum when the length of the control interval is approximately twice the round-trip propagation delay of the system. We also showed that if the propagation delay is unknown for an  $n$ -node scenario, the low-pass filtering coefficient should be set to a value which is greater than  $\frac{1}{n}$  and not very close to  $\frac{2}{n}$ .

to ensure that the convergence time is relatively short. We then analyzed the fair rate calculation process in RPR-CM. We studied the impact of various system parameters on convergence of the fair rate through analytical results. We observed that decreasing the ramping coefficient can mitigate the fair rate oscillations at the expense of longer convergence times. It was also shown that rate threshold values affect the fair rate oscillations and the achieved throughput in RPR-CM. Furthermore, we compared the performance of RPR-CM and RPR-AM in terms of fair rate stability and achieved throughput. We observed that RPR-AM favors achieved throughput over fair rate stability. On the other hand, RPR-CM provides highly dampened fair rate adjustments which favors fair rate stability over achieved throughput.

## Chapter 5

# Virtual Queuing: An Efficient Bandwidth Allocation Algorithm for Resilient Packet Rings

The RPR standard bandwidth allocation algorithms suffer from a number of performance limitations. Specifically, RPR-AM is prone to permanent fair rate oscillations even in some simple unbalanced traffic scenarios [6]. RPR-CM is more robust to unbalanced traffic scenarios, however, it suffers from slow convergence times and lower throughput compared to RPR-AM. The main reasons behind the performance limitations of these algorithms are twofold. First, the fair rate calculated at a congested node in RPR-AM and RPR-CM does not accurately estimate the true fair rate of that node. Therefore, the fair rate may exhibit oscillations before converging to the true fair rate. Second, the control parameters in these algorithms are low-pass filtered in order to mitigate oscillations. However, such a filtering results in delayed congestion detection and slow convergence of the fair rate.

There have been a number of efforts to improve the performance of the RPR standard bandwidth allocation algorithms [49, 50]. However, these proposals only prevent the fair rate oscillations in some specific scenarios and do not improve the performance of these algorithms in general. In [6], a bandwidth allocation algorithm is

proposed for RPR called Distributed Virtual-time Scheduling in Rings (DVSR) which outperforms RPR-AM and RPR-CM in terms of convergence times and achieved throughput. However, we show that DVSR can result in extreme unfairness in some dynamic traffic scenarios.

In this chapter, we propose an efficient bandwidth allocation algorithm for RPR called Virtual Queueing (VQ). The convergence speed of VQ is in the same range as that of DVSR while it has remarkably better fairness characteristics in a dynamic traffic environment. In addition, VQ has a lower computational complexity when compared to DVSR.

This chapter is organized as follows. We first review the DVSR algorithm and evaluate its performance in various traffic scenarios. We then highlight some performance limitations of the DVSR algorithm in a dynamic traffic environment. Afterwards, we introduce the VQ algorithm and characterize its fairness properties. Finally, we evaluate the performance of VQ in different scenarios through simulation results.

## 5.1 DVSR Algorithm

The main goal in the DVSR algorithm is to maintain max-min fairness among IA-flows at each congested node. In order to calculate the fair rate, congested node  $n$  measures the rate of all IA-flows  $IA(i, n)$ ,  $i \leq n$ , during each control interval.<sup>1</sup> Then, the max-min fair rate (see Section 3.3.1) for the set of the observed rates of IA-flows is computed. In other words, it is implicitly assumed in DVSR that the rate of IA-flow  $IA(i, n)$  represents the demand of node  $i$  for bandwidth at node  $n$ . Let us define  $r_i^n(k)$  as the rate of IA-flow  $IA(i, n)$  in control interval  $k$ . Furthermore, assume that  $F_n(k)$  is the fair rate of node  $n$  calculated at the end of control interval  $k$ . To calculate the

---

<sup>1</sup>Note that the rate of IA-flow  $IA(n, n)$  can be more than the service rate of local traffic at node  $n$  as explained in Section 3.3.2.

Table 5.1: DVSR fair rate calculation

<pre> if (<math>\sum_j \dot{r}_j &lt; C</math>)   <math>F_n(k) = \dot{r}_N + (C - \sum_j \dot{r}_j)</math> else{   <math>j = 1</math>   <math>F_n(k) = C/N</math>   Cap = C   while (<math>\dot{r}_j &lt; F_n(k)</math> AND <math>\dot{r}_N \geq F_n(k)</math>){     Cap = Cap - <math>\dot{r}_j</math>     <math>F_n(k) = \text{Cap}/(N - j)</math>     <math>j = j + 1</math>   } }</pre>
---

fair rate in DVSR, the set of measured rates of IA-flows,  $\{r_i^n(k), i \leq n\}$ , is sorted in an increasing order to obtain the set of rates  $\{r_1, r_2, \dots, r_N\}$ , where  $r_1 \leq r_2 \leq \dots \leq r_N$  and  $N$  is the number of IA-flows competing at node  $n$ . The fair rate of node  $n$  at the end of control interval  $k$  is then computed according to the pseudo code in Table 5.1. One can notice that the following cases can occur at node  $n$ :

- When  $\sum_j \dot{r}_j < C$  the max-min fair rate of the IA-flows at node  $n$  is equal to  $\Phi(C, \{r_i^n(k), \forall i\}) = \max_i \{r_i^n\} = \dot{r}_N$ . However, the fair rate of node  $n$  is set to  $F_n(k) = \dot{r}_N + (C - \sum_j \dot{r}_j)$ . Note that if the unused bandwidth, i.e.  $(C - \sum_j \dot{r}_j)$  was not added to the max-min fair rate of the observed IA-flows, it could not be reclaimed as all rates would be limited to  $\max_i \{r_i^n\}$ .
- When  $\sum_j \dot{r}_j \geq C$ , the pseudo-code calculates the max-min fair rate of the observed IA-flows at node  $n$ . Hence,  $F_n(k) = \Phi(C, \{r_i^n(k), \forall i\})$ .

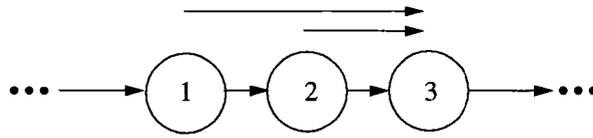


Figure 5.1: A two-node parking-lot scenario

In other words, the DVSR fair rate calculated at node  $n$  in control interval  $k$  can be formulated as

$$F_n(k) = \Phi(C, \{r_i^n(k), \forall i\}) + (C - \sum_{i \leq n} r_i^n(k))^+. \quad (5.1.1)$$

Moreover, by comparing Equations (5.1.1) and (3.3.6) we can conclude that the DVSR fair rate at node  $n$  is obtained as  $F_n(k) = \Phi^+(C, \{r_i^n(k), \forall i\}, \eta = 1)$ . Note that in order to compute the fair rate in DVSR, a sort operation with complexity of  $O(N \log N)$  is required.

### 5.1.1 Performance Evaluation of DVSR

In this section, we compare the performance of the DVSR algorithm with that of RPR-AM and RPR-CM through simulation results. First, we show that even in a simple static traffic scenario, RPR-AM and RPR-CM exhibit oscillations and/or slow convergence times when compared to DVSR. Then, we study a dynamic traffic scenario in which DVSR does not have a satisfactory performance.

#### Static Traffic Scenario

We consider a 2-node parking-lot scenario depicted in Figure 5.1, in which nodes 1 and 2 have constant traffic to send to node 3 with rates  $C$  and  $0.1 \cdot C$ , respectively. In other words,  $d_{13} = C$  and  $d_{23} = 0.1 \cdot C$ . In this scenario, we assume that  $C = 100\text{Mbps}$ ,  $lpCoef = 16$ ,  $rampUpCoef = rampDnCoef = 64$ ,  $\tau_\ell = 0.85 \cdot C$ ,  $\tau_h = 0.95 \cdot C$ , and  $T = 1\text{msec}$ .

The simulation results for the achieved throughput of nodes 1 and 2 are shown in Figures 5.2 to 5.4 for RPR-AM, RPR-CM, and DVSR, respectively. It can be seen in Figure 5.2 that the throughput of upstream node 1 has permanent oscillations in RPR-AM. The reason is that when congestion is detected, node 2 advertises the low-pass filtered of its add-rate which is close to  $0.1 \cdot C$ . Consequently, node 1 decreases its add-rate to the fair rate of node 2. This, in turn, clears congestion at node 2 and, hence, node 1 will be allowed to increase its add-rate gradually until node 2 becomes congested again. This process repeats and, as a result, the rate of node 1 oscillates permanently. The throughput loss in this case is around 36%. It can be seen in Figure 5.3 that RPR-CM is more robust to unbalanced traffic in this scenario compared to RPR-AM. When congestion is detected, the fair rate of node 2 is set to  $0.5 \cdot C$  when RPR-CM is employed. The fair rate is then ramped up gradually until the link usage rate at node 3 becomes greater than  $\tau_\ell$  and less than  $\tau_h$ . The throughput loss in this case is around 14%. Although the fair rate does not oscillate when RPR-CM is employed, its convergence time is relatively large and it does not fully utilize the available capacity. As it can be seen in Figure 5.4, DVSR outperforms RPR-AM and RPR-CM in terms of convergence time and achieved throughput. The fair rate in DVSR is accurately calculated and, as a result, it does not exhibit oscillations and speedily converges. In fact, convergence times of DVSR are an order of magnitude faster than that of RPR-AM and RPR-CM. Furthermore, the available bandwidth is fully utilized when DVSR is employed.

### Dynamic Traffic Scenario

In this section, we evaluate the performance of RPR-AM, RPR-CM, and DVSR algorithms through simulation results in dynamic traffic scenarios. In order to create traffic variations in the ring, we define a source model called *dynamic source*, which

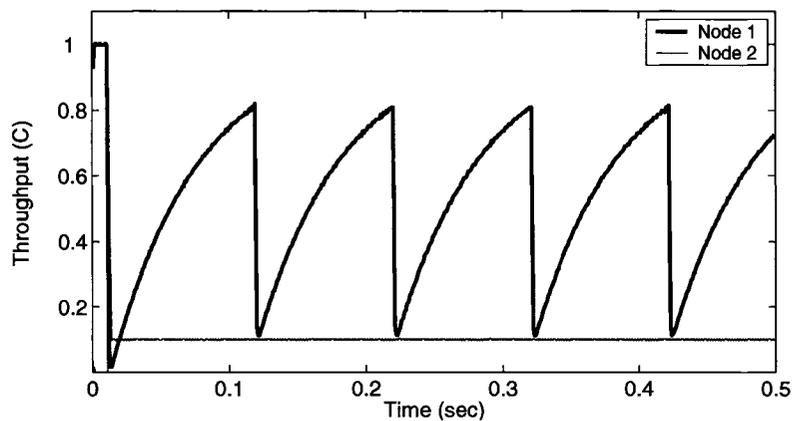


Figure 5.2: Throughput of the nodes in an unbalanced static scenario in RPR-AM

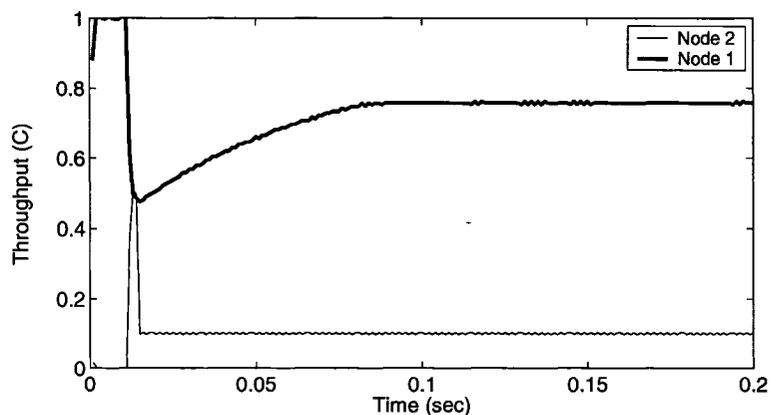


Figure 5.3: Throughput of the nodes in an unbalanced static scenario in RPR-CM

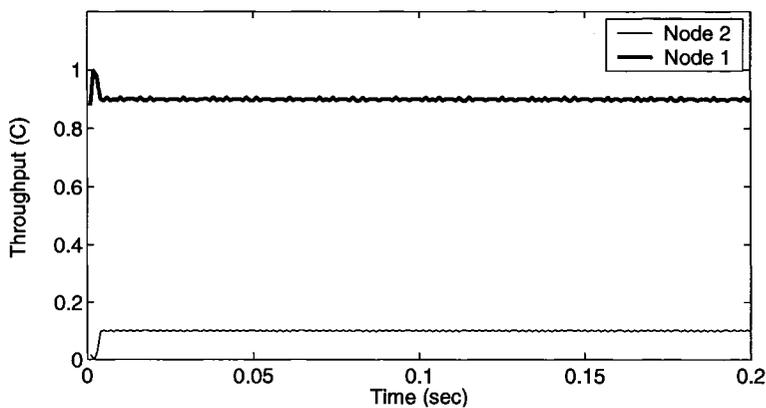


Figure 5.4: Throughput of the nodes in an unbalanced static scenario in DVSR

periodically fluctuates between two states: high-state and low-state. In the high-state, which lasts for  $T_H$  seconds, the dynamic source generates traffic with average rate  $\delta_H$ . Following the high-state, the dynamic source enters the low-state and generates traffic with average rate  $\delta_L$  ( $\delta_L < \delta_H$ ) for  $T_L$  seconds. After the low-state, the dynamic source switches to the high-state again. In the following, we assume that all dynamic sources generate traffic with constant rate at each state.

To evaluate the performance of RPR-AM in a dynamic traffic environment, we consider a 2-node parking-lot scenario (see Figure 5.1), where the dynamic source is introduced at node 2 (head-node) and node 1 can send traffic at rate  $C$ . For the dynamic source we have  $T_L = T_H = 10 \cdot T$ ,  $\delta_H = 0.5 \cdot C$ , and  $\delta_L = 0.05 \cdot C$ . Figure 5.5 shows the throughput of node 1 and the total ring throughput in this scenario. One can see that when congestion is detected, the add-rate of node 1 drops to the fair rate which is the low-pass filtered value of the add-rate of node 2. When the dynamic source enters the low-state, congestion clears at node 2 and the add-rate of node 1 starts increasing. However, before the add-rate of node 1 is fully ramped up, the dynamic source enters the high-state and, consequently, node 2 gets congested again. As the add-rate of node 1 ramps up very slowly, the total achieved throughput is low and the throughput loss is around 25% in this scenario.

We considered the same 2-node parking-lot scenario (as described above) to evaluate the performance of RPR-CM in a dynamic traffic environment. In Figure 5.6, the total ring throughput and the throughput of node 1 are presented. When the dynamic-source enters the low-state, the fair rate of node 2 is ramped up. However, when the dynamic source enters the high-state, the link usage at node 2 exceeds  $\tau_h$  and, as a result, the fair rate is ramped down. In other words, when RPR-CM is employed, congestion does not clear at node 2 in this scenario as its fair rate can not be fully ramped up. The throughput loss in this case is around 22.5%

In order to evaluate the performance of DVSR in a dynamic traffic environment,

we consider a 4-node parking-lot scenario where the dynamic source is introduced at node 1, i.e. the most upstream node. We assume that  $d_{25} = d_{35} = C$  and  $d_{45} = 0.25 \cdot C$ . The dynamic source parameters are as follows:  $T_L = T_H = T$ ,  $\delta_H = 0.25 \cdot C$ ,  $\delta_L = 0.1 \cdot C$ . Assume that the dynamic source at node 1 is in the high-state. In this case, the fair rate of node 4 is equal to  $0.25 \cdot C$ . When the dynamic source enters the low-state, there will be  $0.15 \cdot C$  unused bandwidth available at link  $l(4, 5)$  as the rate of node 4 drops from  $0.25 \cdot C$  to  $0.1 \cdot C$ . As a result, the fair rate of node 4 is increased by  $0.15 \cdot C$  according to Equation (5.1.1). Consequently, nodes 2 and 3 increase their add-rates in the next control interval. However, the dynamic source of node 1 enters the high-state in the same control interval and starts sending traffic at rate  $0.25 \cdot C$ . Hence, there is not enough bandwidth available for node 4 to add its local traffic to the ring as nodes 2 and 3 have already increased their add-rates. The achieved throughput of nodes 2, 3, and 4 is depicted in Figure 5.7. One can see that as the dynamic source at node 1 fluctuates, the throughput of node 4 periodically drops. The average throughput of node 4 in this case is  $0.169 \cdot C$ . Note that as  $d_{45} = 0.25 \cdot C$ , and there are four IA-flows competing for bandwidth on link  $l(4, 5)$ , the fair share of bandwidth for node 4 is equal to  $0.25 \cdot C$ . In other words, node 4 is being starved in this scenario as its achieved throughput is substantially less than its fair share.

The main reason behind the starvation of node 4 in the above scenario is the overestimation of the fair rate in DVSR. That is, when some of the nodes in a congestion span decrease their rates, the new fair rate of the head-node is calculated by adding the unused capacity to the old fair rate (see Table 5.1). In this case, all upstream nodes in the congestion span that were limited to the previous fair rate, can increase their rates to the new fair rate. As a result, the downstream nodes may face starvation as the fair rate is not properly estimated. Hence, the fair rate should

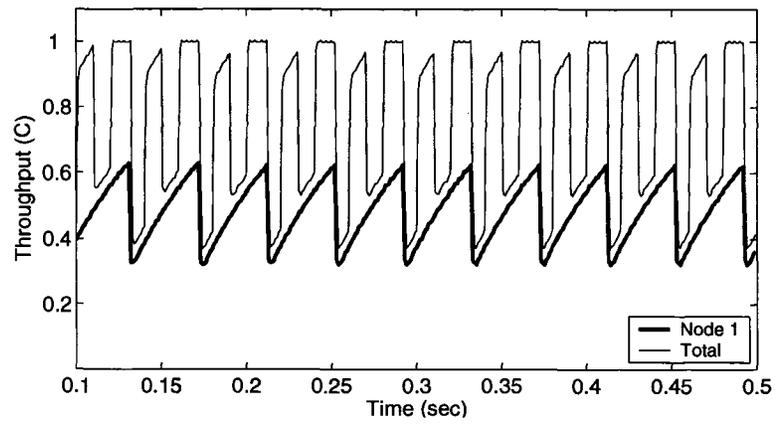


Figure 5.5: Performance of RPR-AM in a dynamic traffic scenario

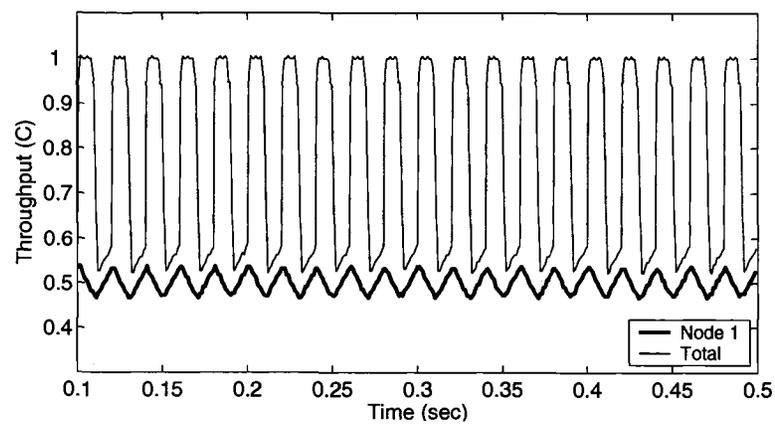


Figure 5.6: Performance of RPR-CM in a dynamic traffic scenario

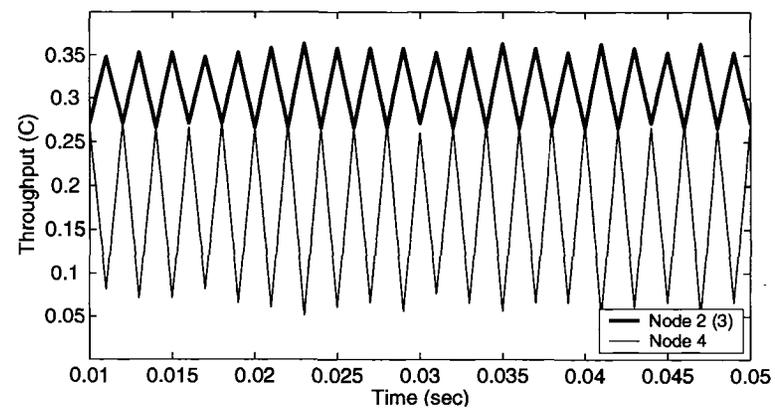


Figure 5.7: Performance of DVSR in a dynamic traffic scenario

be re-adjusted to maintain fairness among all nodes. The impact of fair rate overestimation on starvation of downstream nodes can be significant in a dynamic traffic environment where traffic of each node may frequently change.

## 5.2 Virtual Queuing Algorithm

The RPR standard bandwidth allocation algorithms (RPR-AM and RPR-CM) are particularly attractive for their remarkably simple implementation. However, these algorithms do not exhibit a satisfactory performance in terms of convergence times and achieved throughput. On the other hand, DVSR outperforms RPR-AM and RPR-CM at the expense of a complex implementation. Nonetheless, in the previous section we observed that in some dynamic traffic scenarios, the fair rate overestimation in DVSR may result in starvation of downstream nodes. In this section, we propose a new bandwidth allocation algorithm for RPR which is called Virtual Queuing (VQ). It aims at achieving fairness, high-utilization, and fast convergence times (in the same order of magnitude as DVSR) with a relatively simple implementation. Furthermore, VQ adjusts the fair rate more accurately (compared to DVSR) to prevent starvation of downstream nodes in dynamic scenarios.

### 5.2.1 Model of Rate Control System in VQ

In general, the rate control mechanism in RPR resembles a distributed feedback control system. That is, the add-rate of upstream nodes in a congestion span is controlled by the fair rate calculated at the congested node. At each node, the fair rate is computed based on traffic measurements at that node and, then, it is advertised to upstream nodes. When a node is starved and cannot add traffic to the ring, it becomes congested and decreases its fair rate. In this case, upstream nodes reduce their add-rates and, consequently, the congested node can add traffic to the ring. On

the other hand, when unused capacity is available at a node, its fair rate is increased. As a result, upstream nodes are allowed to add more traffic to the ring to reclaim the unused capacity.

In order to achieve high bandwidth utilizations while maintaining fairness, the fair rate of each node should be calculated accurately based on the demand of all nodes sending traffic through that node. However, the exact information about the demand of other nodes is not available at each node in RPR. Instead, the observed rate of an IA-flow transiting a congested node is used as an approximation to the demand of the source node of that IA-flow. As a result, the fair rate calculated based on this approximate demand information will be an estimation of the true fair rate. Hence, several iterations may be needed for the bandwidth allocation algorithm to adjust the fair rate and converge to the true fair rate. A more accurate fair rate estimation results in a faster convergence to the true fair rate. Similar to DVSR, the prime goal in VQ is to maintain max-min fairness among IA-flows at each node and achieve a RIAMM-fair solution throughout the ring. Hence, a max-min fair rate should be estimated at each node and then advertised to other nodes. In VQ, the max-min fair rate at each node is approximated using a simple scheme in each control interval. It is then re-adjusted according to the new measurements in the succeeding control intervals. Therefore, similar to DVSR, we expect to achieve fast convergence times (as the fair rate is estimated accurately at each node in VQ) and high bandwidth utilizations while maintaining fairness among IA-flows.

In VQ, each node is modeled as a virtual queue with service rate  $C$ . Figure 5.8 depicts a congestion span where each node is represented by its virtual queue model. Consider the virtual queue model of an arbitrary node  $n$  in a congestion span. The arrival traffic to this virtual queue consists of the transit traffic and the local traffic of node  $n$ . Assume that  $A^n(k)$  denotes the average rate of the arrival traffic to the

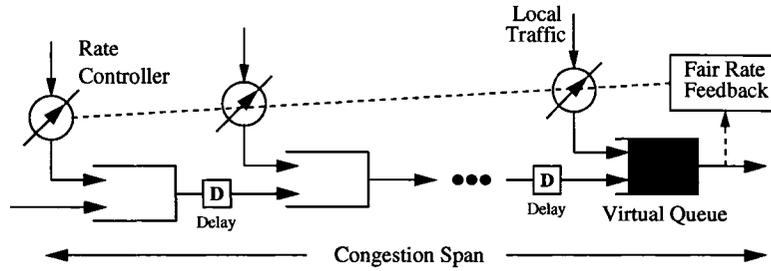


Figure 5.8: A virtual queue model for rate control mechanism in RPR

virtual queue of node  $n$  in control interval  $k$ . Hence,

$$A^n(k) = \sum_{i \leq n} r_i^n(k) = \sum_{i < n} r_i^n(k) + r_n^n(k), \quad (5.2.1)$$

where the first term (respectively the second term) in the right-hand side of this equation represents the average arrival rate of transit traffic (respectively local traffic) to the virtual queue at node  $n$  in control interval  $k$ . Note that  $r_n^n(k)$  is the allocated rate of local traffic at node  $n$  in control interval  $k$  which may be serviced or buffered in the virtual queue (Similar to what was explained in Section 3.3.2). We also have  $\sum_{i < n} r_i^n(k) \leq C$ , i.e. the rate of transit traffic cannot be more than  $C$ . When  $A^n(k) \leq C$ , the virtual queue is stable and there is enough capacity for node  $n$  to add its traffic to the ring. However, when  $A^n(k) > C$ , the local traffic of node  $n$  is not fully served and partially buffered in the virtual queue. In this case, the fair rate of node  $n$  should be reduced such that the virtual queue becomes stable, i.e.  $A^n(k) \leq C$ .

In the following proposition, we show that in a steady-state condition, the stability of all virtual queues is the necessary and sufficient condition for achieving a RIAMM-fair solution throughout the ring. First, we state the following remark.

*Remark 5.2.1.* In Section 3.3.1, we showed that when unused bandwidth exists at an arbitrary node  $n$ , the fair rate of node  $n$  should be set to a value which is higher than the allocated rate of all IA-flows at that node. That is, if  $\sum_{i < n} r_i^n(k) < C$ , we have  $F_n(k) > \max_{i \leq n} \{r_i^n(k)\}$ . In fact, this is the necessary condition for achieving a

RIAMM-fair solution and we assume that all bandwidth allocation algorithms realize this condition.

**Proposition 5.2.1.** *At steady-state conditions, RIAMM fairness is achieved if and only if all virtual queues are stable.*

*Proof.* For sufficiency, if all virtual queues are stable, for an arbitrary node  $n$  we have

$$A^n(k) (= \sum_{i \leq n} r_i^n(k)) \leq C.$$

i) If  $A^n(k) < C$ , we have  $F_n(k) > \max_{i \leq n} \{r_i^n(k)\} = \Phi(C, \{r_i^n(k)\})$  as this is the necessary condition for achieving a RIAMM-fair solution throughout the ring. Since the network in the steady state condition, node  $n$  is not a bottleneck node for any IA-flow and its fair rate can be set to any value greater than  $\max_{i \leq n} \{r_i^n(k)\}$ . Hence, it can be assume that  $F_n(k) = \max_{i \leq n} \{r_i^n(k)\} + \eta \cdot (C - \sum_{i \leq n} r_i^n(k))$ ,  $0 < \eta < 1$ , or equivalently,  $F_n(k) = \Phi^+(C, \{r_i^n(k)\}, \eta)$ .

ii) If  $A^n(k) = C$ , we have  $F_n(k) = r_{\max} = \max_{i \leq n} \{r_i^n(k)\}$ . In this case,  $F_n(k) = \frac{1}{|\{r_i^n(k)\}| - |Z|} \cdot (C - \sum_{\forall i \in Z} r_i^n(k))$ , where  $Z = \{i : 0 < r_i^n(k) < r_{\max}\}$ . In other words,  $F_n(k) = \Phi(C, \{r_i^n(k)\}) = \Phi^+(C, \{r_i^n(k)\}, \eta)$ ,  $0 < \eta < 1$  according to Equation (3.3.3).

Therefore, Inequality (3.3.5) and Equation (3.3.6) are satisfied simultaneously when

$$\sum_{i \leq n} r_i^n(k) \leq C \text{ and a RIAMM fair solution is achieved.}$$

The proof of necessity is straight forward as when RIAMM fairness is maintained throughout the ring, the aggregation of allocated bandwidth among IA-flows is equal to or less than  $C$ . □

The important ramification of this proposition is that a RIAMM-fair allocation can be achieved by stabilizing the virtual queues and reaching a steady-state condition in the network. In DVSR, the virtual-queues are stabilized by calculating explicitly the max-min fair rate at each node. In VQ, however, the max-min fair rate is estimated at each node by calculating a rate increase/decrease factor (according to the measured

rates of IA-flows) and adjusting the fair rate such that the virtual queue at that node becomes stable.

### 5.2.2 VQ Algorithm Description

Assume that at the end of control interval  $k - 1$ , node  $n$  has calculated its fair rate, i.e.  $F_n(k - 1)$ , and advertised to its upstream nodes. During control interval  $k$ , node  $n$  measures the rate of IA-flows entering its virtual queue. Note that the rate of IA-flows at node  $n$  are always conformed to the advertised fair rate of node  $n$ . There are two types of IA-flows observed at node  $n$  according to their rates: *rate-limited* and *input-limited*. The IA-flow  $IA(i, n)$  is considered rate-limited (respectively, input-limited) in control interval  $k$  when its rate is equal to (respectively, less than) the fair rate of node  $n$ , i.e.  $F_n(k - 1)$ .

When unused capacity is available at node  $n$ , the fair rate of node  $n$  should be increased. In this case, the input-limited IA-flows are expected to be input-limited with respect to a larger fair rate as well. In other words, the unused capacity should only be allocated to the rate-limited IA-flows (if any), which are restricted to the advertised fair rate of node  $n$ . In VQ, rate-limited flows are identified at each node and the fair rate is adjusted accordingly. Let us define the following notations for an arbitrary node  $n$  (the node subscript is omitted for simplicity):

- ▶  $Q_i(k)$ ,  $i \leq n$ , is the amount of traffic (in bits) from IA-flow  $IA(i, n)$  entering the virtual queue of node  $n$  measured during control interval  $k$  and we have  $Q_i(k) = r_i^n(k) \cdot T$ .
- ▶  $E_i(k)$  is the amount of eligible traffic of IA-flow  $IA(i, n)$  in control interval  $k$  conforming to the fair rate of node  $n$ . That is,  $E_i(k) = \min \{Q_i(k), F_n(k-1) \cdot T\}$ ,  $\forall i \leq n$ .

- ▶  $I = \{i : E_i(k) < F_n(k-1) \cdot T\}$  is the set of input-limited IA-flows entering the virtual queue of node  $n$ .
- ▶  $R = \{i : E_i(k) = F_n(k-1) \cdot T\}$  is the set of rate-limited IA-flows entering the virtual queue of node  $n$ .
- ▶  $E^I(k)$  is the total amount of eligible traffic from input-limited IA-flows arriving to the virtual queue of node  $n$ , i.e.  $E^I(k) = \sum_{i \in I} E_i(k) = \sum_{i \in I} Q_i(k) = \sum_{i \in I} r_i^n(k) \cdot T$ .
- ▶  $E^R(k)$  is the total amount of eligible traffic from rate-limited IA-flows arriving to the virtual queue of node  $n$ , i.e.  $E^R(k) = \sum_{i \in R} E_i(k) = |R| \cdot F_n(k-1) \cdot T$ .
- ▶  $f(k)$  is a rate increase/decrease factor calculated in control interval  $k$  ( $f(k) > 0$ ).

At the end of control interval  $k$ , node  $n$  calculates its rate increase/decrease factor based on the above measurements as

$$f(k) = \begin{cases} (C \cdot T - E^I(k)) / E^R(k), & \text{if } E^I(k) < C \cdot T \text{ and } E^R(k) > 0, \\ 1, & \text{if } E^I(k) < C \cdot T \text{ and } E^R(k) = 0, \\ C \cdot T / (E^R(k) + E^I(k)), & \text{if } E^I(k) \geq C \cdot T. \end{cases} \quad (5.2.2)$$

The fair rate of node  $n$  in control interval  $k$  is then given by

$$F_n(k) = \min \{C, f(k) \cdot F_n(k-1)\}. \quad (5.2.3)$$

The following observations can be made on the fair rate calculation of node  $n$  at the end of control interval  $k$ .

- Node  $n$  measures traffic of each IA-flow entering its virtual queue, i.e.  $Q_i(k)$ , which is used as an estimation for the demand of each node in the next control interval. For each node  $i \leq n$ ,  $Q_i(k)$  is equal to the rate of IA-flow  $IA(i, n)$  multiplied by  $T$ . Then, for each node  $i$ ,  $E_i(k)$ , which is the eligible part of traffic of IA-flow  $IA(i, n)$  conforming to the fair rate of node  $n$ , is computed.

According to the value of  $E_i(k)$ , node  $i$  is identified as a rate-limited node (which can send traffic at the current fair rate) if  $E_i(k) = F_n(k-1) \cdot T$ ; or it is an input-limited node (which sends traffic with a rate less than the current fair rate) if  $E_i(k) < F_n(k-1) \cdot T$ .

- It can be seen that when  $E^l(k) < C \cdot T$  and at least one rate-limited IA-flow exists (i.e.  $E^R(k) > 0$ ), the rate increase/decrease factor is given by  $f(k) = (C \cdot T - E^l(k)) / E^R(k)$  and one of the following cases occurs.
  - i) If  $E^R(k) + E^l(k) < C \cdot T$  (i.e. the amount of eligible arrival traffic to the virtual queue of node  $n$  is less than the amount of traffic that can be serviced), we have  $f(k) > 1$  and the fair rate is increased (i.e.  $F_n(k-1) < F_n(k)$ ). Assume that  $\delta$  is the unused bandwidth at node  $n$  and we have  $E^R(k) + E^l(k) + \delta \cdot T = C \cdot T$ . In this case, we have  $f(k) = (\delta \cdot T + E^R(k)) / E^R(k)$ . As  $E^R(k) = |R| \cdot F_n(k-1) \cdot T$ , we have  $F_n(k) = F_n(k-1) + \frac{1}{|R|} \cdot \delta$ . In other words, the new fair rate is computed based on the assumption that all rate-limited IA-flows can increase their rates to the new fair rate. Therefore, the the unused bandwidth at node  $n$  ( $\delta$ ) is equally distributed among rate-limited IA-flows.
  - ii) If  $E^R(k) + E^l(k) > C \cdot T$  (i.e. the amount of eligible arrival traffic to the virtual queue of node  $n$  exceeds the amount of traffic that can be serviced), we have  $f(k) < 1$ . This means that the virtual queue of node  $n$  is not stable and the fair rate should be decreased (i.e.  $F_n(k) < F_n(k-1)$ ). Assume that  $E^R(k) + E^l(k) = C \cdot T + \delta \cdot T$ . Similar to the previous case, we can show that the new fair rate is computed as  $F_n(k) = F_n(k-1) - \frac{1}{|R|} \cdot \delta$ . In other words, when the amount of eligible arrival traffic to the virtual queue of node  $n$  exceeds  $C \cdot T$  by  $\delta \cdot T$ , the new fair rate is approximated by reducing the rate of the rate-limited nodes equally by  $\frac{1}{|R|} \cdot \delta$ .
  - iii) If  $E^R(k) + E^l(k) = C \cdot T$ , the virtual queue is stable and the fair rate does

not change ( $f(k) = 1$ ).

- When  $E^l(k) < C \cdot T$  and all IA-flows at node  $n$  are input-limited ( $E^R(k) = 0$ ), the rate of each IA-flow is less than the fair rate. Moreover, the total arrival rate of the virtual queue of node  $n$  is less than its service rate. Therefore, the fair rate is not required to change and we have  $F_n(k) = F_n(k - 1)$  as  $f(k) = 1$ .
- When the total amount of eligible arrival traffic from input-limited nodes exceeds the amount of traffic that can be served in the virtual queue ( $E^l(k) > C$ ), the fair rate is very large and should be decreased. In this case, the fair rate is proportionally decreased to match the rate of arrival traffic with the service rate of the virtual queue, i.e.  $F_n(k) = F_n(k - 1) \cdot C \cdot T / (E^l(k) + E^R(k))$ .

The following example illustrates the fair rate calculations in VQ.

**Example 5.2.1.** Consider a 4-node parking-lot scenario where  $d_{15} = d_{25} = C$  and  $d_{35} = d_{45} = 0.2 \cdot C$ . In control interval  $k$ , when steady-state conditions are reached and all virtual queues are stable, we have  $F_4(k) = 0.3 \cdot C$ ,  $r_{15}(k) = r_{25}(k) = 0.3 \cdot C$ , and  $r_{35}(k) = r_{45}(k) = 0.2 \cdot C$ . Now assume that in control interval  $k + 1$ , node 3 increases its rate to  $0.3 \cdot C$ . Hence,  $r_1^4(k + 1) = r_2^4(k + 1) = r_3^4(k + 1) = 0.3 \cdot C$ , and  $r_4^4(k + 1) = 0.2 \cdot C$ . As the rate of arrival traffic of the virtual-queue at node 4 exceeds its service rate, the fair rate should be decreased. In this case,  $R = \{1, 2, 3\}$ ,  $I = \{4\}$ ,  $E_R = 3 \cdot F_4(k) \cdot T = 0.9 \cdot C \cdot T$ , and  $E_I = 0.2 \cdot C \cdot T$ . Therefore,  $f(k + 1) = \frac{8}{9}$  and  $F_4(k + 1) = \frac{0.8}{3} \cdot C$ . This means that the allocated rates of the nodes in control interval  $k + 2$  are  $r_{15}(k + 2) = r_{25}(k + 2) = r_{35}(k + 2) = \frac{0.8}{3} \cdot C$ , and  $r_{45}(k + 2) = 0.2 \cdot C$ .

Now assume that instead of increasing, the rate of node 3 drops to  $0.1 \cdot C$  in control interval  $k + 1$ . That is,  $r_1^4(k + 1) = r_2^4(k + 1) = 0.3 \cdot C$ ,  $r_3^4(k + 1) = 0.1 \cdot C$ , and  $r_4^4(k + 1) = 0.2 \cdot C$ . In this case, the arrival rate of traffic to the virtual queue of node 4 is less than its service rate and, consequently, the fair rate increases. We have

$R = \{1, 2\}$ ,  $l = \{3, 4\}$ ,  $E^R(k+1) = 2 \cdot F_4(k) \cdot T = 0.6 \cdot C \cdot T$ , and  $E^l(k+1) = 0.3 \cdot C \cdot T$ . Therefore,  $f(k+1) = \frac{7}{6}$  and  $F_4(k+1) = 0.35 \cdot C$ . In this case, the allocated rates of the nodes in control interval  $k+2$  are  $r_{15}(k+2) = r_{25}(k+2) = 0.35 \cdot C$ ,  $r_{35}(k+2) = 0.1 \cdot C$ , and  $r_{45}(k+2) = 0.2 \cdot C$ . Note that if DVSR was employed, the fair rate of node 4 would be set to  $0.4 \cdot C$  (instead of  $0.35 \cdot C$ ) in this case. As a result, nodes 1 and 2 would increase their rates to  $0.4 \cdot C$ . Consequently, the available bandwidth for node 4 in control interval  $k+2$  would be  $0.1 \cdot C$ , i.e. node 4 would be starved in control interval  $k+2$ . It would take another control interval for DVSR to adjust the fair rate of node 4 to  $0.35 \cdot C$ . ◀

*Remark 5.2.2.* In VQ, when unused capacity exists at the head-node of a congestion span, the fair rate is increased. In this case, assuming unchanged demands, the head-node of the congestion span can always utilize its fair share of bandwidth (i.e. it is not starved) with respect to fair rates calculated before the steady-state condition is reached. The reason is that in VQ, the fair rate is increased with respect to the number of rate-limited nodes which are potentially able to increase their add-rates. However, in DVSR, the head-node may face starvation as the fair rate is overestimated in this case.

*Remark 5.2.3.* The fair rate in VQ is calculated at each node such that the virtual queue of that node becomes stable. All nodes then limit the add-rate of their local traffic transiting a downstream node to the advertised fair rate of that downstream node. Hence, all virtual queues become stable and according to Proposition 5.2.1 a RIAMM-fair solution can be achieved.

### 5.2.3 Control Message Transmission

Similar to DVSR, VQ employs a proactive approach in order to resolve congestion throughout the ring. That is, all nodes calculate their fair rates at the end of each

control interval and advertise their rates to other nodes through a single control message circulating along the ring. The control message contains separate fields for the fair rate of each node. Upon receiving the fair rate, each node inserts its fair rate in the designated field and forwards the message to its upstream node. Each node also adjusts the rate of its local traffic such that its add-rate for destinations beyond each downstream node, does not exceed the advertised fair rate of that node.

#### 5.2.4 Complexity of VQ Algorithm

Similar to DVSR, in order to calculate the fair rate of an arbitrary node  $n$  in VQ we need to measure the rate of each IA-flow  $IA(i, n)$  in every control interval. The rate of IA-flow  $IA(i, n)$  is computed by measuring the number of bytes originating from node  $i$  transiting node  $n$ . Therefore, we need a separate byte counter for each source node at an arbitrary node  $n$ . Such byte counters can be easily implemented in hardware.

Assume that  $N$  is the maximum number of nodes competing with each other. The fair rate in DVSR is computed through a sort operation with complexity  $O(N \log N)$  and a loop with complexity  $O(N)$ . The implementation of a sort operation in hardware is extremely complex and it should be calculated in software. The overall complexity of DVSR algorithm is  $O(N \log N)$ .

However, VQ does not need a sort operation and the fair rate can be calculated by identifying the rate-limited and the input limited nodes. This operation is performed by comparing the rate of each IA-flow at the end of each control interval with the previous fair rate. The complexity of such an operation is  $O(N)$ . Furthermore, this operation can be easily implemented in hardware.

Note that the complexity of RPR-AM and RPR-CM is much less than that of VQ and DVSR. In RPR-AM the fair rate of node  $n$  is simply set to the low-pass filtered

version of the add-rate of node  $n$ . This means that there is no need to provide separate byte counters for each source. In RPR-CM, the fair rate is gradually ramped-up or down based on the link usage at the congested node. Therefore, the complexity of RPR-AM and RPR-CM is roughly  $O(1)$ .

### 5.2.5 Drawbacks of VQ Algorithm

One of the main design goals of VQ algorithm was to prevent starvation of the head-node by adjusting the fair rate more accurately when unused bandwidth exist. To achieve this goal, the fair rate is increased conservatively in VQ. This approach may result in longer convergence times and throughput loss compared to DVSR in some scenarios. The following example illustrates this issue further.

**Example 5.2.2.** Consider a 4-node parking-lot scenario where  $d_{15} = C$  and  $d_{25} = d_{35} = d_{45} = 0.25 \cdot C$ . In control interval  $k$ , when steady-state conditions are reached and all virtual queues are stable, we have  $F_4(k) = 0.25 \cdot C$ ,  $r_{15}(k) = r_{25}(k) = r_{35}(k) = r_{45}(k) = 0.25 \cdot C$ . Now assume that the demand of flow  $f(4,5)$  decreases and we have  $d_{45} = 0.1 \cdot C$ . Therefore, for control interval  $k + 1$  we have  $r_{15}(k + 1) = r_{25}(k + 1) = r_{35}(k + 1) = 0.25 \cdot C$ , and  $r_{45}(k + 1) = 0.1 \cdot C$ . In this case, node 4 should increase its fair rate to allocate  $0.15 \cdot C$  unused capacity on link  $l(4,5)$ . As  $F_4(k) = 0.25 \cdot C$  and  $r_{15}(k + 1) = r_{25}(k + 1) = r_{35}(k + 1) = 0.25 \cdot C$ , node 4 considers all nodes 1,2 and ,3 as rate-limiter nodes. Therefore, the new fair rate is calculated as  $F_4(k + 1) = F_4(k) + \frac{1}{3} \cdot 0.15 \cdot C = 0.3 \cdot C$ . Node 4 expects all nodes 1, 2, and 3 to increase their rates to the new fair rate, i.e.  $0.3 \cdot C$ . However, nodes 2 and 3 cannot increase their rates as their demand is limited to  $0.25 \cdot C$ . In fact, nodes 2 and 3 was considered rate-limited incorrectly as their demand happened to be equal to the fair rate in control interval  $k$ . Thus, in control interval  $k + 2$  we have,  $r_{15}(k + 2) = 0.3 \cdot C$ ,  $r_{25}(k + 2) = r_{35}(k + 2) = 0.25 \cdot C$ , and  $r_{45}(k + 2) = 0.1 \cdot C$ . It can be seen that there

is still  $0.1 \cdot C$  unused bandwidth available at node 4. Hence, node 4 increases its fair rate again. In this case, node 4 is the only rate-limited node and the fair rate is set to  $F_4(k+2) = F_4(k+1) + 0.1 \cdot C = 0.4 \cdot C$ . As a result, in control interval  $k+3$  we have  $r_{15}(k+3) = 0.4 \cdot C$ ,  $r_{25}(k+3) = r_{35}(k+3) = 0.25 \cdot C$ , and  $r_{45}(k+3) = 0.1 \cdot C$  and the fair rate converges.

Note that in this particular scenario, the DVSR fair rate in control interval  $k+1$  is calculated as  $F_4(k+1) = F_4(k) + 0.15 \cdot C = 0.4 \cdot C$ . Therefore, the fair rate converges in control interval  $k+2$  when DVSR is employed. In fact, using VQ in this scenario results in  $0.1 \cdot C$  bandwidth loss in one control interval while DVSR does not cause any bandwidth loss. ◀

We observed in the above scenario that the conservative approach for fair rate adjustment in VQ may result in bandwidth loss in some scenarios. That is, when the demand of some nodes is exactly equal to the advertised fair rate of the head-node, those nodes are considered as rate-limited nodes. Consequently, when unused bandwidth exists, it may not be reclaimed instantly. In fact, this is the price we pay in those cases to ensure that nodes are not starved as a result of the fair rate overestimation in DVSR in other scenarios.

### 5.2.6 Characterizing Fairness Properties of VQ Algorithm

In this section, we study some fairness properties of the VQ algorithm. In a congestion span, all nodes adjust their rates according to the fair rate advertised by the head-node. In other words, the head-node of the congestion span has to regulate the traffic of its upstream nodes to be able to add its own local traffic to the ring. When upstream nodes in a congestion span change their rates, the head-node has to adjust its fair rate to maintain fairness among nodes. However, during a convergence period and before reaching a steady-state condition, some nodes may be temporarily starved

as the head-node may not be able to estimate the fair rate correctly. In particular, the head-node of the congestion span is starved more than other nodes. In order to characterize some fairness properties of VQ, we define the following parameters measured at the head-node (node  $n$ ) of a congestion span.

- $\sigma(k)$  is the amount of traffic of the head-node (in bits) throttled below its fair share during control interval  $k$ . That is,

$$\sigma(k) = r_n^n(k) \cdot T - r_n^a(k) \cdot T, \quad (5.2.4)$$

where  $r_n^a(k)$  is the add-rate (service-rate) of local traffic of the head-node in control interval  $k$ .

- $ATT(k)$  is the accumulated throttled traffic (ATT) of the head-node (in bits) throttled below its fair share over  $k$  control intervals. That is,

$$ATT(k) = \sum_{m=1}^k \sigma(m). \quad (5.2.5)$$

In fact,  $ATT(k)$  is equal to the increase in the size of the virtual queue at the head-node in the past  $k$  control intervals. Further, we may consider the average amount of throttled traffic of the head-node per unit of time (expressed in bits per second) and denoted by  $\langle ATT \rangle(k)$ . It measures the rate of the throttled traffic of the head-node and calculated as  $\langle ATT \rangle(k) = ATT(k)/(k \cdot T)$ .

**Definition 5.2.1.** Consider a congestion span where a RIAMM-fair solution is achieved and the network is in the steady-state conditions. In case of a change in the network traffic, the fair rate of the head-node will change and the steady-state condition is reached again after several control intervals. We define *starvation time*, which is denoted by  $S$ , as the number of control intervals after a change in the network traffic during which the head-node of a congestion span is starved (i.e. its service rate is less than its fair share) before reaching the steady state conditions again.

*Remark 5.2.4.* According to Remark 5.2.2, the starvation time in VQ is equal to zero when the network traffic decreases in a congestion span.

In order to characterize some fairness properties of the VQ algorithm, we derive some upper bounds on starvation time and accumulated throttled traffic of the head-node during starvation time. Unless otherwise specified, the following assumptions have been made in this analysis.

- ▶ We consider a congestion span in which node  $n$  is the head-node and there is at least a rate-limited IA-flow sending traffic through node  $n$ . This means that the network is congested as at least the rate of an IA-flow is limited to the fair rate of node  $n$ .
- ▶ We suppose that there is no propagation delay in the network and, as a result, the fair rate of the head-node of the congestion span is known by all other nodes right after it is advertised.
- ▶  $D$  is the total demand of node  $n$ , i.e.  $D = \sum_{j>n} d_{nj} \leq C$  (the total demand of node  $n$  for adding FE traffic to the ring cannot usually be more than the unreserved rate).
- ▶ It is assumed that there is no congested node in the downstream of node  $n$ . Hence, the local traffic of the head-node is only limited to the fair rate of node  $n$  and we have  $r_n^n(k) = \min\{F_n(k-1), D\}$ . Note that  $E^l(k) + E^R(k)$  is the total eligible traffic entering the virtual queue of node  $n$  in control interval  $k$ . As the rate of eligible transit traffic entering the virtual queue is equal to or less than  $C$ , we have  $E^l(k) + E^R(k) \leq C \cdot T + r_n^n(k) \cdot T = (C + \min\{F_n(k-1), D\}) \cdot T$ .

We first state and prove the following proposition which is used in deriving the upper bounds.

**Proposition 5.2.2.** *Suppose that the virtual queue of the head-node in a congestion span is unstable and the rate of the eligible transit traffic entering the virtual queue is less than  $C$  in a given control interval. If the demands do not change, the head-node will not be starved in the following control intervals when VQ is employed.*

*Proof.* Assume that in control interval  $k$  the virtual queue of node  $n$  is unstable, i.e.  $E^l(k) + E^R(k) > C \cdot T$ . As the rate of transit traffic entering node  $n$  is less than  $C$  (i.e. local traffic of node  $n$  is partially served), we also have  $E^l(k) + E^R(k) < (C + \min\{F_n(k-1), D\}) \cdot T$  or equivalently,  $E^l(k) < C \cdot T + \min\{F_n(k-1), D\} \cdot T - E^R(k)$ . We assumed that there is at least one rate-limited nodes which means  $E^R(k) \neq 0$  and, hence,  $E^R(k) \geq F_n(k-1) \cdot T \geq \min\{F_n(k-1), D\} \cdot T$ . It then follows that  $E^l(k) < C \cdot T$ . As a result, according to Equations (5.2.2) and (5.2.3), the new fair rate of node  $n$  is calculated as  $F_n(k) = (C \cdot T - E^l(k)) / (|\mathbf{R}| \cdot T)$ . One of the following cases occurs:

i) All input-limited nodes are still input-limited with respect to the new fair rate (i.e.  $E^l(k+1) = E^l(k)$ ) in control interval  $k+1$ . In this case, we have  $E^l(k+1) + E^R(k+1) = E^l(k) + |\mathbf{R}| \cdot F_n(k) \cdot T = C \cdot T$ , which means that the virtual queue of node  $n$  is stable and node  $n$  is not starved in control interval  $k+1$ .

ii) Some of the nodes which were input-limited according to the previous fair rate, become rate-limited with respect to the new fair rate as  $F_n(k) < F_n(k-1)$ . Suppose that node  $i$  which was considered input-limited in control interval  $k$  (i.e.,  $r_i^n(k) < F_n(k-1)$ ), is in fact rate-limited with respect to the new fair rate (i.e.,  $r_i^n(k+1) = F_n(k) < r_i^n(k)$ ). Hence,  $E^l(k+1) + E^R(k+1) = (E^l(k) - r_i^n(k)) + (|\mathbf{R}| + 1) \cdot F_n(k) \cdot T = C \cdot T - r_i^n(k) + F_n(k) < C \cdot T$ , where  $|\mathbf{R}|$  is the number of rate-limited nodes in control interval  $k$ . In other words, the new fair rate is less than the true fair rate and the total amount of traffic entering the virtual queue is less than the amount of traffic that can be serviced. That is, unused capacity exists at the head-node and the fair rate should be increased. However, according to Remark 5.2.2, the head-node

is not starved in VQ when the fair rate is increased.

Hence, when the fair rate of node  $n$  decreases and the rate of eligible transit traffic entering the virtual queue of node  $n$  is less than  $C$  in control interval  $k$ , the head-node will not be starved in the following control intervals.  $\square$

**Proposition 5.2.3.** *Consider a congestion span which is in the steady-state condition and the fair rate of the head-node is  $\frac{C}{m_1}$ ,  $m_1 \geq 1$ , and its total demand is  $D$ . Assume that there is an increase in the network traffic and the new fair rate of the head-node in the steady-state is  $\frac{C}{m_2}$ ,  $m_2 > m_1$ . An upper bound for the starvation time of the head-node in VQ as a function of  $m_1$ ,  $m_2$ , and  $D$ , is given by*

$$U_S(m_1, m_2, D) = \begin{cases} \lceil m_2 - m_1 \rceil, & \text{if } D \geq \frac{C}{m_1}, \\ \left\lceil \frac{\log(m_2/m_1)}{\log(1+D/C)} \right\rceil, & \text{if } D \leq \frac{C}{m_2}, \\ \left\lceil m_2 - \frac{C}{D} \right\rceil + \left\lceil \frac{\log(C/(D \cdot m_1))}{\log(1+D/C)} \right\rceil, & \text{otherwise.} \end{cases}$$

*Proof.* See Appendix B.  $\square$

**Proposition 5.2.4.** *An upper bound for accumulated throttled traffic of the head-node during the starvation time in Proposition 5.2.3 is given by*

$$U_{ATT}(m_1, m_2, D) = \begin{cases} \sum_{i=0}^{\lceil m_2 - m_1 \rceil - 1} \frac{C \cdot T}{m_1 + i}, & \text{if } D \geq \frac{C}{m_1}, \\ \left\lceil \frac{\log(m_2/m_1)}{\log(1+D/C)} \right\rceil \cdot D \cdot T, & \text{if } D \leq \frac{C}{m_2}, \\ \left\lceil \frac{\log(C/(D \cdot m_1))}{\log(1+D/C)} \right\rceil \cdot D \cdot T + \sum_{i=0}^{\lceil m_2 - C/D \rceil - 1} \frac{C \cdot T}{C/D + i}, & \text{otherwise.} \end{cases}$$

*Proof.* See Appendix B.  $\square$

## 5.3 Performance Evaluation of VQ Algorithm

In this section, we evaluate the performance of the VQ algorithm through simulation results. Moreover, we compare the performance of the VQ algorithm with that of RPR-AM, RPR-CM, and DVSR. The simulation model is implemented in OPNET. Unless otherwise specified, we assume that  $C = 100\text{Mbps}$ ,  $\tau_h = 0.95 \cdot C$ ,  $\tau_\ell = 0.85 \cdot C$ ,  $T = 1\text{msec}$ ,  $lpCoef = 16$ ,  $rampUpCoef = rampDnCoef = 64$ , 512 bits packet size, and  $25\mu\text{sec}$  link propagation delay. In addition, we only consider FE traffic in all simulation scenarios and assume that the transit buffer in all nodes is implemented in single-queue mode.

### 5.3.1 Spatial Reuse and Steady-State Performance

In this section, we study the steady-state performance of VQ in achieving spatial reuse and fairness in some benchmark scenarios. In order to quantify fairness of rate allocations in VQ, we consider a fairness index defined in [70]. Assume that  $\bar{r}_{ij}$  is the measured allocated rate of flow  $f(i, j)$  in a simulation scenario where VQ algorithm is employed. Further, assume that  $r_{ij}$  is the ideal RIAMM-fair allocated rate of flow  $f(i, j)$  in the same scenario. The fairness index ( $FI$ ) is then defined as

$$FI = \left( \sum_{i,j} x_{ij} \right)^2 / \left( N \cdot \sum_{i,j} x_{ij}^2 \right), \quad (5.3.1)$$

where  $x_{ij} \equiv \bar{r}_{ij}/r_{ij}$  and  $N$  is the total number of the flows. It can be shown that  $FI$  is bounded and we have  $0 \leq FI \leq 1$ . The fairness index of the ideal fair solution is equal to 1. In fact, a solution has better fairness properties when its corresponding fairness index is closer to 1.

In this experiment, we consider Pareto traffic in order to model a very bursty traffic. We assume that packet inter-arrival times of each traffic each flow are distributed according to a Pareto distribution with shape parameter  $\mathbf{a}$  and scale (location)

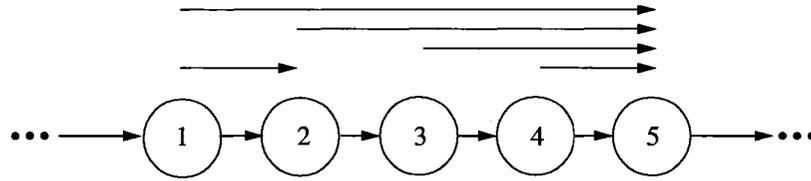


Figure 5.9: Parallel parking-lot scenario

parameter  $\mathbf{b}$ . For  $\mathbf{a} > 1$ , the mean is defined as  $\frac{\mathbf{a}\cdot\mathbf{b}}{\mathbf{a}-1}$ . Further, for  $1 < \mathbf{a} < 2$  the variance does not converge and, as a result, the generated traffic is extremely bursty. In the following, we assume that  $\mathbf{a} = 1.5$  and then calculate  $\mathbf{b}$  from the mean value. We also assume that the MMP source behavior is employed at each node (see Section 3.4.3).

We first consider a 4-node parallel parking-lot scenario depicted in Figure 5.9. In this scenario, all nodes have local traffic destined to node 5. Moreover, node 1 has a local flow destined to node 2. We assume that the demand of flow  $f(1, 2)$  is  $C$  and other sources generates Pareto traffic with mean  $0.5 \cdot C$ . In this case, the RIAMM-fair allocated rates are  $r_{15} = r_{25} = r_{35} = r_{45} = 0.25 \cdot C$ . In addition,  $r_{12} = 0.75 \cdot C$  as spatial reuse is fully achieved with the MMP source behavior. The simulation results of the normalized allocated rates with 95% confidence interval are  $\bar{r}_{12} = 0.7453 \pm 0.0045$ ,  $\bar{r}_{15} = 0.2533 \pm 0.0026$ ,  $\bar{r}_{25} = 0.2535 \pm 0.0023$ ,  $\bar{r}_{35} = 0.2520 \pm 0.0016$  and  $\bar{r}_{45} = 0.2410 \pm 0.0063$ . It can be seen that the allocated rates are within  $\pm 4\%$  of the RIAMM fair rates. For this solution we have  $FI = 0.99965$ , which means that VQ approximately achieves the ideal fair solution in this scenario. This fair solution can also be closely achieved by DVSR. In order to study the effect of different traffic distributions, we repeat this experiment with Poisson traffic with the same mean rates. In this case, the normalized allocated rates are  $\bar{r}_{12} = 0.74846 \pm 0.00014$ ,  $\bar{r}_{15} = 0.25140 \pm 0.00018$ ,  $\bar{r}_{25} = 0.25154 \pm 0.00011$ ,  $\bar{r}_{35} = 0.24968 \pm 0.00016$  and  $\bar{r}_{45} = 0.24726 \pm 0.00031$ . For this set of allocated rates we have  $FI = 0.99996$ . It can be seen that the steady-state results are similar to those obtained with Pareto

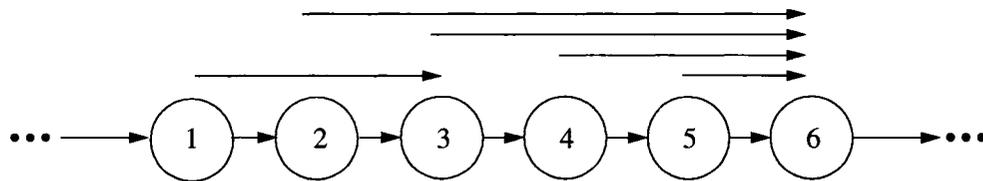


Figure 5.10: Upstream parking-lot scenario

distribution and the average throughput in both cases are very close. Note that the network is overloaded in this scenario. This, in fact, reduces the effect of traffic variations on the steady-state results. Moreover, the fair rate is calculated at the congested node based on the average traffic received from each node during a control interval. In other words, the fair rate does not depend on the traffic variations of flows during a control interval. Hence, the steady-state results with different traffic distributions are very close.

In the second scenario, we consider an upstream parking-lot scenario depicted in Figure 5.10. In this scenario, node 1 has a local flow destined to node 3 and nodes 2, 3, 4, and 5 have local flows destined to node 6. We assume that the demand of flow  $f(1,3)$  is  $C$  and other sources generates Pareto traffic with mean  $0.5 \cdot C$ . As demands are unbalanced in this scenario, RPR-AM exhibits permanent oscillations (see Section 5.1.1). The normalized RIAMM-fair allocated rates in this scenario are  $r_{26} = r_{36} = r_{46} = r_{56} = 0.25$  and  $r_{13} = 0.75$ . The allocated rates obtained from simulation when VQ employed at all nodes are  $\bar{r}_{13} = 0.7539 \pm 0.0045$ ,  $\bar{r}_{26} = 0.2399 \pm 0.0056$ ,  $\bar{r}_{36} = 0.2557 \pm 0.0101$ ,  $\bar{r}_{46} = 0.2583 \pm 0.0019$  and  $\bar{r}_{56} = 0.2420 \pm 0.0038$ . It can be seen that the allocated rates are within  $\pm 4\%$  of the ideal rates. The fairness index for this solution is equal to 0.99914.

Finally, we consider the two-exit parking-lot scenario depicted in Figure 5.11 to study the performance of the VQ algorithm in achieving fairness among local flows as well as among IA-flows. In this scenario, nodes 1, 2, 3 and 4 have local flows destined to node 5. Node 4 has an additional flow destined to node 6. We assume that all

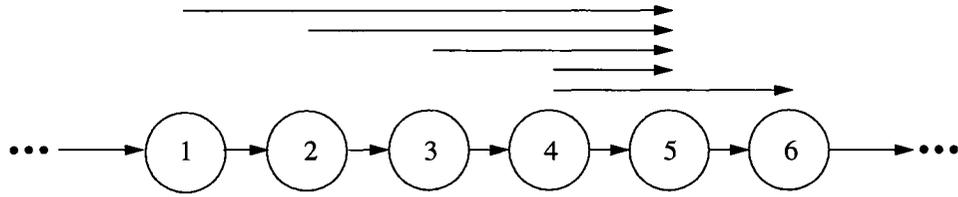


Figure 5.11: Two-exit parking-lot scenario

sources generate Pareto traffic with mean rate  $0.5 \cdot C$ . To maintain source-based fairness, the bandwidth on link  $l(4, 5)$  should be allocated equally among all competing nodes. This means that the RIAMM-fair normalized allocated rates for this scenario are  $r_{15} = r_{25} = r_{35} = 0.25$  and  $r_{45} = r_{46} = 0.125$ . The allocated rates obtained from simulation results with VQ are  $\bar{r}_{15} = 0.25162 \pm 0.00057$ ,  $\bar{r}_{25} = 0.25138 \pm 0.00065$ ,  $\bar{r}_{35} = 0.25012 \pm 0.00055$  and  $\bar{r}_{45} = \bar{r}_{46} = 0.12344 \pm 0.00029$ . It can be seen that fairness is maintained among flows  $f(4, 5)$  and  $f(4, 6)$  as well as among IA-flows. The simulation results in this case are within  $\pm 2\%$  of the ideal RIAMM-fair rates and the fairness index of this rate allocation is equal to 0.99993.

### 5.3.2 Comparison with RPR-AM and RPR-CM

In this section, we compare the performance of VQ with that of RPR-AM and RPR-CM in terms of convergence times and throughput loss.

#### Convergence Times

In this experiment, we compare convergence times of VQ with that of RPR-AM and RPR-CM. We consider a 4-node parking-lot scenario in which all nodes have local traffic destined to node 0 with rate  $C$ . We assume that nodes 4, 3, 2, and 1 start sending traffic to node 0 at times 0, 0.1, 0.2, and 0.3 seconds, respectively. In order to decrease the congestion detection time and observe the fair rate oscillations in RPR-AM, we assume that  $lpCoeF = 2$ . Figures 5.12, 5.13, and 5.14 present the throughput

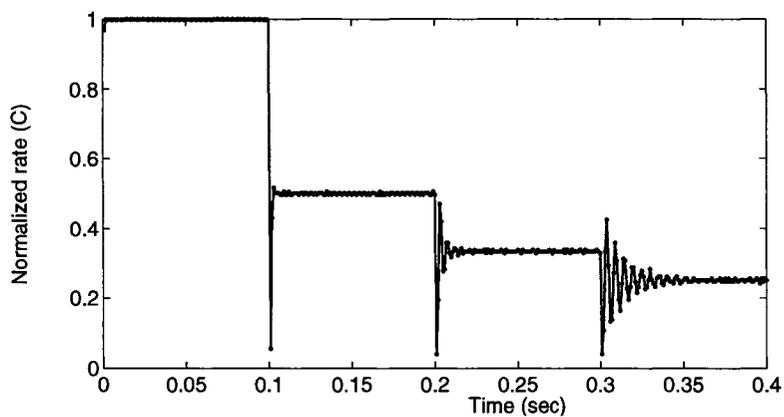


Figure 5.12: Convergence times of the add-rate of the head node with RPR-AM

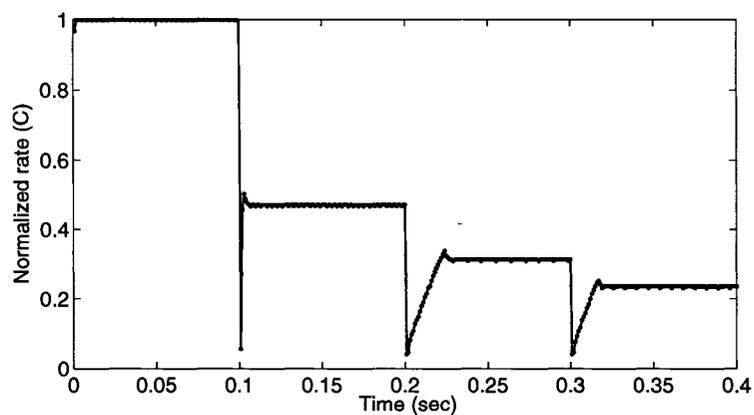


Figure 5.13: Convergence times of the add-rate of the head node with RPR-CM

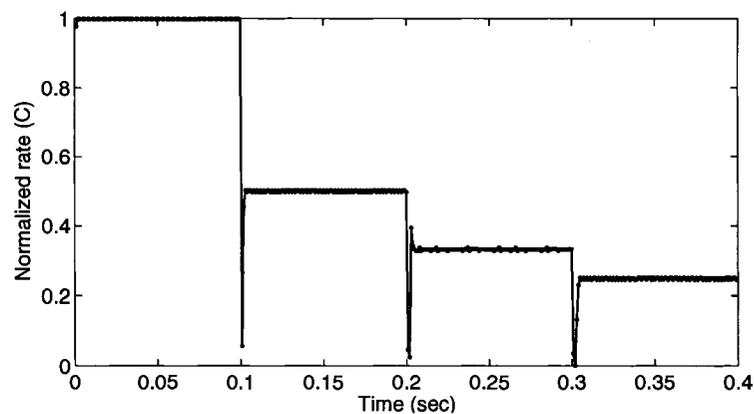


Figure 5.14: Convergence times of the add-rate of the head node with VQ

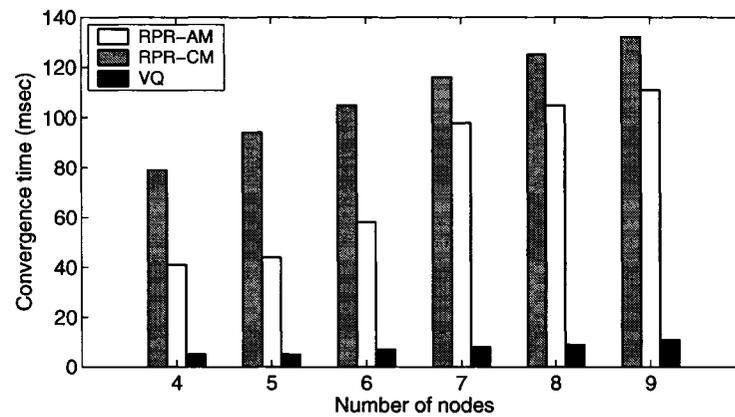


Figure 5.15: Convergence times of RPR-AM, RPR-CM, and VQ

(add-rate) of node 4 (head-node) with RPR-AM, RPR-CM, and VQ, respectively. It can be seen that the add-rate of the head-node converges to  $\frac{C}{2}$ ,  $\frac{C}{3}$ , and  $\frac{C}{4}$ , after nodes 3, 2, and 1 starts sending traffic. In RPR-AM, the add-rate of node 1 exhibits some oscillations before convergence is achieved and the convergence time can be as long as 40 msec. In RPR-CM, the fair rate is highly dampened and, as a result, the add-rate of node 1 converges without oscillations. In this particular case, the convergence times of RPR-CM can be up to 30 msec. The fair rate in RPR-AM and RPR-CM is not accurately adjusted and, hence, the convergence times of the fair rate are usually long. On the other hand, the fair rate in VQ is calculated more accurately which results in a very short convergence time. This can be seen in Figure 5.14 where the add-rate of the head-node converges in less than 5 msec.

We also consider a parking-lot scenario in which all nodes start sending traffic at 0 seconds. In this experiment, we assume that each node has traffic destined to node 0 at rate  $C$  and  $lpCoef = 16$ . Figure 5.15 compares convergence times of RPR-AM, RPR-CM and VQ for different number of nodes. It can be seen that in all algorithms convergence time increases with the number of nodes. Further, the convergence time of VQ is an order of magnitude less than that of RPR-AM and RPR-CM. The convergence times of DVSR are also within the same range as that of

VQ. Note that in this scenario, RPR-CM converges more slowly compared to RPR-AM. In general, the convergence times of RPR-AM and RPR-CM depend on the values of low-pass filtering and ramping coefficients and, hence, there exist scenarios in which RPR-CM converges faster than RPR-AM.

### Throughput

We now compare the performance of VQ with that of RPR-AM and RPR-CM in terms of achieved throughput. Figure 5.16 presents the simulation results for a 2-node parking-lot scenario (Figure 5.1) in a dynamic traffic environment. We have assumed that a dynamic source (described in Section 5.1.1) is introduced at node 2 (head-node). In this scenario, all sources generate Poisson traffic with the specified mean rates; node 1 generates traffic at rate  $C$  and the dynamic source generates Poisson traffic with mean rates  $\delta_L = 0.01 \cdot C$ , and  $\delta_H = 0.5 \cdot C$  in the low-state and the high-states, respectively. We assume that  $T_H = 5 \cdot T$  and  $T_L$  varies from  $0.01 \cdot T_H$  to  $20 \cdot T$ . One can see that the throughput loss for RPR-CM is a non-monotonic function of  $T_L$ . When  $T_L$  is low, the unused bandwidth of the dynamic source is not significant and the throughput loss is low. The throughput loss increases with  $T_L$  up to  $T_L = T_H$ . However, for  $T_L > T_H$ , it can be seen that the throughput loss decreases as RPR-CM converges and node 1 can reclaim the unused bandwidth of the dynamic source. The throughput loss increases with  $T_L$  for RPR-AM. Note that RPR-AM never converges in this scenario and, as a result, node 1 periodically limits its add-rate to the fair rate of node 2. As  $T_L$  increases, the add-rate of node 1 decreases and, hence, the fair rate decreases. Therefore, the throughput loss increases with  $T_L$  in RPR-AM. For VQ, the throughput loss is less than 1% in all cases.

In Figure 5.17, we consider the same 2-node parking-lot scenario with  $T_L = 2 \cdot T_H = 10 \cdot T$ , where the mean rate of Poisson traffic generated by the dynamic source in the low-state ( $\delta_L$ ) varies from  $0.01 \cdot C$  to  $0.5 \cdot C$ . It can be seen that the throughput loss

decreases with  $\delta_L$  in RPR-AM and RPR-CM. In RPR-AM, the fair rate increases with  $\delta_L$  and, hence, the throughput loss decreases. Note that for  $\delta_L = 0.5 \cdot C$ , the throughput loss is zero. In fact, the dynamic source does not fluctuate in this case as  $\delta_L = \delta_H$  and, hence, the fair rate converges. In RPR-CM, the unused bandwidth of the dynamic source decreases with  $\delta_L$  and, as a result, the throughput loss decreases. One can see that the throughput loss for VQ is again less than 1% in all cases. In Figure 5.18, we considered the same scenario as in Figure 5.17 with  $T_L = 10 \cdot T_H = 50 \cdot T$ . By comparing Figures 5.17 and 5.18, we can see that as  $T_L$  increases, the throughput loss in RPR-CM decreases while it increases in RPR-AM. The reason is that as  $T_L$  increases, RPR-CM converges and the throughput loss decreases. However, the fair rate decreases with  $T_L$  in RPR-AM and, hence, the throughput loss increases. In VQ, the throughput loss is less than 0.5% in all cases.

In the following, we repeat the above experiments with Pareto traffic to study the effect of different distributions on throughput loss. We assume that all sources generate Pareto traffic with shape parameter 2 (i.e.  $\mathbf{a} = 2$ ). Similar to the previous scenario, node 1 generates traffic at rate  $C$  and the dynamic source is introduced at node 2 generating traffic with mean rates  $\delta_H = 0.5 \cdot C$  and  $\delta_L$  in the high-state and the low-state, respectively. Figures 5.19 and 5.20 present throughput loss versus  $\delta_L$  for  $T_L = 2 \cdot T_H = 10 \cdot T$  and  $T_L = 10 \cdot T_H = 50 \cdot T$ , respectively. Comparison of Figure 5.19 (respectively Figure 5.20) and Figure 5.17 (respectively Figure 5.18) shows that throughput loss does not significantly change for different traffic distributions. The reason is similar to the one explained in Section 5.3.1. RPR-AM and RPR-CM suffer from high throughput loss with Pareto traffic as well. However, throughput loss with VQ is still less than 1% in all cases. Note that DVSR can also achieve high throughput in these scenarios similar to VQ.

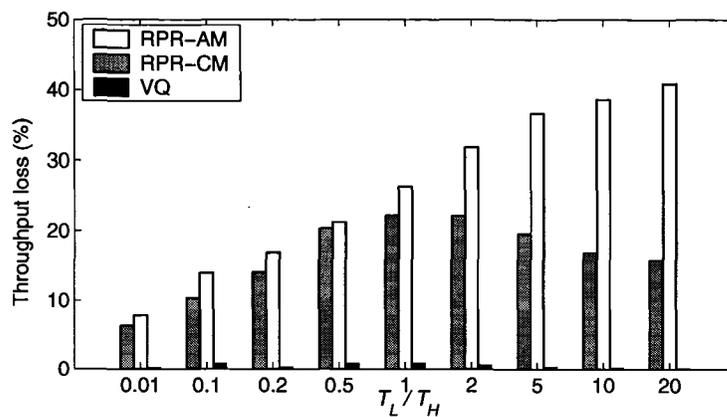


Figure 5.16: Throughput loss versus  $\frac{T_L}{T_H}$  with Poisson traffic

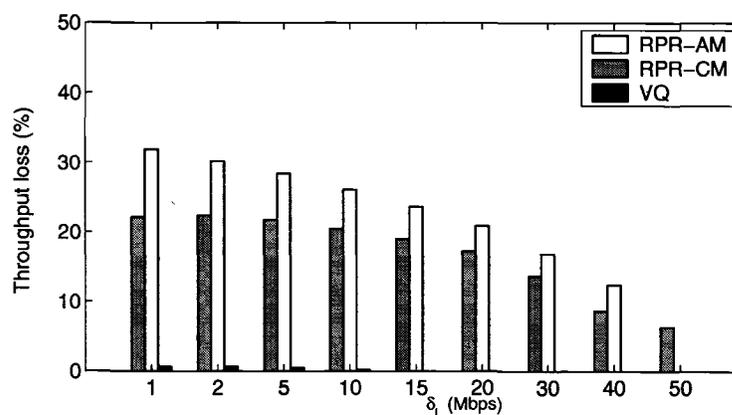


Figure 5.17: Throughput loss versus  $\delta_L$  for  $T_L = 2 \cdot T_H$  with Poisson traffic

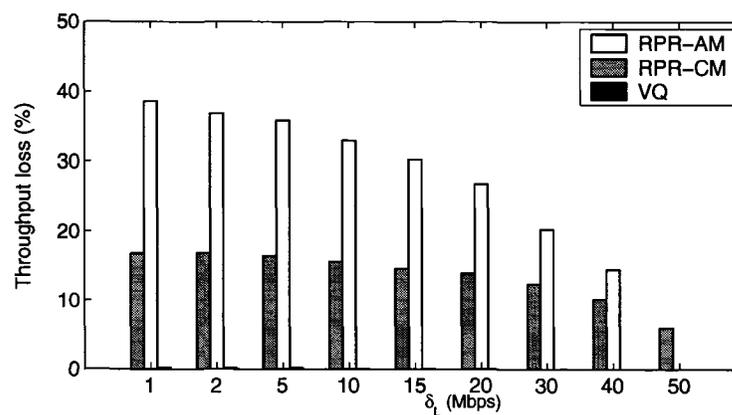


Figure 5.18: Throughput loss versus  $\delta_L$  for  $T_L = 10 \cdot T_H$  with Poisson traffic

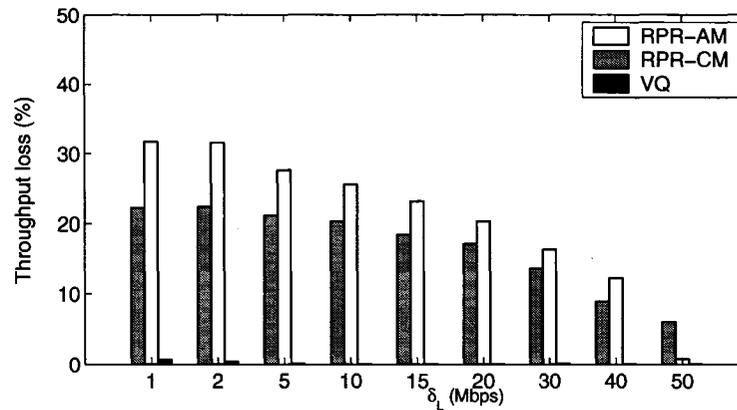


Figure 5.19: Throughput loss versus  $\delta_L$  for  $T_L = 2 \cdot T_H$  with Pareto traffic

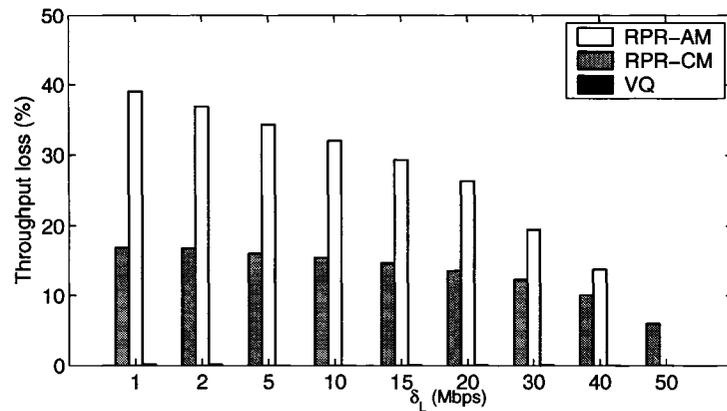


Figure 5.20: Throughput loss versus  $\delta_L$  for  $T_L = 10 \cdot T_H$  with Pareto traffic

### 5.3.3 Comparison with DVSR

In this section, we compare the performance of VQ with that of DVSR in terms of the aggregate throttled traffic of the head-node. We consider a four-node parking-lot scenario in which a dynamic source is applied at node 1 (tail-node). We assume that nodes 2, 3, and 4 (head-node) can send traffic at rate  $C$ . Moreover, we assume that  $T_H = T_L = T = 2$  msec and the dynamic source generates constant traffic at rates  $\delta_H = 0.25 \cdot C$  and  $\delta_L = 0.1 \cdot C$  in the high-state and the low-state, respectively.<sup>2</sup>

<sup>2</sup>Here, we consider constant traffic to make the results visually more understandable. We will shortly study more realistic scenarios.

Figure 5.21 presents the achieved throughput of nodes 2, 3 and 4 when DVSR is employed. In this scenario, we have assumed that the dynamic source at node 4 starts fluctuating at 0.1 seconds. Notice that before this time, all nodes send traffic at rate  $0.25 \cdot C$ . When the dynamic source is in the high-state the calculated fair rate is equal to  $0.25 \cdot C$ . When the dynamic source at node 1 is in the low-state, the fair rate of node 4 in DVSR is increased to  $0.375 \cdot C$ . As a result, nodes 2 and 3 increase their rates to the new fair rate. (Note that the add-rate of nodes 2 and 3 is always equal to the fair rate of node 4 in this scenario.) At the same time, the dynamic source enters the high-state and sends traffic at rate  $0.25 \cdot C$ . Consequently, there is not enough bandwidth available for node 4 and it is starved. In Figure 5.22, the same scenario is studied with VQ. It can be seen that when the dynamic source is in the low-state, the estimated fair rate in VQ is around  $0.3 \cdot C$ . In other words, the rate increase of nodes 2 and 3 in VQ is less than the one in DVSR. Hence, there will be more bandwidth available for node 1 when VQ is employed. The minimum add-rate of node 4 is around  $0.17 \cdot C$  with VQ (Figure 5.22), while it can be as low as  $0.03 \cdot C$  with DVSR (Figure 5.21). We measure the accumulated throttled traffic (ATT) of node 4 which shows the amount of traffic of this node throttled below its fair share in this scenario. Figure 5.23 presents the ATT of node 4 (in Mbits) for VQ and DVSR. It can be seen that as node 4 is periodically starved in DVSR, its ATT increases over time. In DVSR, the ATT of node 4 increases from 0.5Mb at 0.1msec to 1.03Mb at 0.15msec. In fact, the average rate of the ATT in DVSR is around 10Mbps. That is, the add-rate of node 4 is 10Mbps less than its fair share on average. In VQ, however, the ATT of node 4 does not significantly increase over time and remains in the same range.

One can see in Figure 5.23 that the ATT of node 4 in DVSR and VQ increases in one control interval (when the dynamic source is in the high-state) and then decreases in the following control interval (when the dynamic source is in the low-state). The

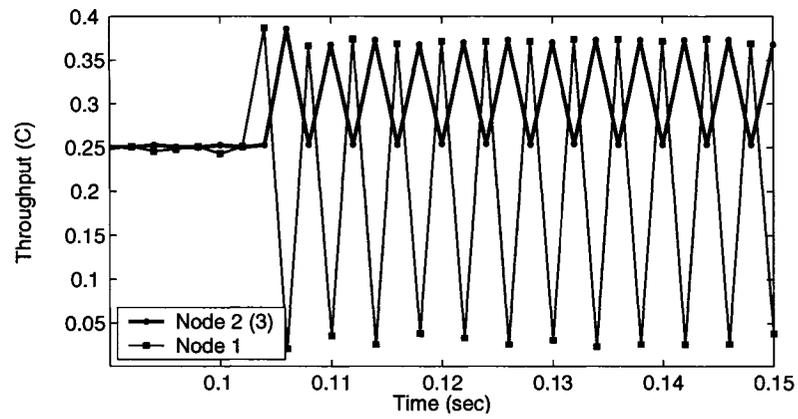


Figure 5.21: Throughput with DVSR (node 4 reclaims unused bandwidth)

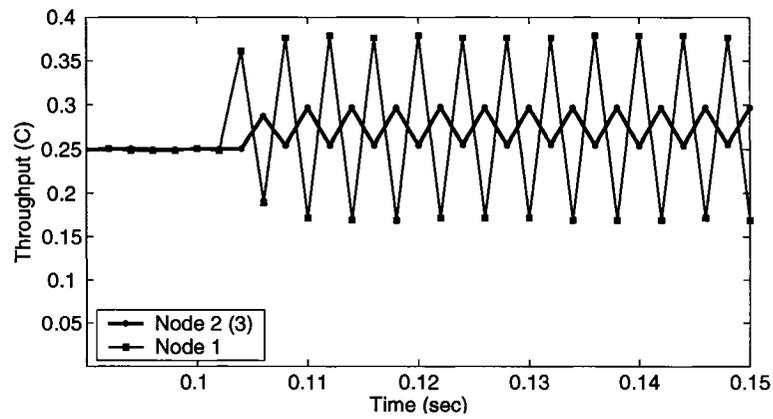


Figure 5.22: Throughput with VQ (node 4 reclaims unused bandwidth)

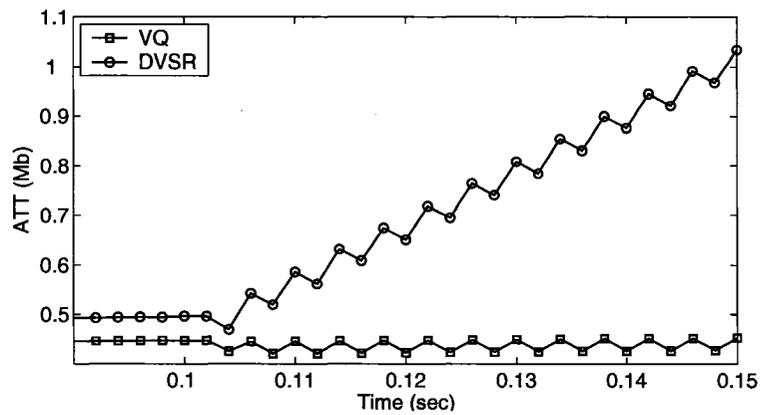


Figure 5.23: ATT in DVSR and VQ (node 4 reclaims unused bandwidth)

reason is that when the dynamic source at node 1 is in the low-state, unused bandwidth is available at node 4. In this case, node 4 is allowed to reclaim the unused bandwidth and send traffic beyond its fair share to make up partially for being starved in the previous control intervals. This can also be observed in Figures 5.21 and 5.22 where the fair rate of node 1 is more than  $0.35 \cdot C$  while its fair rate is  $0.25 \cdot C$ . In fact, allowing the head-node to reclaim the unused bandwidth improves the performance of both algorithms in terms of the achieved throughput and the ATT of node 4.

In Figures 5.24 and 5.25, the achieved throughput of different nodes is shown for DVSR and VQ when node 4 is not allowed to reclaim the unused bandwidth. It can be seen that when the dynamic source enters the low-state, the rate of node 4 does not exceed the fair rate, i.e.  $0.25 \cdot C$ , in DVSR and VQ. As a result, the ATT of node 4 is more than the previous case, where node 4 was allowed to reclaim the unused bandwidth. However, as it can be observed in Figure 5.26, the ATT of node 4 with VQ is still less than that of DVSR. In this case, the average rate of the ATT of node 4 in VQ is around 7Mbps while it is around 18Mbps for DVSR.

In order to compare the effect of traffic variations on throttled traffic of the head node in DVSR and VQ, we have studied various scenarios. In the following, we assume that the head-node in DVSR and VQ is allowed to reclaim the unused bandwidth beyond its fair share and  $T = 2$  msec.

In the first scenario, we consider a 4-node parking-lot scenario, in which a dynamic source is applied at node 1 (tail-node) generating Poisson traffic with mean rates  $\delta_H$  and  $\delta_L$  in the high-state and the low-state, respectively. We assume that  $T_H = T_L = 2 \cdot T$  and  $\delta_H = 0.25 \cdot C$  and other nodes can send Poisson traffic at rate  $C$ . Figure 5.27 shows the average rate of throttled traffic of the head-node ( $\langle ATT \rangle$ ), which is normalized to the calculated fair rate of the head-node, for various values of the low-rate of the dynamic source (i.e.  $\delta_L$ ). Note that when the dynamic source enters the

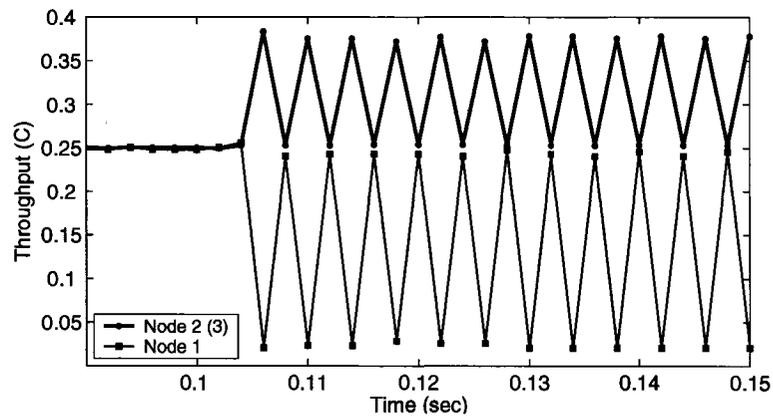


Figure 5.24: Throughput with DVSR (node 4 does not reclaim unused bandwidth)

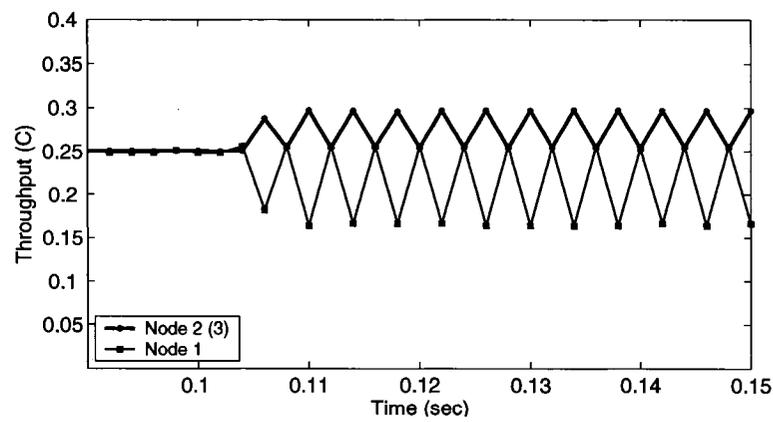


Figure 5.25: Throughput with VQ (node 4 does not reclaim unused bandwidth)

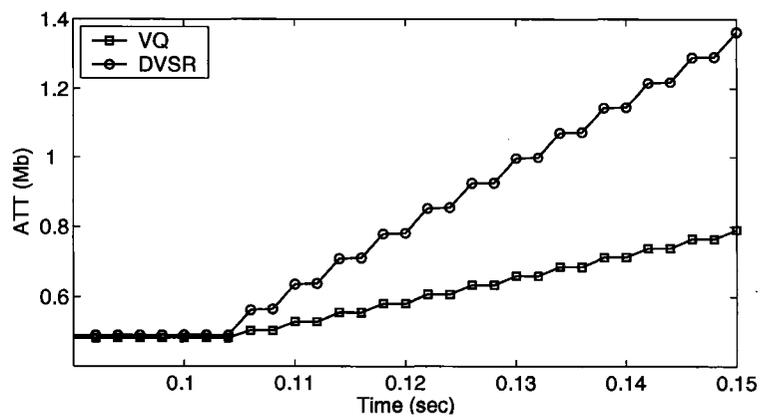


Figure 5.26: ATT in DVSR and VQ (node 4 does not reclaim unused bandwidth)

high-state, the head-node is starved in VQ and DVSR similarly. In this case, the head-node quickly decreases its fair rate to end the starvation. On the other hand, when the dynamic-source enters the low-state, unused bandwidth is available at the head-node. In this case, the head-node is allowed to reclaim the unused bandwidth beyond its fair share with both algorithms. This makes up for the starvation of the head-node which happened when the dynamic source entered the high-state. (This phenomenon was explained in Figures 5.21 to 5.23.) However, the fair rate should be increased to distribute the unused capacity of the dynamic-source among all nodes in the following control intervals. In VQ, the fair rate is computed accurately according to the number of rate-limited nodes to prevent starvation of the head-node. (In fact, the throttled traffic of the head-node in VQ is ideally zero over a cycle of  $T_H + T_L$  seconds in which the dynamic source enters a high-state and then a low-state.) However, when the dynamic source enters the low-state, the fair rate is overestimated in DVSR and, as a result, the head-node is extremely starved. Thus, the fair rate of the head-node should be re-adjusted to end the starvation. In this case, the fair rate converges within a few control intervals during which the head node is starved. In other words, the throttled traffic of the head-node can be significant over a cycle of  $T_H + T_L$  seconds as shown in Figure 5.27 for different values of  $\delta_L$ . Note that as  $\delta_L$  increases, the unused bandwidth of the dynamic source decreases and, hence, the overestimation and fluctuations of the fair rate in DVSR is less. As a result, the rate of throttled traffic of the head-node in DVSR decreases with  $\delta_L$  as shown in Figure 5.27.

Although the head-node should not ideally be starved in this scenario when VQ is employed, it can be seen in Figure 5.27 that the head-node is still starved for different values of  $\delta_L$  and  $\langle ATT \rangle$  for VQ is slightly higher than zero. The reason is that the fair rate of the head-node is gradually received by the upstream nodes due to the propagation delay. Hence, the head-node may be starved for a very short period of time until all nodes receive the fair rate. Clearly, this short period of time is

independent of  $\delta_L$  and as it can be seen in Figure 5.27,  $\langle ATT \rangle$  does not significantly change for different values of  $\delta_L$ . In this scenario,  $\langle ATT \rangle$  can be as high as 31% with DVSR while it is less than 1.5% in all cases with VQ.

Next, we study the effect of  $T_L + T_H$  ( $T_H = T_L$ ) on  $\langle ATT \rangle$  in the same 4-node parking-lot scenario. In this scenario, we assume that the dynamic source is located at node 1 generating Poisson traffic with mean rates  $\delta_H = 0.25 \cdot C$  and  $\delta_L = 0.1 \cdot C$ . We also assume that other nodes generate Poisson traffic at rate  $C$ . Figure 5.28 shows  $\langle ATT \rangle$  for different values of  $T_H + T_L$ . It can be seen that  $\langle ATT \rangle$  decreases with  $T_L + T_H$  when DVSR is employed. The reason is that as  $T_L + T_H$ , the DVSR fair rate can be corrected (and eventually converge) before the next change in the state of the dynamic source. Thus, as  $T_L + T_H$  increases, the starvation time of the head-node decreases and, hence,  $\langle ATT \rangle$  drops. For the same reason explained in the previous scenario, the head-node should not be starved ideally when VQ is employed as the fair rate is computed correctly. However, due to the propagation delay, the head-node may be starved for a short period of time before all nodes receive the fair rate. Hence,  $\langle ATT \rangle$  is slightly higher than zero when VQ is employed. In this scenario, the rate of throttled traffic of the head-node can be as high as 30% with DVSR while it is less than 1% with VQ in all cases.

We also study the effect of the number of nodes ( $N$ ) on the rate of throttled traffic of the head-node. In all cases, we assume that a dynamic-source is applied to the tail-node. The dynamic source generates Poisson traffic with mean rates  $\delta_H = C/N$  and  $\delta_L$  in the high-state and in the low-state, respectively, where  $T_L = T_H = 4 \cdot T$ . Figures 5.29 and 5.30 show the average rate of the throttled traffic of the head-node for  $\delta_L = 0.01 \cdot C$  and  $0.05 \cdot C$ , respectively. In DVSR, when the link capacity is not fully utilized at the head-node, the new fair rate is calculated by adding the unused bandwidth to the old fair rate. Consequently, the rate of all rate-limited flows can be increased to the new fair rate. As the number of nodes increases, more nodes

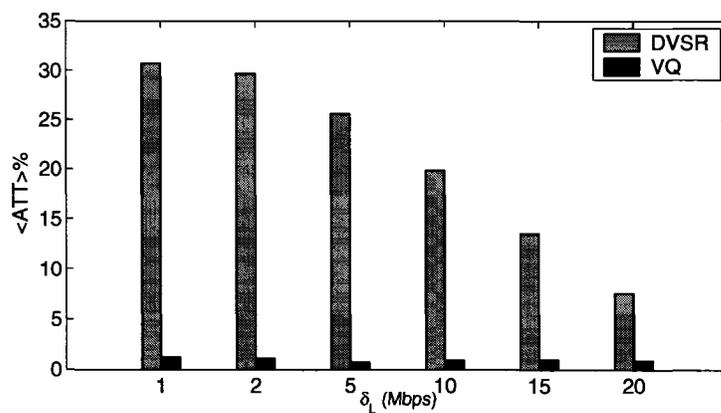


Figure 5.27: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $\delta_L$

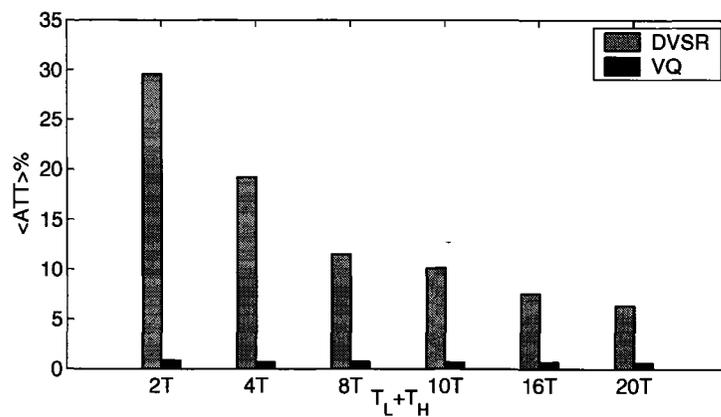


Figure 5.28: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $T_H + T_L$

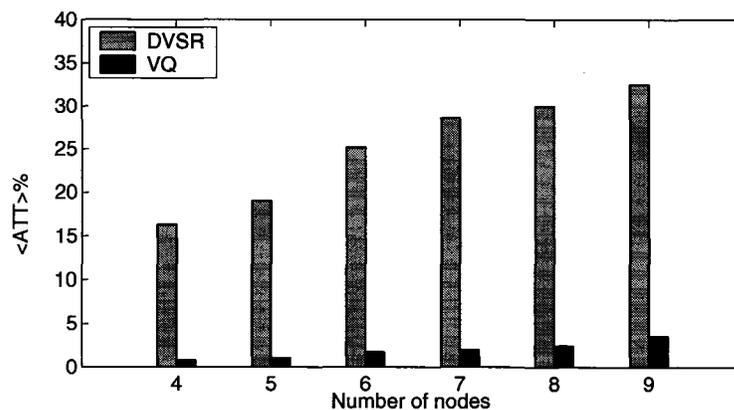


Figure 5.29: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $N$  for  $\delta_L = 0.01 \cdot C$  with Poisson traffic

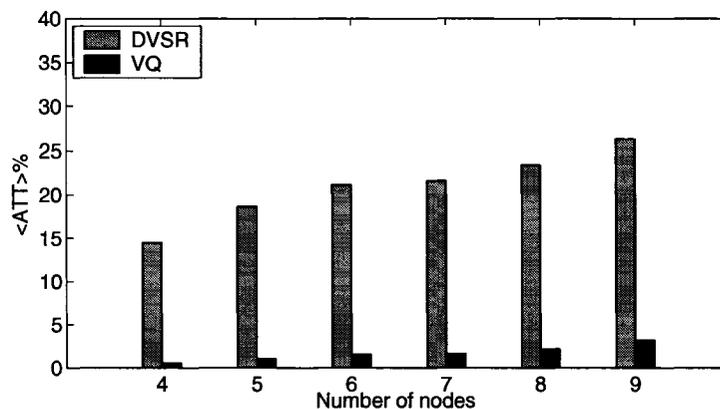


Figure 5.30: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $N$  for  $\delta_L = 0.05 \cdot C$  with Poisson traffic

are able to increase their rates. Hence, the rate of throttled traffic of the head-node increases with the number of nodes in this scenario as shown in Figures 5.29 and 5.30. One can also notice that when  $\delta_L$  increases to  $0.05 \cdot C$ , the average rate of throttled traffic of the head-node for a given  $N$  decreases in Figure 5.30 when compared to the one in Figure 5.29. The reason is that as  $\delta_L$  increases, the unused bandwidth at the head-node decreases and, consequently, the calculated fair rate in DVSR decreases. This means that the rate of transit traffic at the head-node decreases and, as a result, the starvation of the head node is less. Note that in Figures 5.29 and 5.30, the rate of throttled traffic of the head-node with VQ is less than 4% in all cases, while it can be as high as 33% with DVSR.

In order to study the effect of different traffic distributions, we repeat the last two scenarios with Pareto traffic and measure the rate of throttled traffic of the head-node for different values of  $N$ . In all cases, we assume that a dynamic-source is introduced at the tail-node (node 1). The dynamic source generates Pareto traffic with  $\mathbf{a} = 2$  and mean rates  $\delta_H = C/N$  and  $\delta_L$  in the high-state and in the low-state, respectively, where  $T_L = T_H = 4 \cdot T$ .

In Figures 5.31 and 5.32, the average rate of throttled traffic of the head-node is

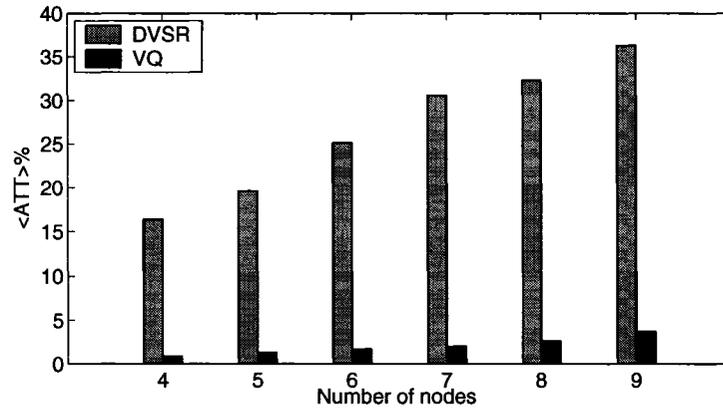


Figure 5.31: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $N$  for  $\delta_L = 0.01 \cdot C$  with Pareto traffic

shown for  $\delta_L = 0.01 \cdot C$  and  $0.05 \cdot C$ , respectively. Comparison between Figure 5.31 (respectively Figure 5.32) and Figure 5.29 (respectively Figure 5.30) shows that the rate of throttled traffic of the head-node does not significantly change for different traffic distributions. The reason is that the network is overloaded in these scenarios and, as a result, the effect of traffic variations is reduced. Furthermore, the fair rate in VQ and DVSR is calculated based on the measured average rates during a control interval. Hence, the results should only depend on the average values and not on traffic variations during a control interval. However, in the scenarios presented in Figures 5.31 and 5.32, there is a dynamic source periodically switching between two states generating Pareto traffic with mean rates  $\delta_H$  and  $\delta_L$  for duration of  $T_H$  and  $T_L$  seconds, respectively. As we assumed that  $\mathbf{a} = 2$ , the variance of Pareto distribution is very large in these scenarios. Therefore, the average rate of generated traffic during a high-state (respectively low-states) may deviate substantially from  $\delta_H$  (respectively  $\delta_L$ ). This means that the calculated fair rate may exhibit more fluctuations increasing the starvation of the head-node. When the number of nodes (i.e.  $N$ ) is small, the fair rate converges very fast. In this case, variations of the average rate of traffic generated by the dynamic source do not have a significant effect on starvation of the head-node.

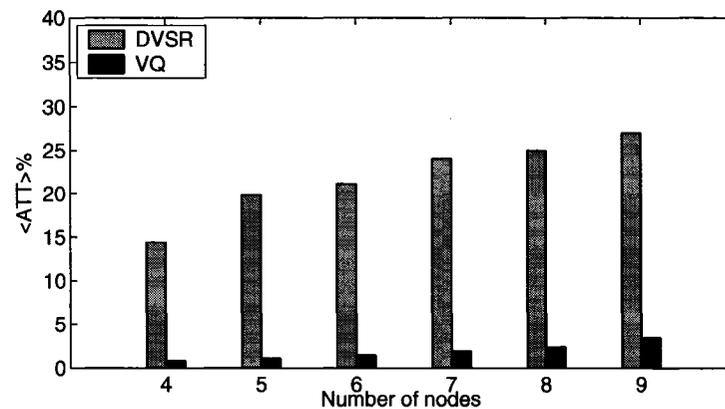


Figure 5.32: Comparison of  $\langle ATT \rangle$  in DVSR and VQ for various values of  $N$  for  $\delta_L = 0.05 \cdot C$  with Pareto traffic

However, when  $N$  is large, the convergence time increases and becomes comparable to  $T_H$  and  $T_L$ . In this case, variations of the average rate of traffic generated by the dynamic source can increase the rate of throttled traffic of the head-node. This can be seen by comparing Figures 5.29 and 5.31 where for  $N > 6$  the rate of throttled traffic of the head-node with Pareto traffic is slightly higher than the one with Poisson traffic.

It is noticeable that the rate of throttled traffic of the head-node with VQ is approximately an order of magnitude less than the ones with DVSR in all scenarios for different traffic distributions (see Figure 5.27 to Figure 5.32).

### 5.3.4 Starvation Time and ATT Upper bounds in VQ

In this section, we provide simulation results for ATT and starvation time of the head-node in different scenarios and compare these results with upper bounds derived in Propositions 5.2.3 and 5.2.4. Recall that in those propositions, we considered a scenario in which the steady state condition is reached and the fair rate of the head-node is  $\frac{C}{m_1}$ ,  $m_1 > 1$ , and its demand is  $D$ . Then, some nodes in the congestion span

increase their rates and, as a result, the steady state fair rate is decreased to  $\frac{C}{m_2}$ ,  $m_2 > m_1$ . To obtain simulation results for starvation time and accumulated throttled traffic of the head-node for various values of  $m_1$  and  $m_2$ , we consider a parking-lot scenario which consists of 9 nodes. It is assumed that the demand of node 9 (head-node) is  $D$  ( $D = 0.05 \cdot C, 0.1 \cdot C$ , and  $0.2 \cdot C$ ) and node 1 (the most upstream node) has traffic to send through node 9 with rate  $C$ ; other nodes do not initially have any traffic to send. Hence, the fair rate in the steady state condition is equal to  $C - D$  and nodes 1 and 9 send traffic with rates  $C - D$  and  $D$ , respectively. (In other words, we have  $\frac{C}{m_1} = C - D$  or, equivalently,  $m_1 = \frac{C}{C-D}$ .) This steady state condition is assumed to be the initial state of all following simulation scenarios.

For each  $D$ , we consider 7 scenarios with the same initial state where nodes 1 and 9 send traffic with rates  $C - D$  and  $D$ , respectively. In scenario  $i$ ,  $i = 1, \dots, 6, 7$ , each node 2 to  $2 + i - 1$  also wants to send traffic with rate  $C$  transiting the head-node (node 9). Therefore, the fair rate of node 9 is decreased to  $\frac{C}{m_2}$ . In fact, there is an  $m_2$  corresponding to each  $i$ . For instance, for  $D = 0.05 \cdot C$  and  $i = 1$ , the initial fair rate is equal to  $C - D = 0.95 \cdot C$  and  $m_1 = \frac{1}{0.95}$ . As  $i = 1$ , node 2 also wants to send traffic transiting node 9 with rate  $C$ . As a result, the fair rate in the steady state is decreased to  $\frac{0.95}{2} \cdot C$ , i.e.  $m_2 = \frac{2}{0.95}$ . Similarly, for  $i = 2, \dots, 6, 7$ , we have  $m_2 = \frac{3}{0.95}, \dots, \frac{7}{0.95}, \frac{8}{0.95}$ , respectively. In general, in scenario  $i$ ,  $i = 1, \dots, 6, 7$ , we have  $\frac{C}{m_2} = \max \left\{ \frac{C-D}{i+1}, \frac{C}{i+2} \right\}$ . For each  $D$ , we run the simulation for  $i = 1, 2, \dots, 7$  to obtain the ATT and the starvation time of node 9 for various  $m_2$ .

Figure 5.33 presents upper bounds and simulation results for starvation time of the head node for  $D = 0.05 \cdot C, 0.1 \cdot C$  and  $0.2 \cdot C$ . First, it can be seen that in this scenario, starvation time of the head node is a decreasing function of  $D$ . The reason is that, in each scenario  $i$ , it can be verified that the difference between the initial fair rate ( $\frac{C}{m_1} = C - D$ ) and the steady state fair rate ( $\frac{C}{m_2} = \max \left\{ \frac{C-D}{i+1}, \frac{C}{i+2} \right\}$ ) decreases with  $D$ ,  $D = 0.05, 0.1, 0.2$ . In other words, as  $D$  increases, the fair rate

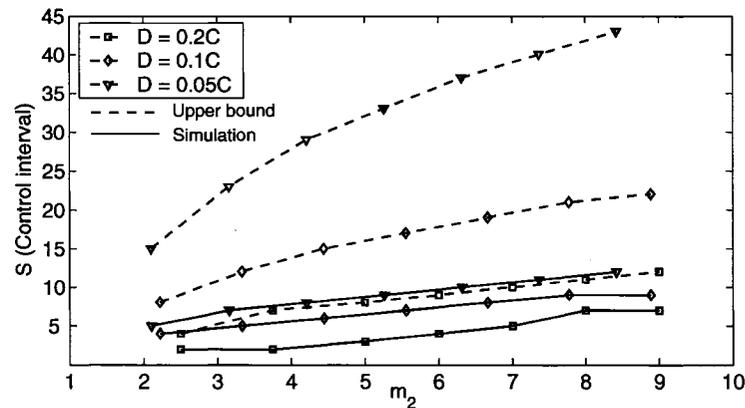


Figure 5.33: Comparing upper bounds on starvation time with simulation results

converges faster and starvation time decreases. For the same reason, starvation time is an increasing function of  $m_2$  for a given  $D$ .

Figure 5.34 shows upper bounds and simulation results for accumulated throttled traffic of the head node (corresponding to starvation times depicted in Figure 5.33) for  $D = 0.05 \cdot C$ ,  $0.1 \cdot C$  and  $0.2 \cdot C$ . One can notice that ATT upper bounds for  $D = 0.05 \cdot C$ ,  $0.1 \cdot C$  and  $0.2 \cdot C$  are very close. The reason is that in this scenario, the starvation time decreases with  $D$  as explained before. However, the throttled traffic of the head-node in each control interval during starvation time increases with  $D$ . Therefore, the accumulated throttled traffic of the head-node, which is the aggregate throttled traffic during starvation time, remains in the same range for  $D = 0.05 \cdot C$ ,  $0.1 \cdot C$ , and  $0.2 \cdot C$ .

It can be observed in Figures 5.33 and 5.34 that the derived upper bounds for starvation time and ATT are not tight. The reason is the fact that in deriving these upper bounds, we considered the worst case scenario for the decreasing rate of the fair rate (see Proposition 5.2.3). Moreover, it was assumed that  $D$ , which represents the demand of the head-node, is constant and does not change during starvation time. In fact,  $D$  was considered as the total arrival rate of traffic to the local buffers at the head-node. Note that during starvation time, the head-node cannot add its fair

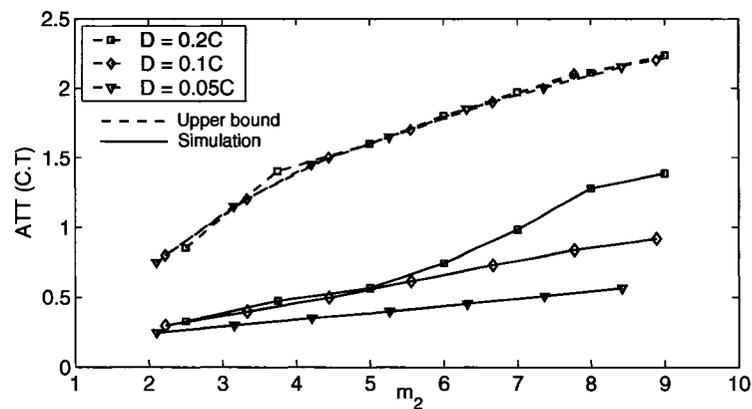


Figure 5.34: Comparing upper bounds on ATT with simulation results

share of traffic to the ring and as, a result, its throttled traffic is buffered in the local queues. In our simulation model, we considered the size of the local queue along with the arrival rate of traffic to the local queues to calculate the demand of the head-node. This, in fact, reduces the starvation time of the head-node by decreasing the fair rate faster.<sup>3</sup> For this reason, starvation times obtained from simulations were much less than corresponding upper bounds.

## 5.4 Conclusion

A new bandwidth allocation algorithm called Virtual Queuing (VQ) for RPR was proposed. VQ is based on a virtual queuing model of each node and aims at stabilizing the virtual queue while maximizing the bandwidth utilization. We studied the performance of the existing RPR bandwidth allocation algorithms and observed that they suffer from a number of performance limitations. In particular, we showed that the RPR standard bandwidth allocation algorithms exhibit oscillations and slow

<sup>3</sup>It is worth mentioning that in our simulation model, the head-node can increase its demand beyond the arrival rate of traffic to its local queues when it has back-logged (throttled) traffic due to starvation. As a result, starvation time is decreased and throttled traffic of the head node can gradually be added to the ring whenever bandwidth is available.

convergence times. Furthermore, we evaluated the performance of DVSR, which is an RPR bandwidth allocation algorithm, through simulation results. It was shown that in dynamic traffic scenarios, the overestimation of the fair rate in DVSR may result in starvation of the head-node in a congestion span. However, the VQ algorithm adjusts the fair rate more precisely when traffic variations exist in order to reduce starvation of the head-node. Our simulation results indicate that VQ performs significantly better than DVSR in a dynamic traffic environment in terms of accumulated throttled traffic of the head-node. We also derived analytical upper bounds on starvation times and accumulated throttled traffic of the head-node during a starvation period when the fair rate of the head-node is decreasing. We then verified the upper bounds in a number of scenarios by simulation results. The VQ algorithm has also a lower computational complexity compared to DVSR. VQ is an efficient algorithm with remarkable fairness characteristics which can be used as an attractive alternative bandwidth allocation algorithm for RPR.

# Chapter 6

## Concluding Remarks

This thesis was concerned with different aspects of bandwidth allocation and fairness in resilient packet rings. In this chapter, we briefly review the contributions of this thesis and suggest some directions for future research work.

### 6.1 Summary of Contributions

In this thesis, we first proposed a new fairness reference model for RPR (Chapter 3). We argued that there is no specific recommendation in the RPR standard on how each node should distribute its share of bandwidth among its local flows. This means that each node is allowed to employ any proprietary policy for allocating its share of bandwidth to its local flows as long as the ring-level fairness among the nodes is maintained. We showed that the existing RPR fairness reference model dictates to all nodes how they should distribute their bandwidth share among their local flows (Section 3.1.3). Our proposed model, however, is a general model and does not depend on any particular source behavior (Section 3.3). In fact, we showed that the previous proposed model was a special case of our model (Section 3.5). We also studied different policies which may be employed at each node to distribute the allocated bandwidth among local flows and, then, compared the performance of these

policies in a number of scenarios (Section 3.4). Some of the results from this part of research were presented in [10, 11].

As the second contribution of this research, we modeled the fair rate calculation process in the RPR standard bandwidth allocation algorithms (i.e. RPR-AM and RPR-CM) in the parking-lot scenario (Chapter 4). This analysis provided insights on dynamics of the RPR standard bandwidth allocation algorithms (Sections 4.1 and 4.2). These analytical models can be used to evaluate the performance of the RPR standard bandwidth allocation algorithms in terms of stability and convergence times. Using the developed models, we studied the effect of different system parameters on stability of the calculated fair rate in these algorithms. We also verified the accuracy of the developed models through simulation results (Sections 4.1.4 and 4.2.4). The results from this part of the work were presented in [12–17].

The third contribution of this proposal was designing a new bandwidth allocation algorithm called Virtual Queuing (VQ) for RPR (Chapter 5). VQ is based on a virtual queuing model of each node and aims at stabilizing the virtual queue while maximizing the bandwidth utilization. We studied the performance of existing RPR bandwidth allocation algorithms and observed that they suffer from a number of performance limitations (Section 5.1.1). In particular, we showed that the RPR standard bandwidth allocation algorithms exhibit oscillations and slow convergence times. Furthermore, we evaluated the performance of the DVSR algorithm through simulation results. It was shown that in some dynamic traffic scenarios, the over-estimation of the fair rate in DVSR may result in starvation of the head-node in a congestion span (Section 5.1.1). However, the VQ algorithm adjusts the fair rate more precisely when traffic variations exist in order to reduce starvation of the head-node. Our simulation results indicated that VQ performs significantly better than DVSR in terms of accumulated throttled traffic of the head-node in a dynamic traffic environment (Section 5.3.3). We also derived analytical upper bounds on starvation

times and accumulated throttled traffic of the head-node during a starvation period when the fair rate of the head-node is decreasing (Section 5.2.6). We then verified the upper bounds in a number of scenarios by simulation results (Section 5.3.4). The VQ algorithm has also a lower computational complexity compared to DVSR. VQ is an efficient algorithm with remarkable fairness characteristics which can be used as an attractive alternative bandwidth allocation algorithm for RPR. We presented some of the results from this part of the work in [18].

## 6.2 Future Work

In Chapter 3, a number of source behaviors, which can be employed at an RPR node, were introduced. We also showed that according to the RIAMM fairness model, each node is allowed to implement its own source behavior to distribute its bandwidth share among its local flows. A possible future work is investigating other possible source behaviors with different objective functions which can be employed at an RPR node.

In Chapter 4, we modeled the fair rate calculation process in the RPR standard bandwidth allocation algorithm for the parking-lot scenario. A possible direction for future work is to investigate this problem in a general case for an arbitrary traffic scenario where all nodes may send traffic to multiple destinations. We also assumed that the transit buffer is implemented in single-queue mode and that the ring fairness round-trip time is less than the length of control interval. This problem can also be analyzed for dual-queue transit buffer and for scenarios where the ring fairness round-trip time is longer than the length of the control interval. In this problem, we also considered a deterministic traffic scenario where all nodes have infinite traffic to send to the hub. An interesting problem is to extend this analysis to stochastic traffic scenarios.

In Chapter 5, we studied the performance of different RPR bandwidth allocations algorithms using Poisson and Pareto traffic. It is also desirable to study the performance of these algorithms in more realistic traffic scenarios. No major work has been done on this issue and it can be an interesting problem for further work.

# Appendix A

## Stability of Developed Analytical Models for Resilient Packet Ring Aggressive Mode

In this appendix, we study some properties of the analytical models presented in Section 4.1. We use the *equivalent gains* method described in [71] to analyze stability conditions of our models developed as nonlinear systems. This method is an intuitive technique where a nonlinear component is replaced by an equivalent gain and, then, the stability of the linear system with the equivalent gain is examined. In the following, a brief description of the equivalent gains method is provided.

**Definition A.0.1** (p. 551 in [71]). A function  $\phi(x)$ , with a scalar input and a scalar output, is assumed to belong to the sector  $[k_1, k_2]$  if for all inputs  $x$  we have

$$k_1 \cdot x^2 \leq \phi(x) \cdot x \leq k_2 \cdot x^2, \quad (\text{A.0.1})$$

where  $k_1$  and  $k_2$  are allowed to be  $-\infty$  or  $+\infty$ . Note that  $k_1$  and  $k_2$  do not necessarily represent the minimum and maximum slopes of  $\phi(x)$ .

To analyze stability of a nonlinear system using the equivalent gains method, first, the corresponding  $k_1$  and  $k_2$  of the nonlinear block of the system ( $\phi(x)$ ) should be determined. Then, the nonlinear block should be replaced by a linear gain  $k_{eq} \in$

$[k_1, k_2]$ . Finally, the stability of the obtained linear system is examined at each point in the interval  $[k_1, k_2]$ .

## A.1 Stability Analysis of Model without Link Propagation Delay

In this section, we analyze the stability condition of the calculated fair rate in the model without the propagation delay. In Section 4.1.2, we showed that the fair rate in the model without the link propagation delay can be calculated by the nonlinear difference equation in Equation (4.1.8). In the following proposition, we investigate the condition with respect to which the fair rate calculation system is stable and the convergence time of the calculated fair rate is finite.

**Proposition A.1.1.** *For a given  $n$ , the nonlinear system corresponding to the fair rate calculation of the model without the link propagation delay is stable when  $\alpha < \frac{2}{n}$ .*

*Proof.* The nonlinearity in the system depicted in Figure 4.2 is  $\phi(x) = (x)^+ = \max\{0, x\}$ . One can verify that the function  $\phi(x) = \max\{0, x\}$  belongs to the sector  $[0, 1]$ , i.e.,  $k_1 = 0$  and  $k_2 = 1$ . Using the equivalent gain method, the stability condition of the nonlinear system depicted in Figure 4.2 can be analyzed by examining the stability of the equivalent linear system for all  $k_{eq} \in [0, 1]$ . Replacing the nonlinear block in Figure 4.2 by a linear gain  $k_{eq}$  yields the equivalent linear system with the following transfer function:

$$H_{k_{eq}}(z) = \frac{\alpha \cdot k_{eq}}{1 + (\alpha \cdot (k_{eq} \cdot (n - 1) + 1) - 1) \cdot z^{-1}}. \quad (\text{A.1.1})$$

This system has a real pole at  $z = 1 - \alpha \cdot (k_{eq} \cdot (n - 1) + 1)$  and is stable if  $-1 < z < 1$ . Combining the inequalities  $0 \leq k_{eq} \leq 1$ , and  $0 < \alpha < 1$ , we can derive the following:

$\alpha \leq \alpha \cdot (k_{eq} \cdot (n - 1) + 1) \leq \alpha \cdot n$ . Therefore,  $1 - \alpha \cdot n \leq z \leq 1 - \alpha$  and the linear system is stable if  $\alpha < \frac{2}{n}$ . This means that the equivalent linear system is stable for all  $k_{eq} \in [0,1]$  if  $\alpha < \frac{2}{n}$ . Hence, the nonlinear system depicted in Figure 4.2 is stable when  $\alpha < \frac{2}{n}$ .  $\square$

## A.2 Stability Analysis of Model with Link Propagation Delay

It was shown in Section 4.1.3 that the fair rate calculation in the system with the propagation delay can be modeled by a second-order nonlinear discrete-time filter. The approach used to analyze the stability of the model without the propagation delay can be applied to this system as well in order to determine its stability conditions: After substituting Equation (4.1.14) in Equation (4.1.15), the fair rate calculation is modeled as a second-order nonlinear filter with a unit step function as its input. Then, each nonlinear block  $i$  (i.e.  $\phi(x) = (x)^+ = \max\{0, x\}$ ) is replaced by a linear gain  $k_{eq}^i$ . Finally, the stability of the equivalent linear system is investigated for all  $k_{eq}^i \in [0,1]$ . However, there are  $n$  nonlinear blocks in Equation (4.1.14) and the stability analysis of such a system would be very complicated. In the following, we only investigate the stability of the system with the propagation delay for  $k_{eq}^i = 1, \forall i$ . In fact, we only analyze the stability of the corresponding linear system of the model with the propagation delay. Note that as mentioned in Section 4.1.3, by implementing the analytical model in MATLAB, we observed that the stability domain of the second-order nonlinear system (with respect to  $\alpha$  for a fixed  $n$ ) is similar to its corresponding linear system which is obtained by bypassing all nonlinear blocks.

By substituting Equation (4.1.14) in Equation (4.1.15) and replacing each nonlinear block  $i$  by a linear gain  $k_{eq}^i = 1$  we have

$$F_1(k) = (1 - \alpha) \cdot F_1(k - 1) + \alpha \cdot \left[ \left( 1 - (n - 1) \cdot \frac{d}{T} \right) \cdot (u(k) - (n - 1) \cdot F_1(k - 1)) + \frac{d}{T} \cdot \sum_{j=1}^{n-1} (u(k) - (n - j) \cdot F_1(k - 2) - (j - 1) \cdot F_1(k - 1)) \right], \quad (\text{A.2.1})$$

where  $u(k)$  is the unit step function as the input of the second-order discrete-time filter. The transfer function corresponding to the linear difference equation in Equation (A.2.1) is  $H_d(z) = \frac{N(z)}{D(z)}$  where  $N(z) = \alpha \cdot \left( 1 + (n - 1) \cdot (n - 2) \cdot \frac{d}{2T} \right)$  and

$$D(z) = 1 - \left( (1 - \alpha \cdot n) + \alpha \cdot n \cdot (n - 1) \cdot \frac{d}{2T} \right) \cdot z^{-1} + \alpha \cdot n \cdot (n - 1) \cdot \frac{d}{2T} \cdot z^{-2}. \quad (\text{A.2.2})$$

Note that Equation (A.2.1) represents the difference equation of the response of  $H_d(z)$  to the unit step function. We also have  $H(z) = H_d(z) \Big|_{d=0}$ , where  $H(z)$  is the transfer function described in Equation (4.1.9).

For a fixed  $n$  and  $d$ , let  $\alpha_n^*(d)$  be the supremum value of  $\alpha$  such that  $\forall \alpha < \alpha_n^*(d)$ ,  $H_d(z)$  is stable, i.e., its poles are within the unit circle. For  $H(z)$ , we have  $\alpha_n^*(0) = \frac{2}{n}$  according to Inequality (4.1.10).

**Proposition A.2.1.** *For a fixed  $n$  and any  $d > 0$  we have  $\alpha_n^*(0) < \alpha_n^*(d)$ .*

*Proof.*  $H_d(z)$  has two poles as follows:

$$z_1 = \frac{1}{2} \cdot \left( 1 - \gamma + \gamma \cdot \frac{\delta}{2} + \sqrt{g(\gamma)} \right),$$

and

$$z_2 = \frac{1}{2} \cdot \left( 1 - \gamma + \gamma \cdot \frac{\delta}{2} - \sqrt{g(\gamma)} \right),$$

where  $\gamma \equiv \alpha \cdot n$ ,  $\delta \equiv (n - 1) \cdot \frac{d}{T} < 1$ , and  $g(\gamma) \equiv (1 - \gamma + \gamma \cdot \frac{\delta}{2})^2 - 2 \cdot \gamma \cdot \delta$ .  $H_d(z)$  is stable when  $|z_1| < 1$  and  $|z_2| < 1$ . For a given  $n$  and an arbitrary  $d > 0$ , the

poles may be a real or a conjugate pair depending on the sign of  $g(\gamma)$ . In order to determine the sign of  $g(\gamma)$  let us solve the equation  $g(\gamma) = 0$  for  $\gamma$ . It has two real poles:  $\gamma_1 = (\frac{\delta}{2} + 1 - \sqrt{2 \cdot \delta}) / (\frac{\delta}{2} - 1)^2$  and  $\gamma_2 = (\frac{\delta}{2} + 1 + \sqrt{2 \cdot \delta}) / (\frac{\delta}{2} - 1)^2$ , with  $\gamma_1 < \gamma_2$ . The following three cases are distinguished based on the position of  $\gamma$  with respect to  $\gamma_1$  and  $\gamma_2$ :

1)  $\gamma < \gamma_1$ : In this case  $g(\gamma) > 0$  and both poles are real with  $z_1 < z_2$ . As  $\delta < 1$ , one can check that  $\gamma_1 < 1$ , thus  $\gamma < 1$ . Consequently, we can show by inspection that  $1 - \gamma + \gamma \cdot \frac{\delta}{2} > 0$  and, as a result,  $z_2 > 0$ . Since  $z_1$  is an upper bound of  $z_1$  and  $z_2$ , and  $z_1 < 1 - \gamma + \gamma \cdot \frac{\delta}{2} < 1 - \frac{\gamma}{2} < 1$ , both poles are within the unit circle and the system is always stable.

2)  $\gamma_1 < \gamma < \gamma_2$ : In this case,  $g(\gamma) < 0$  and  $H_d(z)$  has two conjugate poles such that  $|z_1|^2 = |z_2|^2 = \gamma \cdot \frac{\delta}{2} = \alpha \cdot n \cdot \frac{\delta}{2}$ . Therefore,  $H_d(z)$  is stable when  $\alpha \cdot n \cdot \frac{\delta}{2} < 1$  which means that  $\alpha_n^*(d) = \frac{2}{n \cdot \delta}$ . As  $\delta < 1$ , we have  $\alpha_n^*(d) > \alpha_n^*(0) = \frac{2}{n}$ .

3)  $\gamma > \gamma_2$ : In this case  $g(\gamma) > 0$  and both poles are real. As  $\gamma > \gamma_2$ , we can show that  $(1 - \gamma + \gamma \cdot \frac{\delta}{2}) < 0$  and  $z_2 < z_1 < 0$ . This means that  $H_d(z)$  is stable as long as  $z_2$  is inside the unit circle. We have  $0 > z_2 > 1 - \gamma + \gamma \cdot \frac{\delta}{2} > 1 - \gamma = 1 - \alpha \cdot n$ . Note that  $z = 1 - \alpha \cdot n$  is the pole of  $H(z)$ . In other words, for some value of  $\alpha$ , the pole of  $H(z)$  is outside the unit circle while  $z_2$  is still inside the unit circle. That is,  $\alpha_n^*(d) > \alpha_n^*(0)$ .  $\square$

**Proposition A.2.2.** *For a given  $n$ , the maximum value of  $\alpha_n^*(d)$  for  $H_d(z)$  is achieved when  $(n - 1) \cdot d = 0.5 \cdot T$ .*

*Proof.* Recall that for any  $\gamma < \gamma_1$ , both poles of  $H_d(z)$  are within the unit circle. As a result, the determination of  $\alpha_n^*(d)$  depends on the position of  $\gamma$  with respect to  $\gamma_2$ . We distinguish two cases as follows:

1)  $\delta < 0.5$ : We prove that for  $\delta < 0.5$ ,  $\alpha_n^*(d)$  is an increasing function of  $\delta$ . One can check that when  $\delta < 0.5$ , we have  $\gamma_2 < 4$  and, consequently, the inequality

$\alpha \cdot n \cdot \frac{\delta}{2} < 1$  (stability condition of  $H_d(z)$ ) holds for any  $\gamma_1 < \gamma < \gamma_2$ . In other words,  $H_d(z)$  is stable for any  $\gamma < \gamma_2$ . When  $\gamma > \gamma_2$ , we showed that  $z_1$  and  $z_2$  are real and  $z_2 < z_1 < 0$ . Therefore,  $H_d(z)$  is stable as long as  $z_2$  remains within the unit circle. From the definition of  $\alpha_n^*(d)$ , one can notice that in this case  $\alpha_n^*(d)$  is obtained by solving the equation  $z_2 = -1$  (as for any  $\alpha < \alpha_n^*(d)$ ,  $z_2$  remains within the unit circle). This yields  $\alpha_n^*(d) = \frac{2}{(1-\delta) \cdot n}$ . Hence, for  $\delta < 0.5$ ,  $\alpha_n^*(d)$  is an increasing function of  $\delta$ . Figure A.1 shows the locations of  $H_d(z)$  poles due to variations of  $\gamma$  in an example with  $\delta < 0.5$ . It can be seen that  $z_2$  is real when it falls outside the unit circle for some  $\gamma > \gamma_2$ .

2)  $\delta > 0.5$ : We prove that when  $\delta > 0.5$ ,  $\alpha_n^*(d)$  is a decreasing function of  $\delta$ . For  $\delta > 0.5$ , we can check that  $\gamma_2 > 4$ . In this case, the inequality  $\alpha \cdot n \cdot \delta < 2$  (stability condition of  $H_d(z)$ ) is satisfied only for  $\gamma$  such that  $\gamma_1 < \gamma < \gamma_2$  and  $|z_1|^2 = |z_2|^2 = \gamma \cdot \frac{\delta}{2} < 1$ . As a result, for any  $\gamma = \alpha \cdot n < \frac{2}{\delta}$ , both poles are within the unit circle. From definition of  $\alpha_n^*(d)$  it follows that  $\alpha_n^*(d) = \frac{2}{\delta \cdot n}$ , which is a decreasing function of  $\delta$ . Figure A.2 depicts the locations of  $H_d(z)$  poles due to variations of  $\gamma$  in an example with  $\delta > 0.5$ . It can be seen that  $z_1$  and  $z_2$  are conjugate when they fall outside the unit circle for some  $\gamma_1 < \gamma < \gamma_2$ .

As  $\alpha_n^*(d)$  is an increasing function of  $\delta$  for  $\delta < 0.5$  and it is a decreasing function of  $\delta$  for  $\delta > 0.5$ , the maximum value of  $\alpha_n^*(d)$  is achieved when  $\delta = (n-1) \cdot \frac{d}{T} = 0.5$ .  $\square$

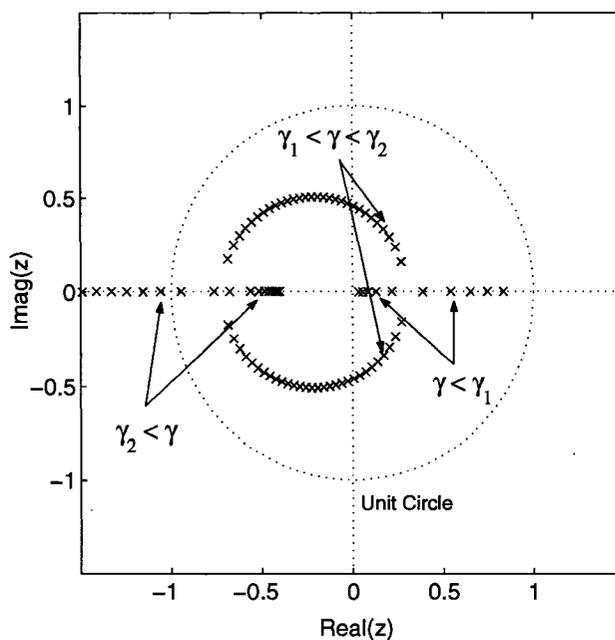


Figure A.1: Locations of  $H_d(z)$  poles in the  $z$ -plane due to variations of  $\alpha$  for  $n = 8$  and  $d = 0.05 \cdot T$ .

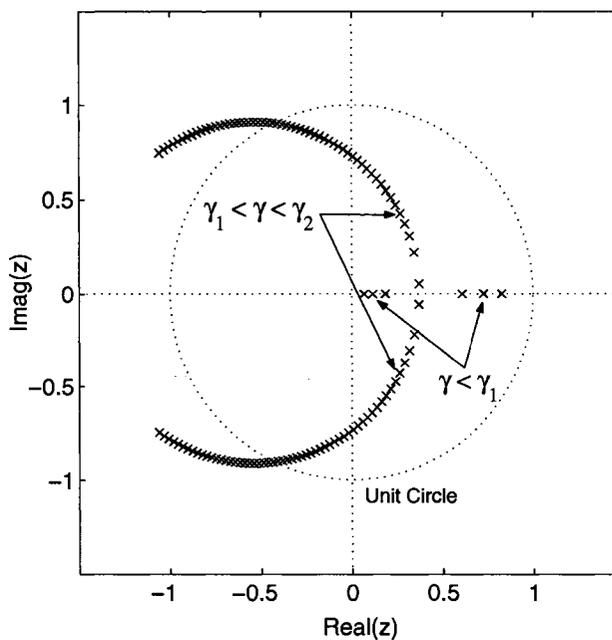


Figure A.2: Locations of  $H_d(z)$  poles in the  $z$ -plane due to variations of  $\alpha$  for  $n = 8$  and  $d = 0.1 \cdot T$ .

# Appendix B

## Upperbounds for Characterizing Fairness Properties of VQ Algorithm

### B.1 Proof of Proposition 5.2.3

Assume that the congestion span is in the steady-state in control interval  $k$  and  $F_n(k) = \frac{C}{m_1}$ . In control interval  $k + 1$ , some of input-limited nodes in the congestion span increase their rates and, hence,  $C < E^l(k+1) + E^R(k+1) \leq C + \min\{F_n(k), D\}$ . As  $C < E^l(k+1) + E^R(k+1)$ , we have

$$\frac{C - E^l(k+1)}{E^R(k+1)} < \frac{C}{E^l(k+1) + E^R(k+1)}, \quad (\text{B.1.1})$$

and, therefore, the rate increase/decrease factor is upper-bounded as follows:

$$f(k+1) \leq \frac{C}{E^l(k+1) + E^R(k+1)}. \quad (\text{B.1.2})$$

**Case 1:** If  $E^l(k+1) + E^R(k+1) < C + \min\{F_n(k), D\}$ , as we have assumed that  $E^R(k+1) \neq 0$ , the head-node is only starved at the current control interval according to Proposition 5.2.2. As a result,  $S = 1$ .

**Case 2:** If  $E^l(k+1) + E^R(k+1) = C + \min\{F_n(k), D\}$ , the head-node is completely starved in control interval  $k + 1$  and the fair rate is decreased. In control interval

$k + 2$ , if  $E^I(k + 2) + E^R(k + 2) < C + \min\{F_n(k + 1), D\}$ , node  $n$  will not be starved in the next control interval according to Case 1. Therefore, the worst case is when node  $n$  is completely starved in all control intervals up to control interval  $k + S$  where  $F_n(k + S) \leq \frac{C}{m_2}$ . The fair rate of node  $n$  in control interval  $k + i$  is given by

$$F_n(k + i) = \underbrace{f(k + i) \cdots f(k + 2) \cdot f(k + 1)}_i \cdot F_n(k), \quad 1 \leq i \leq S, \quad (\text{B.1.3})$$

where

$$f(k + i) \leq \frac{C}{C + \min\{F_n(k + i - 1), D\}}. \quad (\text{B.1.4})$$

To derive an upper-bound for  $S$ , we consider the following cases:

**Case 2-A:** If  $D \geq \frac{C}{m_1}$ , we have  $\min\{F_n(k + i), D\} = F_n(k + i)$ ,  $1 \leq i \leq S$  (as the fair rate is decreasing) and, hence,

$$f(k + i) \leq \frac{C}{C + F_n(k + i - 1)}, \quad 1 \leq i \leq S. \quad (\text{B.1.5})$$

For  $F_n(k + 1) = f(k + 1) \cdot F_n(k)$ , with  $F_n(k) = \frac{C}{m_1}$ , we have

$$F_n(k + 1) \stackrel{(\text{B.1.5})}{\leq} \frac{C}{C + F_n(k)} \cdot F_n(k) = \frac{C}{m_1 + 1}. \quad (\text{B.1.6})$$

Similarly, for  $F_n(k + 2) = f(k + 2) \cdot F_n(k + 1)$ , we have

$$\begin{aligned} F_n(k + 2) &\stackrel{(\text{B.1.5})}{\leq} \frac{C}{C + F_n(k + 1)} \cdot F_n(k + 1) \\ \implies F_n(k + 1) &\geq \frac{C}{C - F_n(k + 2)} \cdot F_n(k + 2) \\ \implies \frac{C}{C - F_n(k + 2)} \cdot F_n(k + 2) &\stackrel{(\text{B.1.6})}{\leq} \frac{C}{m_1 + 1} \\ \implies F_n(k + 2) &\leq \frac{C}{m_1 + 2}. \end{aligned}$$

In general, for  $F_n(k + i) = f(k + i) \cdot F_n(k + i - 1)$ ,  $1 \leq i \leq S$ , we have

$$F_n(k + i) \leq \frac{C}{m_1 + i} \quad (\text{B.1.7})$$

Note that we have assumed that  $F_n(k + S) \leq \frac{C}{m_2}$ . Hence, an upper bound for  $S$  is obtained by solving  $\frac{C}{m_2} \leq \frac{C}{m_1 + S}$  which yields

$$S \leq \lceil m_2 - m_1 \rceil, \quad (\text{B.1.8})$$

where  $\lceil \cdot \rceil$  is the ceiling function.

**Case 2-B:** If  $D \leq \frac{C}{m_2}$ , we have  $\min \{F_n(k + i), D\} = D$ ,  $1 \leq i \leq S$ . In this case, we have

$$f(k + i) \leq \frac{C}{C + D}, \quad 1 \leq i \leq S. \quad (\text{B.1.9})$$

It then follows from Equation (B.1.3) that  $F_n(k + S) \leq \left(\frac{C}{C + D}\right)^S \cdot F_n(k)$ . As we have assumed that  $F_n(k + S) \leq \frac{C}{m_2}$ , an upper bound for  $S$  is given by

$$S \leq \left\lceil \frac{\log \frac{m_2}{m_1}}{\log\left(1 + \frac{D}{C}\right)} \right\rceil. \quad (\text{B.1.10})$$

**Case 2-C:** If  $\frac{C}{m_2} < D < \frac{C}{m_1}$ , from  $F_n(k) = \frac{C}{m_1}$  to  $F_n(k + i) \leq D$  Case 2-B, and from  $F_n(k + i) = D$  to  $F_n(k + S) \leq \frac{C}{m_2}$  Case 2-A can be applied. This yields that

$$S \leq \left\lceil \frac{\log \frac{C/D}{m_1}}{\log\left(1 + \frac{D}{C}\right)} \right\rceil + \lceil m_2 - \frac{C}{D} \rceil. \quad (\text{B.1.11})$$

The upper bound of  $S$  follows from Equations (B.1.8), (B.1.10), and (B.1.11).

## B.2 Proof of Proposition 5.2.4

It is assumed in Proposition 5.2.3 that in the worst case, the head-node is completely starved and cannot add any traffic to the ring. Therefore, the throttled traffic of the head-node in control interval  $k$  during the starvation period is obtained as  $\sigma(k + i) = \min \{F_n(k + i - 1), D\} \cdot T$ ,  $1 \leq i \leq S$ . Note that the accumulated throttled traffic of the head-node during the starvation time is equal to  $ATT(k + S) - ATT(k) =$

$\sum_{i=1}^S \sigma(k+i)$ ,  $1 \leq i \leq S$ . In order to derive an upper bound for  $ATT(k+S) - ATT(k)$ , we consider the following cases.

i) When  $D \geq \frac{C}{m_1}$  (Case 2-A in Proposition 5.2.3), we have  $\min \{F_n(k+i-1), D\} = F_n(k+i-1)$ ,  $1 \leq i \leq S$ . Moreover, an upper bound for the fair rate of the head-node is given by (B.1.7) and, consequently, we have

$$\sum_{i=1}^S \sigma(k+i) = \sum_{i=1}^S F_n(k+i-1) \leq \sum_{i=0}^{\lceil m_2 - m_1 \rceil - 1} \frac{C \cdot T}{m_1 + i}. \quad (\text{B.2.1})$$

ii) When  $D \leq \frac{C}{m_2}$  (Case 2-B in Proposition 5.2.3), we have  $\min \{F_n(k+i-1), D\} = D$ ,  $1 \leq i \leq S$ . Therefore,

$$\sum_{i=1}^S \sigma(k+i) = S \cdot D \cdot T \leq \left\lceil \frac{\log \frac{m_2}{m_1}}{\log(1 + \frac{D}{C})} \right\rceil \cdot D \cdot T. \quad (\text{B.2.2})$$

iii) Similarly, for  $\frac{C}{m_2} < D < \frac{C}{m_1}$  (based on Case 2-C in Proposition 5.2.3), we have

$$\sum_{i=1}^S \sigma(k+i) \leq \left\lceil \frac{\log \frac{C/D}{m_1}}{\log(1 + \frac{D}{C})} \right\rceil \cdot D \cdot T + \sum_{i=0}^{\lceil m_2 - C/D \rceil - 1} \frac{C \cdot T}{C/D + i}. \quad (\text{B.2.3})$$

The upper bound on accumulated traffic of the head-node during starvation time is obtained from Equations (B.2.1), (B.2.2), and (B.2.3).

# Bibliography

- [1] W. J. Goralski, *SONET/SDH*. McGraw Hill, Third ed., 2002.
- [2] R. M. Metcalfe and D. R. Boggs, “Ethernet: Distributed packet switching for local computer networks,” *ACM Communications*, vol. 19, pp. 395–404, July 1976.
- [3] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley Longman Publishing Co. Inc., Third ed., 2004.
- [4] P. Yuan, V. Gambiroza, and E. Knightly, “The IEEE 802.17 media access protocol for high-speed metropolitan-area resilient packet rings,” *IEEE Network*, vol. 18, pp. 8–15, May-June 2004.
- [5] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun, “IEEE 802.17 resilient packet ring tutorial,” *IEEE Communications Magazine*, vol. 42, pp. 112–118, March 2004.
- [6] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, “Design, analysis, and implementation of DVSR: A fair, high performance protocol for packet rings,” *IEEE/ACM Transactions on Networking*, vol. 12, pp. 85–102, February 2004.
- [7] D. Wang, K. K. Ramakrishnan, and C. Kalmanek, “Congestion control in resilient packet ring,” in *Proceedings of the 12th International Conference on Network Protocols (ICNP’04)*, (Berlin, Germany), pp. 108–117, October 2004.
- [8] F. Alharbi and N. Ansari, “Low-complexity distributed bandwidth allocation for resilient packet ring networks,” in *Proceedings of IEEE Workshop on High*

- Performance Switching and Routing (HPSR2004)*, (Phoenix, AZ), pp. 277–281, April 2004.
- [9] P. Yue, Z. Liu, and J. Liu, “High performance fair bandwidth allocation algorithm for resilient packet ring,” in *Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA’03)*, (Xi’an, China), pp. 415–420, March 2003.
- [10] S. Khorsandi, A. Shokrani, and I. Lambadaris, “A new fairness model for resilient packet rings,” in *Conference Records of IEEE ICC’05*, (Seoul, Korea), May 2005.
- [11] S. Khorsandi, A. Shokrani, and I. Lambadaris, “Equal opportunity fairness in resilient packet rings,” in *Conference Record of IEEE ICC’06*, (Istanbul, Turkey), June 2006.
- [12] A. Shokrani, J. Talim, and I. Lambadaris, “Analysis of fair rate calculations in resilient packet ring aggressive mode,” *Accepted for publication in Elsevier Optical Switching and Networks (OSN) Journal*, November 2006.
- [13] A. Shokrani, I. Lambadaris, and J. Talim, “An analytical model for fair rate calculation in resilient packet rings,” in *Proceedings of IEEE GLOBECOM’05*, (Saint Louis, MO), December 2005.
- [14] A. Shokrani, I. Lambadaris, and J. Talim, “Modeling and stability analysis of fair rate calculation in resilient packet rings,” in *Proceedings of the International Conference on Broadband Networks (BROADNETS’05)*, (Boston, MA), October 2005.
- [15] A. Shokrani, I. Lambadaris, and J. Talim, “On modeling of fair rate calculation in resilient packet rings,” in *Proceedings of the 10th IEEE International Symposium on Computer Communications (ISCC’05)*, (Cartagena, Spain), June 2005.
- [16] A. Shokrani, I. Lambadaris, and J. Talim, “An accurate model for fair rate calculation in resilient packet rings,” in *Proceedings of the 18th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE’05)*, (Saskatoon, SK), May 2005.

- [17] A. Shokrani, J. Talim, and I. Lambadaris, "Modeling and analysis of fair rate calculation in resilient packet ring conservative mode," in *Conference Records of IEEE ICC'06*, (Istanbul, Turkey), June 2006.
- [18] A. Shokrani, S. Khorsandi, I. Lambadaris, and L. Khan, "Virtual Queuing: An efficient algorithm for bandwidth management in resilient packet rings," in *Conference Records of IEEE ICC'05*, (Seoul, Korea), May 2005.
- [19] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw Hill, First ed., 2000.
- [20] W. D. Farmer and E. E. Newhall, "An experimental distributed switching system to handle bursty computer traffic," in *Proceedings of ACM Symposium on Problems in the Optimization of Data Communication*, (Pine Mountain, GA), pp. 1–33, October 1969.
- [21] IEEE, "Token ring access method and physical layer specifications," IEEE Standard 802.5, 1989.
- [22] W. Bux, "Token-ring local-area networks and their performance," *Proceedings of the IEEE*, vol. 77, pp. 238–256, February 1989.
- [23] H. R. van As, "Media access techniques: The evolution towards terabit/s lans and mans," *Computer Networks and ISDN Systems*, vol. 26, pp. 603–656, March 1994.
- [24] W. A. Shay, *Understanding Data Communications and Networks*. Brooks/Cole, Third ed., 2004.
- [25] F. E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1043–1051, September 1989.
- [26] ANSI, "FDDI token ring media access control," ANSI Standard X3.139, 1987.
- [27] A. S. Tanenbaum, *Computer Networks*. Prentice Hall, Fourth ed., 2003.

- [28] M. A. Gallo and W. M. Hacock, *Computer Communications and Networking Technologies*. Brooks/Cole, 2002.
- [29] J. Pierce, "Network for block switching of data," *Bell Systems Technical Journal*, vol. 51, pp. 1133–1145, July-August 1972.
- [30] E. Hafner, Z. Nendal, and M. Tschanz, "A digital loop communication system," *IEEE Transactions on Communications*, vol. 22, pp. 877–881, June 1974.
- [31] L. Georgiadis, R. Guérin, and I. Cidon, "Throughput properties of fair policies in ring networks," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 718–728, December 1993.
- [32] W. Bux and M. Schlatter, "An approximate method for the performance analysis of buffer insertion rings," *IEEE Transactions on Communications*, vol. 31, pp. 50–55, January 1983.
- [33] L. Tassiulas and J. Joung, "Performance measures and scheduling policies in ring networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 576–584, October 1995.
- [34] L. Georgiadis, W. Szpankowski, and L. Tassiulas, "A scheduling policy with maximum stability region in ring networks with spatial reuse," *Queueing System Theory and Applications*, vol. 19, pp. 131–148, 1995.
- [35] I. Rubin and H. Wu, "Performance analysis and design of CQBT algorithm for a ring network with spatial reuse," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 649–659, August 1996.
- [36] I. Cidon and Y. Ofek, "MetaRing - a full duplex ring with fairness and spatial reuse," *IEEE Transactions on Communications*, vol. 41, pp. 110–120, January 1993.
- [37] G. Anastasi and L. Lenzi, "Performance evaluation of a MetaRing MAC protocol carrying asynchronous traffic," *Journal of High-Speed Networks*, vol. 6, pp. 35–52, January 1997.

- [38] I. Cidon, L. Georgiadis, R. Guérin, and Y. Shavitt, "Improvement fairness algorithms for rings with spatial reuse," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 190–204, April 1997.
- [39] H. R. van As, W. W. Lemppenau, H. R. Schindler, and P. Zafropulo, "CRMA-II: A MAC protocol for ring-based Gb/s LANs and MANs," *Computer Networks and ISDN Systems*, vol. 26, no. 6-8, pp. 831–840, 1994.
- [40] E. Zurfuh, R. Cideciyan, P. Dill, R. Heller, W. Lemppenau, P. Mueller, H. Schindler, and P. Zafropulo, "The ibm Zurich research laboratorys 1.13 Gb/s LAN/MAN prototype," *Computer Networks and ISDN Systems*, vol. 26, no. 2, pp. 163–184, 1993.
- [41] J. Schuringa, G. Remsak, and H. R. van As, "Performance comparison of ATMR and CRMA-II in Gbit/s-LANs," in *Conference Record of IEEE ICC'94*, (New Orleans, LA), pp. 716–720, May 1994.
- [42] J. Schuringa, G. Remsak, and H. R. van As, "Cyclic queuing multiple access (CQMA) for RPR networks," in *Proceedings of the 7th European Conference on Networks and Optical Communications (NOC2002)*, (Darmstadt, Germany), pp. 285–292, June 2002.
- [43] F. Davik, *Fairness aspects of buffer-insertion rings in general and resilient packet rings in particular*. Doctoral dissertation, University of Oslo, (Oslo, Norway), August 2005.
- [44] IEEE, "Resilient packet ring (RPR) access method and physical layer specifications," IEEE Standard 802.17, 2004.
- [45] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in *Proceedings of IEEE INFOCOM'96*, (San Francisco, CA), pp. 296–302, March 1996.
- [46] D. Schupke and A. Riedl, "Packet transfer delay comparison of a store-and-forward and a cut-through resilient packet ring," in *Proceedings of International Zurich Seminar on Broadband Communications (IZS)*, (Zurich, Switzerland), February 2002.

- [47] X. Zhou, D. Jin, and L. Zeng, "A novel fairness algorithm based on rate estimation in resilient packet ring," *Proceedings of SPIE Network Architectures, Management, and Applications*, vol. 5282, pp. 134–142, April 2004.
- [48] S. Gjessing and A. Maus, "A fairness algorithm for high-speed networks based on a resilient packet ring architecture," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, (Hammamet, Tunisia), pp. 279–284, April 2002.
- [49] Y. Robichaud and C. Huang, "Improved fairness algorithm to prevent tail node induced oscillations in RPR," in *Conference Record of IEEE ICC'05*, (Seoul, Korea), May 2005.
- [50] F. Davik, A. Kvalbein, and S. Gjessing, "Improvement of resilient packet rings fairness," in *Proceedings of IEEE GLOBECOM'05*, (Saint Louis, USA), November 2005.
- [51] X. Zhou, G. Shi, H. Fang, and L. Zeng, "Fairness algorithm analysis in resilient packet ring," in *Proceedings of the 2003 International Conference on Communication Technology (ICCT2003)*, (Beijing, China), p. 622624, April 2003.
- [52] D. Tsang and G. Suwala, "The Cisco SRP MAC layer protocol," IETF Networking Group, RFC 2892, August 2000.
- [53] L. Massouiliè and J. Roberts, "Bandwidth sharing: objectives and algorithms," in *Proceedings of IEEE INFOCOM'99*, (New York, NY), pp. 1395–1403, March 1999.
- [54] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness, and stability," *Journal of Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [55] B. Radunović and J. L. Boudec, "A unified framework for max-min and min-max fairness with applications," in *Proceedings of Allerton Conference on Communications, Control, and Computing*, (Allerton, IL), October 2002.
- [56] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, Second ed., 1992.

- [57] J. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, vol. COM-29, pp. 954–962, July 1981.
- [58] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Conference Record of IEEE ICC'95*, (Seattle, WA), pp. 1954–1963, June 1995.
- [59] S. Chong, S. Lee, and S. Kang, "A simple, scalable, and stable explicit rate allocation algorithm for max-min flow control with minimum rate guarantee," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 322–335, June 2001.
- [60] Y. Hou, S. Panwar, and H. Tzeng, "On generalized max-min rate allocation and distributed convergence algorithm for packet networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 401–416, May 2004.
- [61] E. Hahne, "Round-robin scheduling for max-min fairness in data networks.," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. July, pp. 1024–1039, 1991.
- [62] X. Wang and K. Kar, "Distributed algorithms for max-min fair rate allocation in ALOHA networks," in *Proceedings of Allerton Conference on Communications, Control, and Computing*, (Allerton, IL), October 2004.
- [63] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *Proceedings of IEEE INFOCOM'02*, (New York, NY), pp. 763–772, June 2002.
- [64] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, June 1993.
- [65] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.
- [66] A. Charny, "An algorithm for rate allocation in a packet-switching network with feedback," Master's thesis, Massachusetts Institute of Technology, (Cambridge, MA), April 1994.

- [67] C. Huang, H. Peng, and F. Yuan, "A deterministic bound for the access delay of resilient packet rings," *IEEE Communications Letters*, vol. 9, pp. 87–89, January 2005.
- [68] F. Davik, A. Kvalbein, and S. Gjessing, "An analytical bound for convergence of the resilient packet ring aggressive mode fairness algorithm," in *Conference Record of IEEE ICC'05*, (Seoul, Korea), May 2005.
- [69] S. Gjessing and F. Davik, "Performance evaluation of back-pressure fairness in RPR," in *Proceedings of the 7th European Conference on Networks and Optical Communications (NOC2002)*, (Darmstadt, Germany), p. 293300, June 2002.
- [70] R. Jain, D. Chiu, and W. Have, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Technical Report DEC-TR-301, Digital Institution Corporation, (Hudson, MA), September 1984.
- [71] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co. Inc., Second ed., 1997.