MAC PROTOCOLS FOR OPTICAL PACKET-SWITCHED WDM RINGS

Marcos Rogério Salvador

This research work was conducted within Dutch project FLAMINGO, and it was funded by the Dutch Technology Foundation STW, applied science division of NWO and technology programme of the Ministry of Economic Affairs of The Netherlands, and by the Centre for Telematics and Information Technology of the University of Twente, The Netherlands.

Centre for Telematics and Information Technology (CTIT)
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Graduation committee

Chairman/Secretary:	prof.dr. W. H. M. Zijm (University of Twente, NL)
Supervisor:	prof.dr.ir. I. G. M. M. Niemegeers (Tech. Univ. of Delft, NL)
Assistant supervisor:	dr.ir. S. Heemstra de Groot (University of Twente, NL)
Members:	prof.ir. A. C. van Bochove (University of Twente, NL)
	prof.ir. E. F. Michiels (University of Twente, NL)
	prof.ir. A. M. J. Koonen (Tech. Univ. of Eindhoven, NL)
	prof.dr.ir. L. J. M. Nieuwenhuis (University of Twente, NL)
	prof.dr. H. R. van As (Tech. University of Vienna, AT)
	prof.dr.ir. M. Pickavet (University of Gent, BE)

MAC protocols for optical packet-switched WDM rings / M. R. Salvador
Ph.D. dissertation - with references; with preface in English; 192 pages
University of Twente, Enschede, The Netherlands
ISSN 1381-3617 (CTIT Ph.D.-thesis series number 02-47)
ISBN 90-365-1862-8

Cover design by Diego Rios.

MAC PROTOCOLS FOR

OPTICAL PACKET-SWITCHED WDM RINGS

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
under the authority of the rector magnificus,
prof.dr. F. A. van Vught,
on account of the decision of the graduation committee,
to be publicly defended
on Friday, January 31, 2003 at 16:45.

by
Marcos Rogério Salvador
born on August 28, 1972
in Americana (S.P.), Brazil.

This dissertation has been approved by:

Supervisor: prof.dr.ir. I. G. M. M. Niemegeers

Assistant supervisor: dr.ir. S. Heemstra de Groot

To my beloved wife, Graciela,
for her love, care, understanding, and support
during this important and difficult period of our lives.

This is our achievement!

“I have yet to see any problem,
however complicated, which,
when you looked at it in the right way,
did not become still more complicated.”

-Poul Anderson

Preface

The ongoing convergence of applications, services, media, and specialised networks to the Internet is posing new challenges for network architectures in general. In existing network architectures electronics still plays a major role, even though interfaces and wires use optics. Nevertheless, electronics is already struggling to keep up with the current capacity demands of the Internet. To cope with the service demands of the Internet these architectures combine a multiplicity of technologies, and such a combination makes these architectures complex and difficult to manage and scale. Therefore, it is doubtful that existing network architectures will be able to meet the demands of the future Internet at reasonable complexity and cost, if at all.

Developments in optical networking technologies and the evolution in the Internet technologies represent a window of opportunity to move from the complex, inefficient, multi-layer electronics networking model to a much simpler, more efficient, two-layer optical networking model, in which IP lies almost directly on top of the optical layer.

The potential of such an optical Internet have attracted much attention of the research community and led to the design of many candidate architectures. Among them, ring networks using wavelength division multiplexing (WDM) and optical packet switching (OPS) are prominent because they combine the resilience and simplicity of the ring topology, the capacity, scalability, and transparencies of WDM, and the robustness, transparencies, and node bypassing capability of OPS.

Limitations in the state-of-the-art optical networking technologies, in particular, the lack of optical random access memory, and the characteristics of the Internet traffic, such as variable size packets, packet arrival burstness, non-uniform traffic distribution patterns, and volatile traffic patterns make the transport of Internet traffic in WDM rings that perform OPS challenging.

This work is concerned with medium access control (MAC) protocols for such ring networks. It focuses on the efficient, high performance transport of Internet traffic and proposes protocols that take the characteristics of the Internet traffic into consideration to achieve that.

Two complementary sub-classes of MAC protocols are investigated in this work: access control and access fairness. The access control protocols use the destination removal slotted ring technique for high performance, and they aim at the transport of variable size packets. The access fairness protocols use cyclic reservations to adapt to the traffic conditions proactively, and they superpose the access control protocols to enforce fairness.

To evaluate the performance and the effectiveness of the protocols in the transport of Internet-like traffic, this work uses data obtained in computer simulators developed exclusively for this work.

This dissertation is structured into seven chapters. Chapter 1 elaborates on the motivations of this work and defines precisely the goals to achieve. Chapters 2, 3, and partly 5 present the state-of-the-art on the subject. Chapters 4 and 5 introduce four access control protocols and two access fairness protocols. Chapter 6 evaluates comparatively the performance of the protocols. Chapter 7 elaborates on the obtained results and points to further investigations.

Acknowledgements

After several years of hard work, far from my wife, family, friends, and relatives, with ups and downs, I have achieved another goal of my life: the Ph.D. degree. Those who obtained the Ph.D. degree long time ago might have forgotten what a wonderful feeling it is. Those who have not obtained such a degree probably have no idea of what that means. Nevertheless, those who obtained the degree not long time ago understand what I mean. It is a mixture of emotions, such as relief, happiness, achievement, emptiness, and pain.

Besides the degree, I have achieved another goal of my life: to live abroad. Who could have imagined that I would ever have colleagues, or even friends, from countries such as The Netherlands, Belgium, Germany, Austria, France, Italy, Portugal, Spain, Argentina, Russia, Romania, Yugoslavia, Vietnam, Indonesia, China, Albania, India, England, Scotland, Morocco, Mexico, Egypt, United States, Canada ?. One can only imagine how vast the cultural information exchange in an environment like this is, and how such an environment enriches anyone's mind and soul. Certainly, the experiences I have had and what I have learned have changed me and the way I see things, have made me grow further as person. And that is invaluable to me.

I can only thank the people and the institutions that made it possible for me to achieve my dreams, and I will start from Dr.ir. Sonia Heemstra de Groot, my daily supervisor, and Prof.dr.ir. Ignas Niemegeers, my supervisor.

I remember when Sonia called me on a Sunday, at 6AM in Brazil -it was my own fault by the way- to find out a little about me, and I had just gone to bed after coming from a party. Her understanding from that moment until now is remarkable. As a matter of fact, I am not sure you would be reading these acknowledgements right now if another person had called me on the phone that day.

From the beginning Sonia let me free to find my way, only making sure that I would not deviate too much from the target. I might have busted my head against the wall a few times, but that just made me grow further, and I appreciated that immensely.

Sonia's vivacity, strength, and happiness inspired me in so many ways and occasions that she has no idea. To me Sonia is more than a supervisor; she is a friend, who sometimes acts as a mother too.

Ignas has always been very friendly, and I reckon that his friendship was very important, as I cannot stand pure cold hierarchical relationships. Often Ignas and I would engage in talks about subjects as diverse as cars, history, tourism, politics, and, of course, work. His way of looking into the big picture inspired me many times.

My work would not have been the same without two people that, not coincidentally, I named the formal supporters of my graduation ceremony: Diptish, my project colleague and former officemate, and Helen, the secretary. Diptish's join to the project gave me a boost towards the completion of my work, and our discussions contributed greatly to my work. Diptish also helped me improve both my spoken and written English. Helen helped me in so many ways and occasions that I cannot precise. She was always open and smiling -although she might have wished to kick me out of her office some times, and that made the work environment more pleasant.

I am happy to have shared the office with Minh after Diptish left. Minh helped me many times with the Dutch language and often took me deeper into the Dutch culture. I appreciated his sense of humour and his business view of things.

My adventure would not have been possible without the funding provided by STW and CTIT. To that respect I am also indebted to Prof.dr. Wanderley Lopes de Souza, from Federal University of São Carlos (UFSCar), Brazil, my supervisor during my Master of Science studies. Wanderley believed in my potential and provided me with the support and the recommendation necessaries for my acceptance.

I would like to thank the members of the graduation committee for the time and work to review my dissertation and attend its defence.

I would like also to express my gratitude to Prof.dr. Helio Waldman, from the University of Campinas (Unicamp), Brazil. Prof. Waldman provided me with the opportunity to find out about what was going on in my field of research in Brazil, and thanks to him I could establish a contact network in my own country. Prof. Waldman himself and his students welcomed me very warmly in their lab and provided me with an exciting work environment that kept me working during one of my stays in Brazil.

I met many people in The Netherlands, some of which have become colleagues or friends. In particular, I would like to thank those without which I would not have stood a chance here: Clever and Kelen, Ciro, João Paulo and Patty, Giancarlo and Renatinha, José Gonçalves and Sandra, Ana Paula and Pedro, Audrey, Alby, Leo, Amina and Ben, Vlora, Lijun, Gloria, Marlous, Hommad, Rachid, Natasa, Alex Slingerland, Valerie, Val Jones, Maarten, Remco, Luis, Diego, Pedro D'argenio, Victor Nicola, Minh, David, and Nikolay.

Besides the assistance from my supervisors, I could count with the great help from João Paulo, Patty, Diptish, and Minh to improve the legibility of the dissertation, and with the design skills of Diego to create the cover of the dissertation book.

Many people in Brazil are responsible for the happiness I feel right now. The most important of them is, without doubt, my wife, Graciela. Graciela stood up during these years of distance relationship and "shuttles" between São Paulo and Amsterdam, even though that meant sacrifices and painful times. I hope I will some day be able to pay her back for all her love, care, understanding, and support.

Graciela’s family and relatives should not be forgotten for their motivation and support. Graciela’s parents supported and motivated me even knowing that it was tough for their daughter.

My relatives, my family and my parents, in particular, deserve special thanks. They missed me and suffered with that, but still kept motivating me to pursue my dream. As a matter of fact, had my parents not opened my eyes to the big picture -once again-, am I not sure I would have quit my job to come to The Netherlands.

I received great support and strength from my friends Dudão, Fabião, Jaimão, Alessandro, Robson boi, Barison, Rafinha, Carlocha, Emerson PV, Dani, Valeria, and PSampaio.

Table of contents

Chapter 1:
Introduction 1

1.1 Background 1
1.1.1 MOPS rings 3
1.1.2 Medium access techniques and protocols 6
1.2 Motivations of this work 10
1.3 Objectives 13
1.4 Organization 14

Chapter 2:
MOPS ring architectures 15

2.1 Pipeline 15
2.2 BORN 18
2.3 MAWSON 21
2.4 HORNET 22
2.5 RINGO 24
2.6 FLAMINGO 25
2.7 Discussion 27

Chapter 3:
Existing MAC protocols for MOPS rings 31

3.1 Empty slot contention/collision avoidance (ESCA) 31
3.2 BORN 31
3.3 Synchronous round-robin (SRR) 32
3.4 Request/Allocation protocol (RAP) 34
3.5 Multitoken interarrival time (MTIT) 35
3.6 Carrier sense multiple access with collision avoidance (CSMA/CA) 36
3.7 Discussion 38

Chapter 4:
Access control protocols 45

4.1 Preliminaries 45

4.2 Conflict-free protocols 49
4.2.1 PAT 49
4.2.2 SPT+SC 53
4.3 Contention protocols 60
4.3.1 SPT+R 60
4.3.2 SPT+P 64
4.4 Channel multiplicity 67
4.5 Error handling 69
4.5 Discussion 70

Chapter 5:
Access fairness protocols 73

5.1 Fairness definition 73
5.2 Existing protocols 74
5.2.1 Global protocols 74
5.2.1.1 MetaRing 74
5.2.1.2 ATMR 76
5.2.2 Local protocols 77
5.2.2.1 Fault-tolerant distributed local protocol 77
5.2.2.2 Distributed local scheduling with partial information 79
5.2.3 Discussion 82
5.3 Proactive fairness protocols 82
5.3.1 Global cyclic reservation 83
5.3.2 Local cyclic reservation 88
5.4 Discussion 94

Chapter 6:
Performance evaluation 97

6.1 Goals 97
6.2 Approach 97
6.3 Simulators 98
6.4 Traffic characteristics 100
6.5 Performance parameters of interest 102
6.6 Performance evaluation of the fairness protocols 103
6.6.1 Symmetric scenario 104
6.6.2 Asymmetric scenario 107
6.6.3 Worst-case scenario 108
6.7 Performance evaluation of the access control protocols 109
6.7.1 SAT 111
6.7.2 LCR 126

6.8 Discussion..... 145

Chapter 7:
Conclusions 147

7.1 General considerations..... 147
7.2 Main contributions and evaluation of the results 147
7.3 Topics for future research 149

References 151

Appendix A:
List of acronyms..... 161

Appendix B:
Synchronisation 165

Appendix C:
Group communication..... 169

Appendix D:
Addressing..... 171

Chapter 1
Introduction

This dissertation is concerned with high performance medium access control (MAC) protocols for multiple-wavelength optical packet-switched (MOPS) rings carrying Internet traffic. The focus of the dissertation is on access control protocol to transport variable size packets and access fairness protocols to enforce fairness under various traffic conditions.

This chapter provides an overview of this dissertation. It includes background information, explains the main motivations and objectives of this work, and describes how this dissertation is organized.

1.1 Background

The convergence of applications, services, media, and specialized networks to the Internet is changing the world and the way people do things. Although with limitations, distributed computing, teleconferencing, tele-education, listening to radio, watching television, and playing games over the Internet are all a reality today. In the near future, such limitations will disappear, and it will be possible to support more sophisticated applications over the Internet, such as virtual reality collaborative design work and tele-surgery.

For such a future Internet to become a complete reality, network infrastructures at all levels, and metropolitan area network (MAN) infrastructures, in particular, have to meet requirements such as:

- Scalability: as the Internet evolves applications running over it tend to become more diversified and sophisticated, consequently attracting more users. Nevertheless, growth in the number of users and higher sophistication of applications demand more network capacity. Therefore, the sustained evolution of the Internet depends on the sustained increase in the capacity of network infrastructures;
- Robustness: for sophisticated applications to move to the Internet network infrastructures should guarantee that failures rarely occur and that those applications will continue to work when failures do occur;
- Security: security is inherent to commercial and military applications, just to name a few. Therefore, the convergence of such applications to the Internet is conditioned to the provision of secure communications;
- Quality of service (QoS): applications have diverse requirements. Therefore, providing applications with services that meet their requirements is necessary for those applications to run smoothly and, ultimately, for user-satisfaction;

- Efficiency: efficiency in the utilisation of network resources reduces the need of over-provisioning and constant upgrades. Efficiency also contributes towards increasing the network performance and improving the QoS perceived by applications;
- Simplicity: network infrastructures should strive for simplicity. Simplicity brings many benefits, the ultimate being the contribution to the acceptance of these infrastructures and their underlying technologies.

Operational MAN infrastructures fail to meet many of the requirements listed above, and that happens for two primary reasons. First, the technologies underlying such infrastructures are essentially electronics, which has struggled to achieve bit rates of a few tens of gigabit per second (Gb/s). Second, there is no single networking technology that can provide services such as QoS, security, and virtual private networking (VPN). To provide such services, MAN infrastructures combine technologies such as the Internet protocol (IP) suite [Poste1981, Deeri1995], asynchronous transfer mode (ATM) [Dutto1995], synchronous digital hierarchy (SDH) [ITUT2000], and synchronous optical network (Sonet) [ANSI2001]. Combining technologies, however, results in complex and inefficient infrastructures that are difficult to manage and scale.

Recent developments in optical networking technologies and the evolution in the Internet technologies represent a window of opportunity to move from the complex, inefficient, multi-layer electronics networking model to a simpler, more efficient, two-layer optical networking model in which IP lies almost directly on top of the optical layer. Figure 1.1.1 shows the network-layering trend.

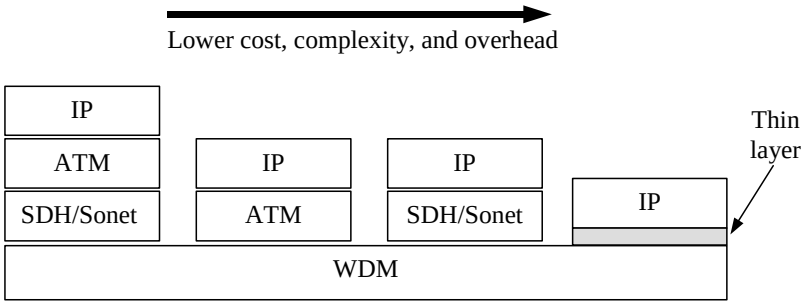


Figure 1.1.1 - Network-layering trend

Optical networking technologies allow for transmission, multiplexing, amplification, and switching in the optical domain, hence exploiting the full potential of optics and avoiding the so-called electronics bottleneck. Internet technologies, in turn, are evolving from the single, insecure, best-effort service to services with QoS and security (e.g., [Blake1998, Atkin1995]).

The potential of the optical Internet have attracted much attention of the research community and led to the design of many candidate architectures. Among them, ring networks using wavelength division multiplexing (WDM) [Ramas1998] and optical packet switching (OPS) are prominent, for they

combine the resilience and simplicity of the ring topology with the robustness of packet switching and the high capacity and transparency of optical networking technologies (see Section 1.1.1).

1.1.1 MOPS rings

A MOPS ring relies on WDM. WDM exploits the frequencies of the light in the low loss region of optical fibbers to create several Gb/s capacity wavelength channels. Independent from each other, these channels can be added gracefully as needed, and can be multiplexed in parallel to provide a total capacity of up to few tens of terabit per second (Tb/s) per optical fibre.

Figure 1.1.2 illustrates how WDM works. In the example, a WDM multiplexer sitting at the entrance of a fibre link multiplexes six 10Gb/s channels. These channels travel in parallel throughout the link and during the journey they are undistinguishable. At the exit of the link a WDM demultiplexer demultiplexes these channels, which become individually selectable again.

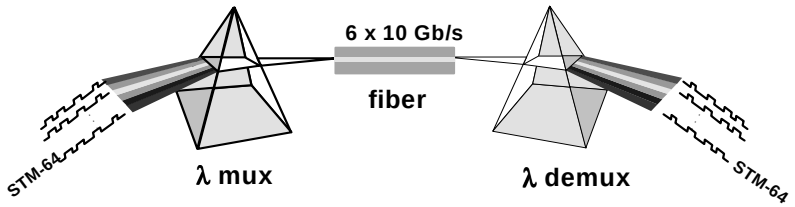


Figure 1.1.2 - Wavelength division multiplexing

A MOPS ring network comprises a number of optical fibre rings and a number of network nodes. Each optical fibre ring consists of a number of wavelength channels. To access these channels, network nodes encompass optical add-drop multiplexers (OADMs); in the literature the term wavelength add-drop multiplexer (WADM) is sometimes used to name such multiplexers.

An OADM accomplishes two basic operations:

- Add: the add operation redirects (that is, adds) an optical signal modulated by a transmitter (Tx) on a particular channel to the ring, on the same channel;
- Drop: the drop operation redirects (that is, drops) an optical signal on a particular channel from the ring to a local receiver (Rx) that operates on the same channel.

Figure 1.1.3 illustrates an OADM dropping channel 1 and adding channel 3.

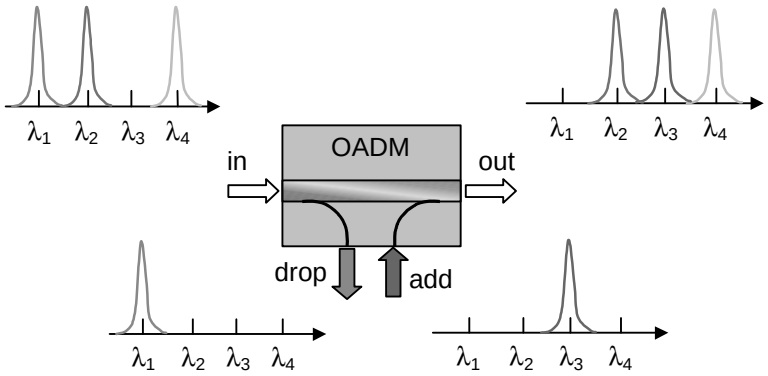


Figure 1.1.3 – Illustration of the functioning of an OADM node

Typically operations add and operations drop are independent. For either one there might be limitations on the number of instances that can be executed simultaneously, and on which channel, or channels, each instance can operate. For instance, an OADM may be capable of dropping a single optical signal at a time, whichever the channel that carries that signal is. Another OADM may be capable of dropping four optical signals simultaneously, each on a pre-determined channel.

Figure 1.1.4 illustrates a MOPS ring with four nodes and four wavelength channels. In this illustration, each node is assigned an integer index address corresponding to the position of that node in the ring and a unique reception channel whose index is equal to the index of that node. Thus, in this scenario each node can receive only on the channel that matches its own address; transmission can take place on any single channel at a time.

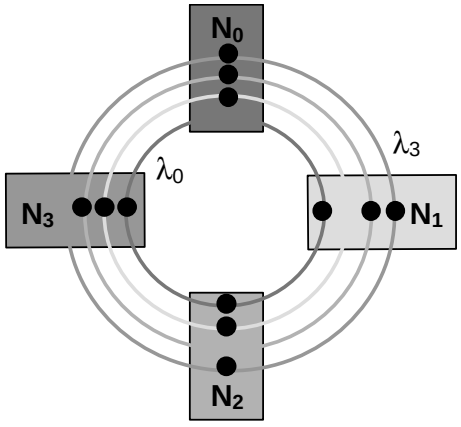


Figure 1.1.4 – Example of MOPS ring (one fibre; four wavelengths; four nodes)

OPS is the traditional packet switching model with the difference that in OPS a packet travels all the way from the source to the destination in the optical domain, bypassing all the intermediate nodes.

The benefits of node bypassing include:

- Intermediate nodes need not to support the particularities of transit communications, such as bit rate, signal modulation, framing;
- Lower layer management activities need to be performed only at ingress and egress network nodes, hence becoming simpler;
- End-to-end packet delays and end-to-end packet delay variations are more predictable and stable since packets do not experience queuing and access delays at intermediate nodes, as in the traditional store-and-forward approach. Predictable, stable packet delays are important because they affect the performance of certain protocols (e.g., the transport control protocol (TCP) [Poste1981]) and the quality of multimedia applications.

To achieve such benefits is very challenging though. In electronics networks, a node converts an incoming packet to the electronic domain, reads the contents of that packet, and buffers that packet while it decides upon either receiving, discarding, or forwarding. Nevertheless, because of limitations in the state-of-the-art of optical networking technologies, the content of a packet is not accessible in the optical domain. Furthermore, there is no optical memory technology available to delay packet by variable time. Therefore, the challenge to OPS is to find means to obtain the forwarding information about a packet, and to delay that packet at least until that information has been processed and the appropriate settings have been done.

MOPS ring architectures transport the forwarding information about a packet, which is usually encoded in the packet itself, separately. Every time a node transmits a packet, it also generates an additional header containing the forwarding information about that packet.

To obtain such information MOPS ring architectures tap off a small fraction of the signal power of the packets. To delay a packet until processing of its header completes and consequent actions are taken, MOPS ring architectures use fibre delay lines (FDLs). FDLs delay a packet for a fixed period of time that approximates the time taken to complete the forwarding decision making process and to react to the decision accordingly.

Figure 1.1.5 depicts a conceptual node architecture upon which almost all MOPS ring architectures rely.

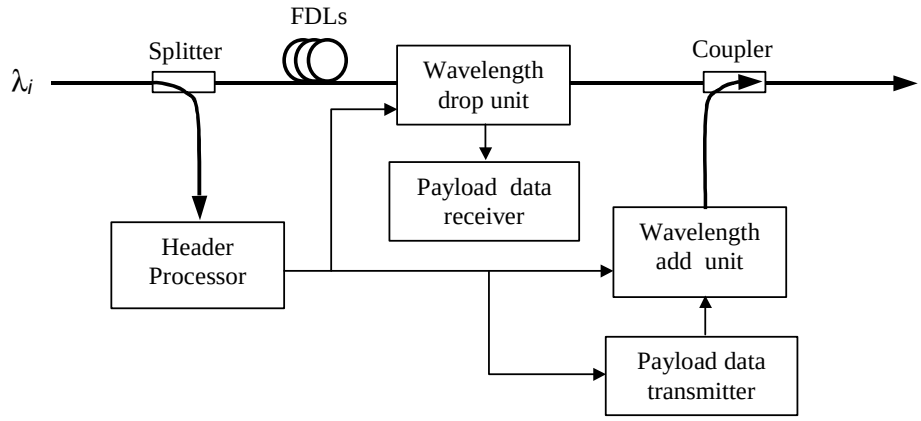


Figure 1.1.5 - Conceptual node architecture

Upon arrival of a packet, a node taps off a small fraction of the packet's signal power; the remaining signal power heads towards FDLs.

Almost all the existing node architectures convert the tapped off signal to the electronics domain before the header processor (HP) processes the corresponding header. Nevertheless, a few approaches exist that process headers optically [Murat2001, Hill2001].

In either case, the HP then processes the corresponding information, and reacts according to the decision made:

- Forward/Receive: the HP signals the wavelength drop unit to either drop the incoming payload, or let the incoming payload go. In either case, when the payload reaches the wavelength drop unit the latter has already been set-up accordingly;
- Transmit: decision made when the HP detects the channel idle; the HP signals the payload data transmitter to modulate the selected packet, and the wavelength add unit to add the modulated signal into the ring on the corresponding channel.

1.1.2 Medium access techniques and protocols

A ring is a shared medium for which geographically distributed nodes compete. Thus, a node can disturb the transmissions of other nodes. Moreover, because of ring symmetry, depending on the position in the ring some nodes may get better-than-average access opportunities while some others may get worse-than-average access opportunities.

Medium access control (MAC) regulates access to shared media to provide nodes distributed geographically with efficient, fair access opportunities. The MAC function belongs to the data link layer in the traditional network layering models. Figure 1.1.6 shows where the MAC function resides in the open systems interconnection (OSI) reference model (RM) [ISO1994] by the international standardization organisation (ISO), in the TCP/IP protocol stack [Carpe1996, Clark1988], and in the institute of electrical and electronics engineers (IEEE) 802 local area networks (LAN)&MAN RM [IEEE2001].

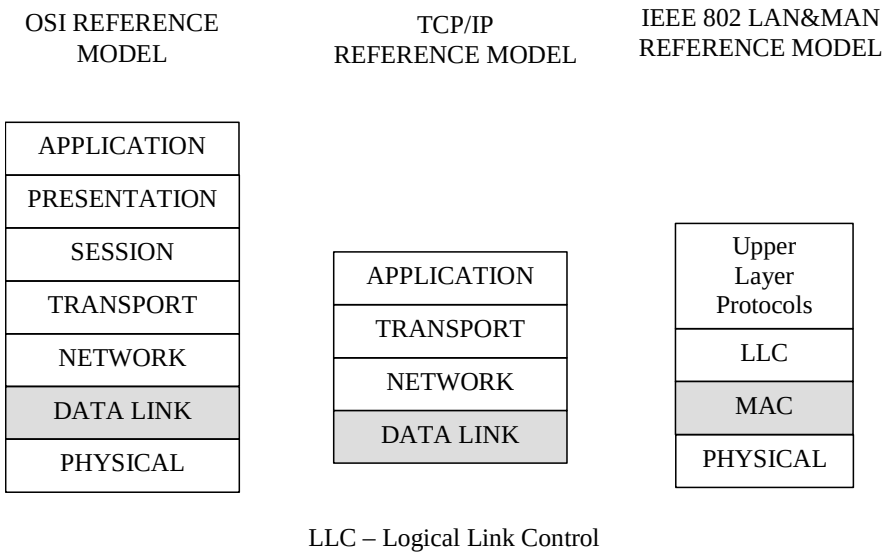


Figure 1.1.6 – Network-layering model

A MAC protocol implements the MAC function. It consists of a set of messages and a set of rules that determine, upon receiving those messages, in which situations access can occur, and in which situations access cannot occur. For instance, if the MAC client at the upper layer wants to transmit a packet then it issues a proper message, containing that packet, at the MAC interface. The physical (PHY) layer, upon detecting the medium idle, sets a specific variable at its interface to inform the MAC protocol that the medium is idle. If both conditions hold then the MAC protocol transmits the packet received from the MAC client.

There are many MAC protocols for packet switching ring networks. These protocols derive from the following media access techniques: token passing, time slotting, buffer insertion, and contention.

Figure 1.1.7 [vanAs1994a] classifies MAC protocols according to their media access technique; the numbers at the top of the figure indicate the year of publication. Note that the classification also includes circuit-switched rings and multiple rings, but such techniques are out of the scope of this dissertation.

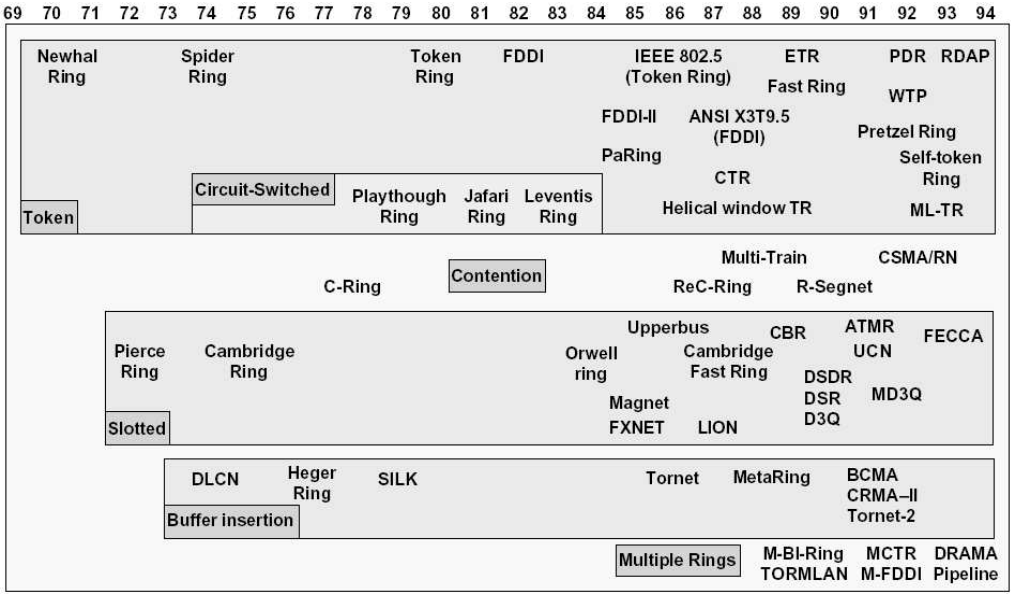


Figure 1.1.7 – Packet-switched ring MAC protocols [vanAs1994a]

Token passing, time slotting, and buffer insertion are conflict-free techniques, which means that once a node started transmitting that node can be sure that its transmission will not be disturbed by a transmission made by another node. The contention technique, as the name suggests, cannot provide such a guarantee. Thus, contention resolution schemes are necessary to cope with conflicts.

The token passing technique uses a special control packet called token to control access to the medium. The token rotates on the ring in the data direction, and each node holds the token for a pre-defined token holding time (THT), which may express actual period of time, number of packets, or number of data units; in general THT is sufficient to transmit one packet. Only the node that holds the token can transmit; all the others refrain from transmitting.

In the original version of token passing, a node forwards the token only upon arrival of the header of the packet transmitted previously. In subsequent versions, each node forwards the token immediately after completing the transmission of the last packet. The term early token release qualifies the latter.

Token ring [IEEE1998] is a classical example of MAC protocol that derives from the original version of token passing. Fibre distributed data interface (FDDI) [ANSI1988] is an example of MAC protocol that derives from the early token release version of token passing.

In the time slotting technique, known as slotted ring, the total capacity of the ring is divided into time slots of fixed length that rotate continuously on the ring. A single bit in each slot indicates whether the slot is empty or busy; transmission can occur upon arrival of an empty slot. To avoid contention, an incoming packet whose size is greater than the slot size is fragmented into smaller packets, and each is transmitted using a single slot.

The original version of slotted ring uses source removal, that is, a destination node copies the content of a slot addressed to it, but does not release the slot. Such a task is executed by the slot's source node. Upon releasing the slot, the source node forwards the empty slot to the next downstream node. Cambridge ring (CR) [Hopp1980] is an example of time-slotted ring network.

Subsequent versions of slotted ring use destination removal, that is, a destination node copies the content of a slot addressed to it, and also releases that slot. In certain versions the node that releases a slot forwards that slot to the next downstream node. Orwell [Falco1985] is an example of this version of time-slotted ring network. In some other versions, the node that releases a slot can reuse that same slot immediately. The slotted mode MetaRing [Ofek1994] is a classical example of this version of time-slotted ring network.

The buffer insertion technique uses a transit buffer to avoid contention. In this technique, transmission is possible whenever the medium is idle and the transit buffer is empty. If a node is transmitting a packet, and it detects the arrival of a transit packet on the ring, then that node deviates the packet into the transit buffer. Only after its own packet has been completely transmitted that node will re-insert the transit packet into the network.

The original version of the buffer insertion technique uses source removal; distributed loop computer network (DLCN) [Reame1977] is an example of buffer insertion with source removal. Subsequent versions of buffer insertion use destination removal; the asynchronous MetaRing, and cyclic reservation multiple access II (CRMA-II) [vanAs1994b] are examples of buffer insertion with destination removal.

In the contention technique, nodes can transmit whenever they sense the medium idle. An example of contention resolution is that proposed in carrier sensed multiple access ring network (CSMA/RN) [Foudr1991]. In CSMA/RN a node that is transmitting and detects a contention stops the transmission and signals the destination node that the transmission has been cancelled; the source node continues the transmission of the packet when the ring becomes idle again.

Destination removal MAC protocols can achieve high throughputs, whereas the maximal achievable throughput depends heavily on the traffic distribution [Marsa1993]. For instance, assuming symmetric traffic distribution, and a single ring, information travels $N / 2$ hops in average, where N denotes the number of nodes in the network. Therefore, the throughput is twice the nominal capacity of the ring.

The extra capacity obtained with destination removal is very difficult to control, and, if not taken care properly, introduces access unfairness in the network. That is, given a particular class that identifies nodes with the same demand characteristics, some of the nodes belonging to that class get better-than-average access opportunities, while some other nodes belonging to that class get worse-than-average access opportunities.

Destination removal MAC protocols need an additional access fairness protocol to ensure that the controlled network is fair. Access fairness protocols are either global, or local. Global protocols consider the network as a single

shared communication resource. Consequently, every node sees the same transmission constraint. Local protocols consider each link as a communication resource and the whole network as a multiplicity of communication resources. Therefore, only nodes competing for the same subset of communication resources see the same transmission constraint.

1.2 Motivations of this work

According to [Claff1998, Thomp1997], Internet packets are variable in size and mostly small, with peaks at 40, 44, 552, 576, 1500 bytes (B). Whilst approximately 60% of the packets are equal to or smaller than 44B, approximately 50% of the traffic is carried in 1500B packets. For such traffic characteristics, asynchronous MAC protocols are preferred to synchronous MAC protocols.

Nevertheless, buffer insertion protocols cannot be implemented optically because of the lack of variable delay optical buffering technologies –variable delay optical buffering technologies are at early stages of development, and it will take a few couple of years until concepts are proven.

Contention protocols can be implemented optically, but because of collisions they may achieve poor performances under certain traffic scenarios, in particular for an increasing number of nodes -the IEEE 802.3 Ethernet [IEEE2002] MAC protocol suffers from the same problem too, but Ethernet is meant for LANs, and in such networks it is more cost effective to over provision rather than to use more efficient but complex protocols.

Token passing protocols, which are synchronous, suffer from poor performances in networks with high delay $\times$ bandwidth product, such as MOPS rings. As the medium length or the medium bandwidth increases, so does the time it takes for the token to return to a node that just released it, and so does the waste of capacity available.

There is only one MOPS ring MAC protocol [Cai2000] that uses token passing, but to achieve an aggregate throughput of approximately 1, the number of network channels must be high, and the node architecture must be capable of transmitting and receiving on all the channels simultaneously.

The only class of protocols that can achieve high performances in the optical domain is slotted ring with destination removal. Therefore, almost all the existing MOPS ring MAC protocols are slotted ring with destination removal.

An issue when using the slotted ring technique is the mismatch between variable size packets, such as Internet's, and the fixed size slots. Almost all the existing slotted ring with destination removal MAC protocols designed for MOPS rings assume fixed-size packets to match the size of the slots exactly. Although Internet packets have variable sizes, the arbitrary use of segmentation and re-assembly (SAR) operations supports the assumption adopted by those protocols.

Figure 1.2.1 illustrates SAR. The convergence layer of the source node segments a packet coming from the upper layer into fragments that are smaller than the slot size, or equal to it, independent of the status of the medium; the

MAC layer transmits such fragments unaware of their correlation. The convergence layer of the destination node buffers incoming fragments. When the complete packet has arrived, the convergence layer re-assembles the fragments and sends the resulting packet to the upper layer.

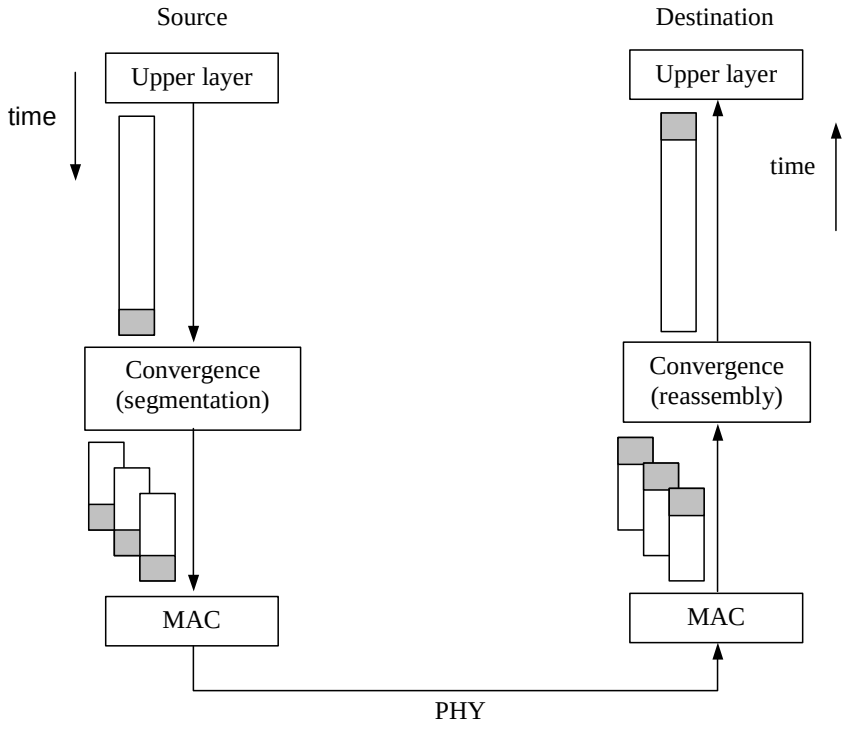


Figure 1.2.1 – Illustration of SAR operations

Despite their simplicity, SAR operations present a few drawbacks:

- Replication of the protocol control information (PCI) contained in each original packet into each of the generated fragments, thus increasing the network overhead;
- Multiplication of the processing overhead necessary to deal with each packet by the number of generated fragments corresponding to that packet;
- Destination nodes have to store fragments until they can be re-assembled and sent to the upper layer, and that implies silica since SAR operations are usually implemented in hardware. Given the number of possible simultaneous communication sessions between a given destination node and any other node, considerable memory has to be available, even if an average number of sessions rather than maximum is assumed to calculate memory requirements. Anyway, a destination node may run out of memory and, consequently, discard fragments. Such an action may impact the corresponding application or even the network overall performance;
- Depending on communication patterns and the access mechanism, it may happen that many re-assembly operations take place at the same time at a

given node. The simultaneous execution of many re-assembly operations may result in a burst of packets being sent to the upper layer. If the upper layer cannot process all the packets simultaneously then it applies packet discard, whose effects have already been mentioned.

Internet traffic is known for being asymmetric, very dynamic, and, for this reason, difficult to predict. Nevertheless, all the existing slotted ring with destination removal MAC protocols use static global fairness mechanisms. Such mechanisms depend on the previous knowledge of the traffic conditions -which is often difficult to obtain- and do not adapt to changes in traffic conditions. Furthermore, they might prevent nodes from transmitting even if the transmissions would not traverse any bottleneck link.

Figure 1.2.2 [vanAs2001] illustrates the performance difference between global and local fairness protocols. The scenario consists of an eight-node single, unidirectional ring with destination removal. As a result of the traffic pattern (shown by the dashed arrows), the ring can be divided into two partitions; consequently, its actual capacity is twice its nominal capacity, that is, 2, or 1 per partition.

In the first partition node 0 demands 100% of the ring capacity to transmit to node 3 and node 1 demands 100% of the ring capacity to transmit to node 2; such a demand of 2 exceeds the total capacity of that partition, thus generating a bottleneck -the term bottleneck qualifies a link whose total demand exceeds the actual capacity of that link. In the second partition node 4 demands 30% of the ring capacity to transmit to node 7 and node 5 demands 70% of the ring capacity to transmit to node 6.

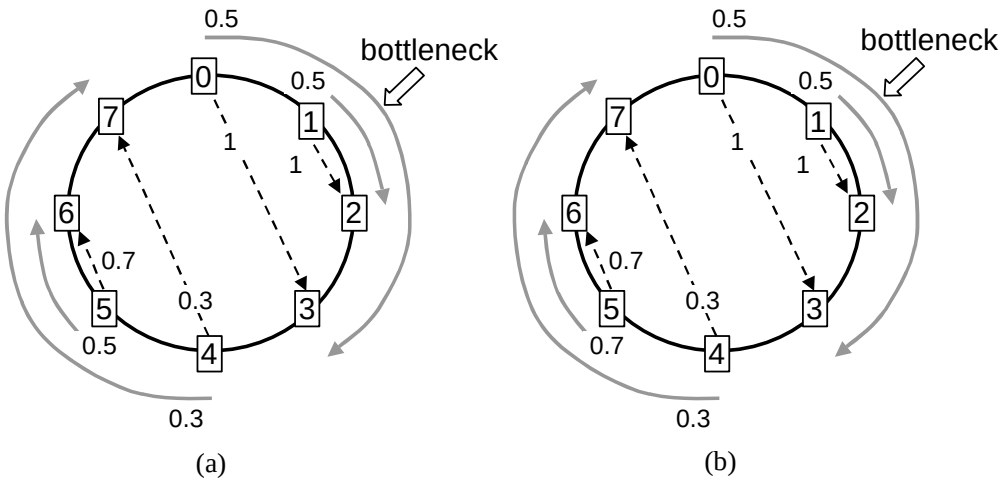


Figure 1.2.2 - Example of (a) global fairness; example of (b) local fairness

As it can be seen in Figure 1.2.2a and Figure 1.2.2b, both protocols assign the same capacity to the nodes over the bottleneck link, that is, 50% of the ring capacity to node 0 and 50% of the ring capacity node 1. Nevertheless, because in

global protocols there is a single transmission constraint, node 5 receives only 50% of the ring capacity (Figure 1.2.2a), the maximum capacity assigned to the nodes over the bottleneck link, even though it could have been assigned the remaining capacity (also known as residual capacity) of the link, which is 70%, as it actually happens under the local protocol (see Figure 1.2.2b).

The simplicity and high performance of static global fairness protocols under symmetric, rather constant traffic conditions, and the complexity of the existing local fairness protocols explain why almost all the MOPS ring MAC protocols use a static global fairness protocol.

1.3 Objectives

This dissertation is concerned with MAC protocols that can transport Internet traffic efficiently and with high performance over MOPS rings. Because of the characteristics of the existing media access techniques, the characteristics of actual MOPS rings, and the characteristics of the Internet traffic, the work contained in this dissertation focuses on combinations of the destination removal slotted ring media access technique with additional mechanisms.

Given the separation between access control and access fairness in the slotted ring with destination removal technique, the major objectives of this work are as follows:

- To investigate and propose access control protocols to allow for the transport of variable size packets efficiently;
- To investigate and propose access fairness protocols that can ensure fair access to the medium under any traffic condition and with minimum impact on the network performance.

Four new access control protocols are described, two of them conflict-free, and the other two protocols are prone to contention. The four protocols are meant for MOPS rings that follow the conceptual node architecture shown in Figure 1.1.5, and they are capable of working with heterogeneous transceiver configurations.

Two new dynamic access fairness protocols are proposed, one global and one local. Although designed for MOPS ring networks, they can be used in packet switching ring networks in general, be they synchronous or asynchronous, all optical or electronics, single-hop or multi-hop.

To verify whether the proposed protocols meet their objectives, this work also aims to answer the following questions:

- How the two proposed access fairness protocols compare with each other and with the static global access fairness protocol used in almost all MOPS ring MAC protocols;
- How each access control protocol behaves when combined with each access fairness protocol;
- How each access fairness protocol affects the performance of each access control protocol.

Note that this dissertation covers the most basic MAC service –access, which does not mean that other services are superfluous or should only be provided by the upper layers. Services like protection and QoS are important as well, but they have been left out of this dissertation deliberately to limit the scope of the dissertation and keep the focus on the basic MAC service.

1.4 Organization

The next chapters are organized as follows:

- Chapter 2 describes a few existing MOPS ring architectures and discusses some of the characteristics of these architectures;
- Chapter 3 describes a few existing access control protocols for MOPS rings and analyses their qualities and limitations;
- Chapter 4 introduces four access control protocols and analyses their qualities and drawbacks;
- Chapter 5 introduces two access fairness protocols, whereas one is global and one is local, analyses the qualities and limitations of each protocol, and discusses the integration of such protocols with the protocols introduced in Chapter 4. Chapter 5 also describes and analyses a few existing access fairness protocols;
- Chapter 6 evaluates comparatively the performance of the protocols introduced in Chapters 4 and 5;
- Chapter 7 presents some final conclusions about the obtained results and identifies possible directions for future research.

Chapter 2
MOPS ring architectures

This chapter describes a few MOPS ring network architectures that have been proposed in the literature or built in laboratories, and analyses their advantages and disadvantages.

All the described architectures follow the slotted ring access technique and, consequently, rely on node architectures that are similar to the conceptual node architecture described previously.

The node architecture of each of the described network architectures assumes homogeneous transceiver tunability. To describe such node architectures this work uses the notation $FTx^{i}TTx^{j}$-$FRx^{m}TRx^{n}$ proposed in [Mukhe1992], where *i* describes the number of fixed-tuned Tx (FTx) elements, *j* describes the number of tuneable Tx (TTx) elements, *m* describes the number of fixed-tuned Rx (FRx) elements, and *n* describes the number of tuneable Rx (TRx) elements.

2.1 Pipeline

One of the precursor MOPS rings, Pipeline [Chlam1995] is a single-fibre, *C*-channel, *N*-node network, where $N = C$. To achieve OPS Pipeline uses the slotted ring media access technique and synchronizes slots in parallel across the channels.

To transport slot header information, Pipeline uses a technique called multiple subcarrier signalling (MSS) that uses an electrical subcarrier to encode slot headers at low bit-rate.

Let also n_d be the destination node, and sc_d be the unique subcarrier frequency pre-assigned to n_d. To send a packet to n_d, MSS encodes the corresponding header information using sc_d.

A slot header subcarrier signal is combined with the corresponding packet baseband signal, and the resulting signal is used to modulate the Tx laser; both payload and header have the same duration: a time slot.

Destination nodes perform clock recovery on a packet-by-packet basis. To make that possible, together with a packet a node always send a tone at the frequency $1/T$, where T is the bit duration. The destination uses this filtered tone as a sampling clock.

Figure 2.1.1 illustrates how MSS works. In the example, the header detector of a node does not sense sc_1, and informs the control logic that sc_1 is missing. The control logic then triggers the packet transmission procedure, which culminates with a packet and its corresponding header being inserted into the ring accordingly.

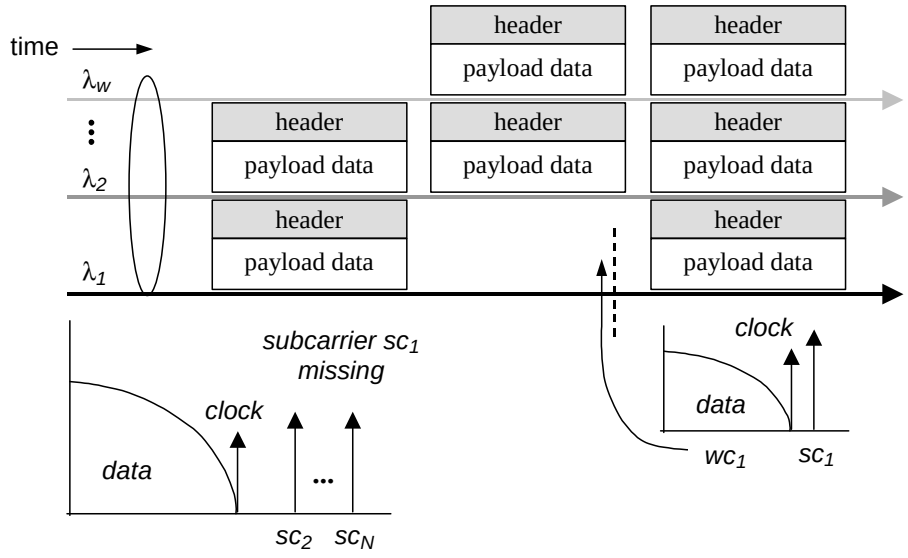


Figure 2.1.1 – Illustration of access under MSS

Pipeline defines two architectures: TR-Pipeline and TT-Pipeline. Figure 2.1.2 depicts the general schematics of a Pipeline node. The schematic diagram represents both architectures, but the functionality and technology of each module may vary depending on the architecture.

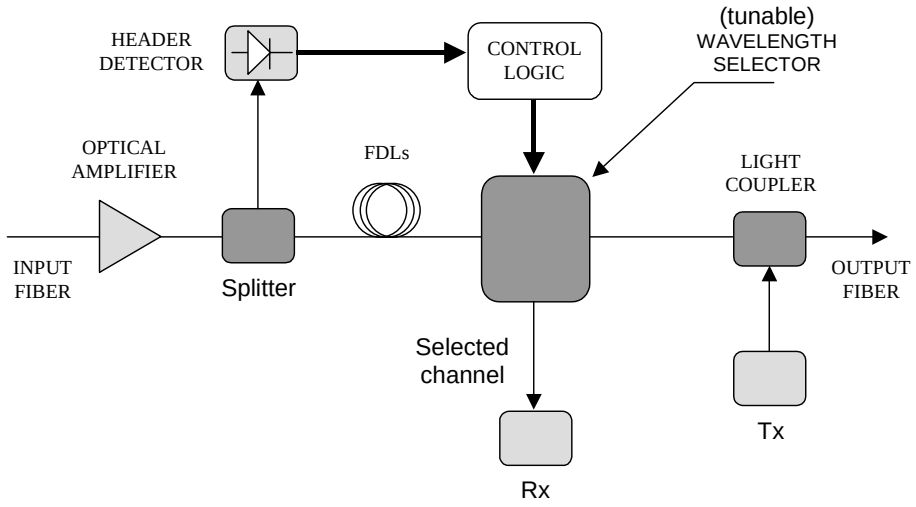


Figure 2.1.2 - Schematic of the Pipeline node

TR-Pipeline

TR-Pipeline is an FTx^1-TRx^1 configuration. The Tx unit is a laser fixed-tuned to a unique transmission channel, and it could be directly modulated. The wavelength selector is tuneable, and it could be implemented using various technologies, such

as acousto-optic tuneable filters (AOTFs). To overcome the long tuning latency of these filters, which is in the order of microseconds, two filters could be used in tandem, whereby one receives while the other is tuning.

MSS adapts to TR-Pipeline as follows. Let n_s be the source node, and wc_s be the unique transmission channel pre-assigned to n_s. Let also n_d be the destination node, and sc_d be the unique subcarrier frequency pre-assigned to n_d. To send a packet to n_d, n_s transmits on wc_s, and encodes the header information using sc_d.

TR-Pipeline works as follows. At each node, C parallel slots, each on a particular channel, undergo optical amplification; for amplification either erbium-doped fibre amplifiers (EDFAs) or semiconductor optical amplifiers (SOAs) can be used.

As the slots arrive at the splitter, the latter taps off a small fraction of their power, and redirects the tapped off signal to the header detector (HD), a photodiode; the remaining signal heads to the FDLs.

In the output current of the photodetector the present data payloads jam at baseband, leaving the subcarriers intact for detection.

The HD informs the control logic unit (CLU) of all the detected subcarrier frequencies. From such information the CLU can find out if there is any slot destined to the node, and which slots are empty, if any. To find out if there is a slot to receive, the CLU matches each detected frequency with its own pre-assigned subcarrier frequency; if a match is found then there is a slot to receive. To find out which slots are empty, the CLU matches each detected subcarrier frequency's associated channel with the list of all possible channels; the slots on the channels that fail to match are empty.

If there is a slot to receive then the CLU signals the wavelength selector unit to tune to the slot's channel, and drop the slot, hence redirecting the packet to the Rx unit and releasing the slot.

At this point the empty slot contention/collision avoidance (ESCA) protocol takes over to decide whether transmission can or cannot occur.

If packet transmission can occur then the CLU:

- signals MSS to generate the corresponding header on the appropriate subcarrier frequency; and
- signals the Tx unit to combine the generated header with the selected packet, and insert the combined signal into the empty slot.

TT-Pipeline

TT-Pipeline is a TTx^1-FRx^1 configuration, and it is very similar to TR-Pipeline. TT-Pipeline uses a fast-tuneable laser Tx unit, such as distributed Bragg reflector (DBR) laser, and a fixed-tuned wavelength selector. The wavelength selector could be based on technologies such as AOTFs, Fabry-Perot filters (FPFs), glass components.

MSS adapts to TT-Pipeline as follows. Let n_s be the source node. Let also n_d be the destination node, wc_d be the reception channel pre-assigned to n_d, and sc_d be the unique subcarrier frequency pre-assigned to n_d. To send a packet to n_d, n_s transmits that packet on wc_d, and encodes the header information using sc_d.

TT-Pipeline works as follows. At each node, W parallel slots, each on a particular channel, undergo optical amplification. As they arrive at the splitter, the latter taps off a small fraction of their power, and redirects the tapped off signal to the HD; the remaining signal heads to the FDLs.

In the output current of the photodetector the present data payloads jam at baseband, leaving the subcarriers intact for detection.

The HD informs the CLU of all the detected subcarrier frequencies. From such information the CLU can find out if there is any slot destined to the node, and which slots are empty, if any. To find out if there is a slot to receive, the CLU matches each detected frequency with its own pre-assigned subcarrier frequency. If a match is found then the CLU matches the corresponding slot's channel with its own pre-assigned reception channel. If there is a match then there is a slot to receive. To find out which slots are empty, the CLU matches each detected subcarrier frequency's associated channel with the list of all possible channels; the slots on the channels that fail the match are empty.

If there is a slot to receive then the CLU signals the wavelength selector unit to drop the slot's channel, hence redirecting the packet to the Rx unit and releasing the slot.

From this point on the ESCA protocol takes over to decide whether transmission can or cannot occur.

If packet transmission can occur then the CLU:

- signals MSS to generate the corresponding header on the appropriate subcarrier frequency; and
- signals the Tx unit to combine the generated header with the selected packet, tune to the appropriate channel, and insert the combined signal into the empty slot.

2.2 BORN

Kang et al. [Kang1995] describes a MOPS ring network that follows an FTx^{C}-FRx^{C} configuration, and adopts the slotted ring media access technique. This dissertation refers to such a network as broadband optical ring network (BORN).

Nodes attach to the ring via the network interface unit (NIU). The NIU consists of *C* OPS devices, and *C* transceivers that share the same transmit buffer (TB) and the same receive buffer (RB), where *C* denotes the number of channels in the network.

Figure 2.2.1 depicts the transceiver and the buffer structures. Figure 2.2.2 depicts the NIU organization.

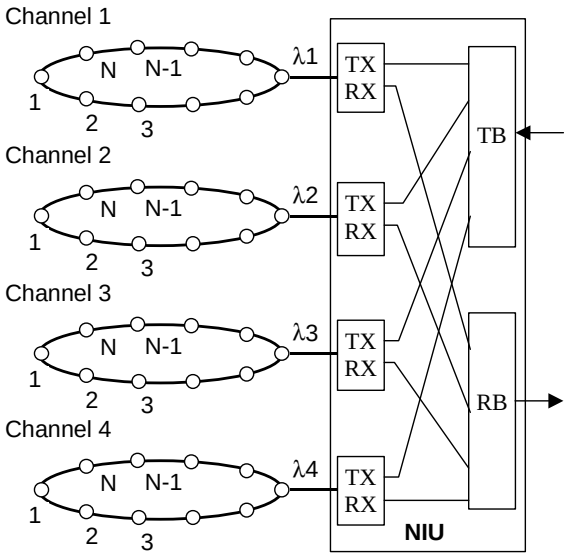


Figure 2.2.1 - Shared buffer and transceivers

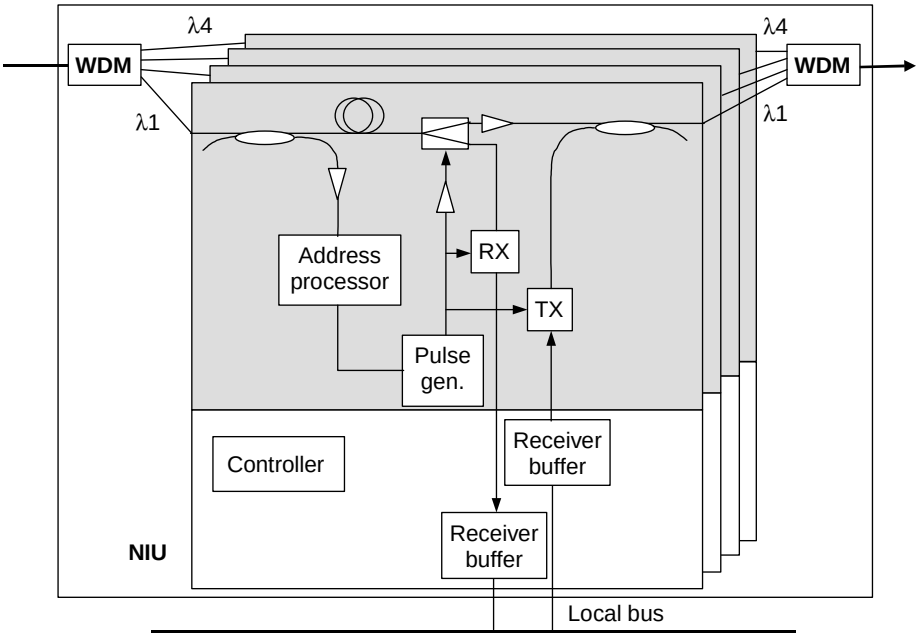


Figure 2.2.2 – NIU organization

The NIU also includes one WDM demultiplexer and one WDM multiplexer.

Each OPS device operates on a particular channel. It is the task of the WDM demultiplexer to separate incoming signals and forward each to the corresponding OPS device.

At each OPS device, a coupler taps off a small fraction of the signal power to obtain the header of a slot; in BORN the header consists of the destination

address and the status of the slot, and it is (assumed to be) transmitted just ahead of its payload. Figure 2.2.3 illustrates the signalling technique.

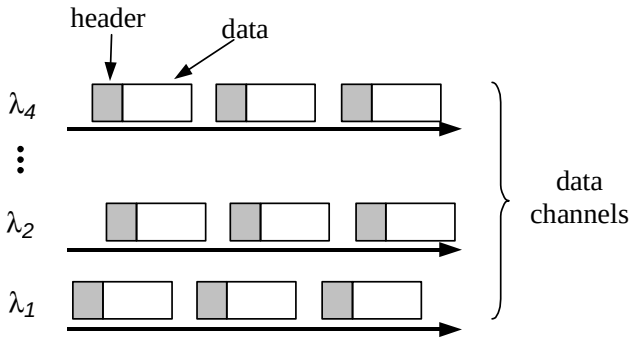


Figure 2.2.3 - In-band serial signalling

The tapped off signal undergoes optical amplification and then feeds the address processor unit; the remaining signal power heads to FDLs, which holds it up for a fixed period of time.

The address processor consists of an FDL-matched filter and a threshold detector. The address of the corresponding node is pre-assigned to the filter, which correlates the assigned address with every input address signal to generate a proper correlation pulse. Since the centre peak value of autocorrelation pulses is always higher than that of cross-correlation ones, the filter can determine whether an input address signal matches the node address by thresholding the correlation outputs.

Figure 2.2.4 illustrates how the optical address recognition mechanism works. For the input address signal 1101, an autocorrelation occurs with value 3 because the signal matches the node address. Nevertheless, for the input address signal 1011, a cross-correlation occurs with peak value 2 because of the mismatch between the input signal and the node address.

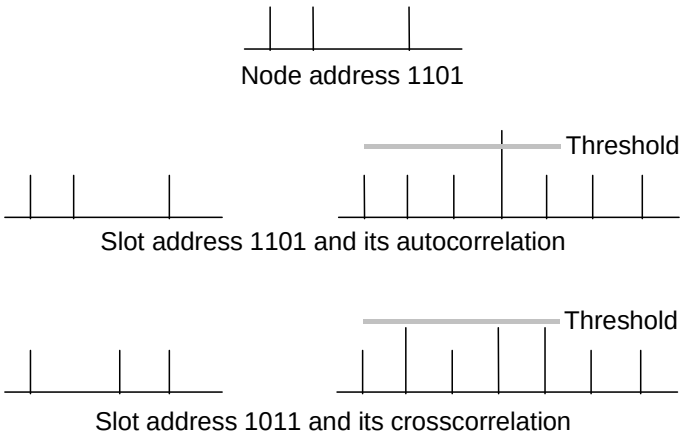


Figure 2.2.4 - Optical address recognition

For information on address generation and recognition refer to [Kang1995].

The address processor decides an address matching, and produces a trigger pulse that the gate pulse generator uses to control the 1×2 optical switch, the Tx unit, and the Rx unit. If the address processor triggers a pulse to indicate an address match, then the gate pulse generator issues a pulse of value 1. Such a pulse tells the switch to drop the packet to the Rx unit, and the Tx unit to modulate a packet ready for transmission. If the address processor triggers a pulse to indicate an address match failure, then the gate pulse generator issues a pulse of value 0. Upon reception of a pulse of value 0 the switch lets the packet go to.

Eventually the WDM multiplexer multiplexes the signals leaving the OPS devices.

2.3 MAWSON

Metropolitan area wavelength switched optical network (MAWSON) [Summe1997] is a passive MOPS slotted ring. MAWSON strives for simplicity and low cost, and it only uses components commercially available.

MAWSON consists of a single optical fibre, *N* OADM nodes and *C* wavelength channels, whereas $N = C$, and it follows an FTx^{C}-FRx^{1} design. Each node is equipped with a multiple-wavelength laser array transmitter that is capable of not only switching very rapidly between channels, but also transmitting common data on a number of channels simultaneously.

Each node is assigned a fixed-tuned reception channel and is addressable by that channel. Therefore, to transmit to a destination node *d*, a source node has to tune to channel *d* first.

Figure 2.3.1 depicts a four-node, four-wavelength MAWSON network.

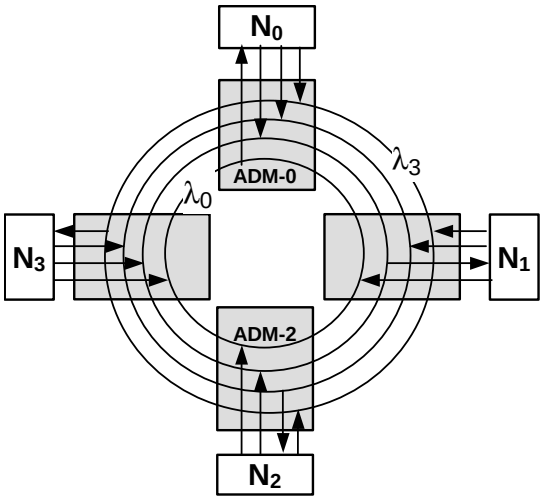


Figure 2.3.1 - Four-node, four-wavelength MAWSON network

The OADM design, shown in Figure 2.3.2, relies on the integration of an optical circulator, a fibre Bragg grating (FBG), and a fibre coupler.

The OADM works as follows. All input wavelengths pass from port 1 to port 2 of the circulator. Nevertheless, the FBG reflects all light within a fixed bandwidth around wavelength λ_i back to port 2 of the circulator. Eventually, the signal on wavelength λ_i emerges at port 3, the drop port of the OADM; the other signals pass through.

Signals at any wavelength(s) may be added to the ring via the coupler.

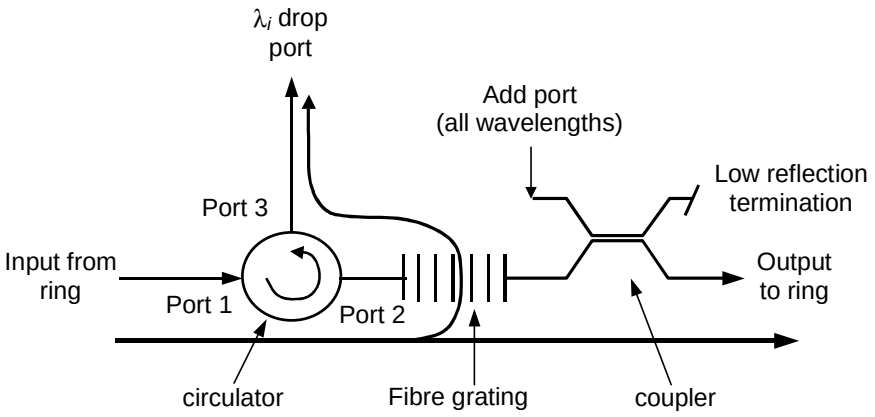


Figure 2.3.2 - Passive OADM design

MAWSON nodes cannot sense the medium to find out whether transmission can or cannot occur. To find out about the status of the medium is the task of the MAC protocol developed for MAWSON.

2.4 HORNET

Hybrid opto-electronic ring network (HORNET) [Shrik2000a] is a MOPS ring meant for MANs. HORNET is designed to scale to 100 access point (AP) nodes and a ring length of approximately 100 km. In the current specification, the bit rate of each channel is 2.5Gbps.

HORNET follows a TTx^1-FRx^1 design, and it has two modes of operation: single-hop and multihop. If the number *N* of AP nodes is smaller than the number *C* of channels, or equal, then each node is assigned an exclusive, dynamically unchangeable reception channel. In this condition, HORNET works in the single-hop mode. If *N* is greater than *C* then two or more AP nodes may share the same dynamically unchangeable reception channel. In this condition, HORNET works in the multihop mode.

In either case, HORNET uses subcarrier signalling to carry packets destination addresses, a mechanism similar to Pipeline's. Each channel is associated with a unique subcarrier frequency. Whenever an AP wants to transmit a packet on a certain channel, that AP multiplexes the subcarrier frequency corresponding to that channel.

An AP node consists of the following three functional blocks: slot manager, smart drop, smart add. Figure 2.4.1 depicts the structure of an AP node.

The slot manager monitors the subcarriers, and to do so the slot manager taps off approximately 10% of the incoming signal power. The slot manager performs two parallel tasks:

- To inform the smart drop of the existence of a packet to drop: the address recovery module informs the packet switch at the smart drop to switch the packet to either the local network, if the node is the destination, or the multihop retransmission queue, if the node is not the destination;
- To inform the smart add of the existence of idle channels and about the idleness length of each idle channel: the slot detector module informs the fast-tuneable packet transmitter at the smart add about the idle channels and about the idleness length of each idle channels, so that the latter can decide whether to transmit and on which channel.

Meanwhile FDLs in the slot manager delay the signal so that when the signal leaves the slot manager the packet switch in the smart drop is set-up properly and the transmission decision in the smart add, if applicable, is taken.

The smart drop uses a circulator with a FBG to drop any signal on the pre-assigned channel. The burst-mode packet receiver [Shrik2001] detects a dropped signal and recovers the bit clock of the signal to synchronize with the packet that the signal modulates. The packet switch then forwards the packet according to its destination address, as explained previously.

The smart add processes the information received by the slot manager and, based on the processing of such information, selects a packet for transmission and signals the fast-tuneable packet transmitter [Shrik2001], a grating coupler sampled reflector (GCSR) tuneable laser, to modulate the selected packet on the selected channel. At the exit of the smart add the coupler adds the generated signal to the others, if any, already on the ring.

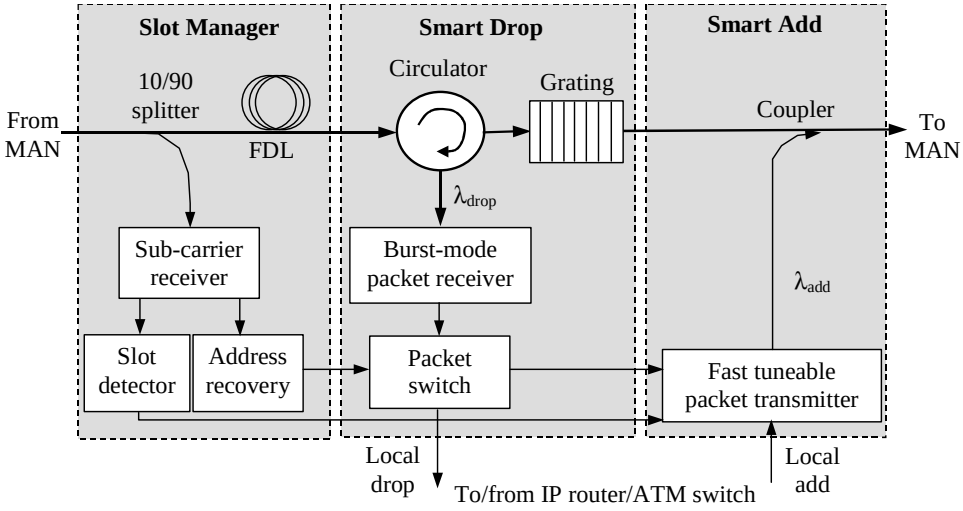


Figure 2.4.1 - Structure of AP node

2.5 RINGO

An alternative to Sonet/SDH rings, ring optical network (RINGO) [Caren2002] is a MOPS rings for MANs and wide area networks (WANs).

RINGO consists of a unidirectional fibre ring, N nodes implementing the interface between the electronic domain and the optical domain, and $C = N$ wavelength channels. Each channel is divided into time slots of fixed-length, and the channels are slot-synchronized in parallel.

RINGO nodes follow an FTx^{C}-FRx^{1} configuration. Each node is equipped with a laser array transmitter and a single FRx operating on a unique channel through which that node is addressable. A node can transmit on any channel except its own reception channel; a node never transmits to itself. To transmit to a destination node *d*, a source node *s* switches its laser to channel *d* first.

An incoming aggregate signal at a RINGO node passes through a gain-locked EDFA to compensate for the loss of the node passive elements as well as the downstream fibre link. An arrayed-waveguide (AWG) demultiplexes the aggregate signal and drops the reception channel. The signals on the channels other than the drop channel pass through a coupler that taps off approximately 10% of the signal power. The second AWG multiplexes back the remaining signal power on each channel, and FDLs delay the resulting aggregate signal while the dropped signal and the tapped off signals are treated.

If the burst-mode receiver detects an incoming signal then it synchronizes with the packet and receives it.

The direct current (DC)-coupled photodiode array detects the tapped off signals, and the threshold comparator informs the node controller of the status of the channels. The threshold comparator matches each signal power with the pre-defined threshold. If the power is above the threshold then the channel is busy; if the power is below the threshold then the channel is idle.

If there is any channel idle then the node controller turns on the appropriate laser(s) by DC-injection modulation. The node controller also signals the data source to transmit the (selected) packet(s). The external modulator receives the packet(s) from the data source and writes the packet(s) into the optical carrier.

At the exit of the node a coupler joins the modulated packet with the outgoing aggregate signal.

Figure 2.5.1 depicts the architecture of a RINGO node.

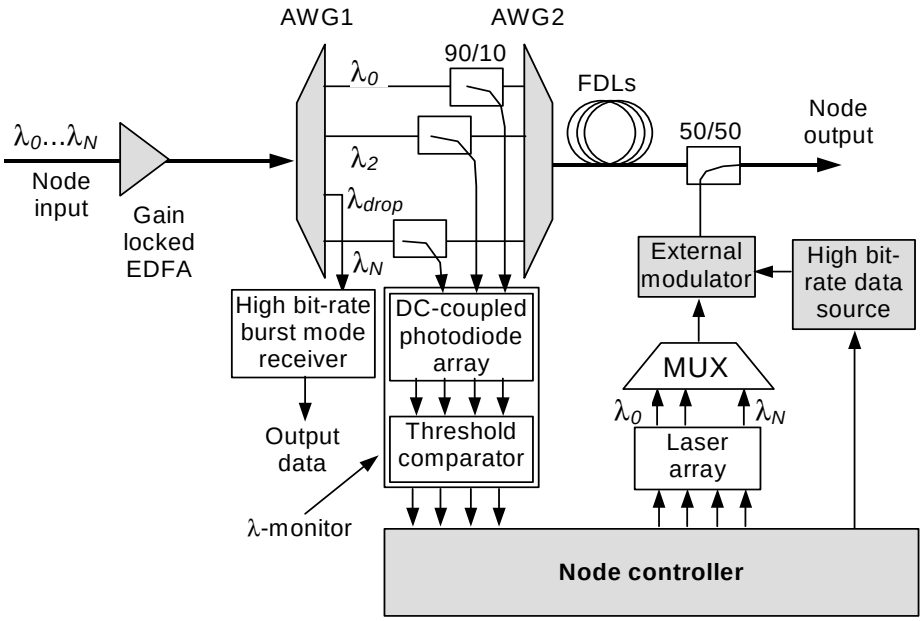


Figure 2.5.1 - Architecture of the RINGO node

2.6 FLAMINGO

Flamingo [Dey2000, Dey2001a, Dey2001b] is a MOPS slotted ring network designed to support unicast as well as multicast communications in the optical domain.

Flamingo in principle follows an FTx^{1}-FRx^{C} configuration, where *C* denotes the number of data channels, but it can also follow an FTx^{C}-FRx^{C} configuration with minimum impact on the node architecture. The total capacity of each wavelength channel is divided into slots of fixed length, and the channels are slot-synchronized in parallel so as to reach each node all at the same time.

One wavelength channel is used exclusively for the transmission of control information. Slots on the control channel, herein called control slots, are sent slightly ahead of their corresponding slots on the payload data channels, herein referred to as payload slots; this is to account for the configuration time of the fabrics at the nodes and, consequently, shorten the length of FDLs that are required to delay payload slots while their corresponding control slot is being electronically processed.

Figure 2.6.1 illustrates how payload slots and control slots are synchronized.

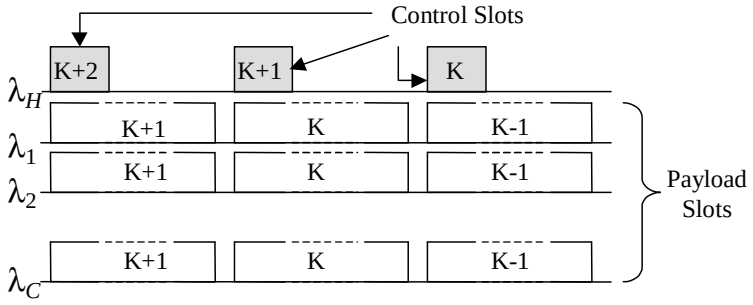


Figure 2.6.1 – Synchronisation among payload slots and between control slots and payload slots

A control slot contains *C* slot headers. Each slot header carries control information of a given payload slot. Figure 2.6.2 illustrates the relation between slot headers and their corresponding payload slots.

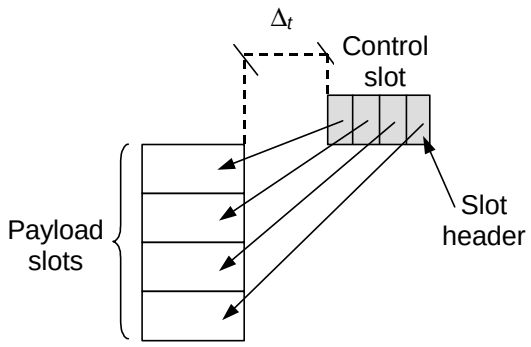


Figure 2.6.2 – Correspondence between slot headers and payload slots

The network consists of *N* OADM nodes. If nodes follow an FTx[1]-FRx[C] configuration then each Tx unit is pre-assigned a transmission channel. Which channel is assigned to each node's Tx unit depends on *N* and *C*. If $N \leq C$ then each node is assigned an exclusive transmission channel. If $N > C$ then N / C nodes[1] share the same transmission channel.

Each node is also equipped with one FTx unit and one FRx unit for the transmission of control information. The FTx unit and the FRx unit of all the nodes are tuned to the same control channel.

At the entrance of the node a slowly tuneable λ-drop separates the wavelength channel carrying the control slot from the wavelength channels carrying the payload slots. The control slot is converted to the electronic domain and processed by the header processor unit (HPU). To account for the time to process a control slot, payload slots are delayed in FDLs.

The payload slots then leave the FDLs and pass through an EDFA before a coupler taps off ten percent of their signal power. Both the tapped off and the

[1] For the sake of simplicity assume *N* to be an integer multiple of *C*.

remaining aggregate signals head to the demultiplexers for separation. The demultiplexer that receives the tapped off aggregate signal demultiplexes the latter and directs each signal to the appropriate Rx unit. The demultiplexer that receives the remaining aggregate signal power demultiplexes the latter and directs each signal to the appropriate 1×2 switch.

The HPU reacts to the information contained in the control slot by setting each of the switches to either drop, or forward the incoming slot, and signalling each of the Rx units to either accept, or discard the incoming slot.

Given a certain slot on a particular channel, if the node is the slot's destination then the HPU signals the corresponding Rx unit to accept the incoming slot and the corresponding switch to drop the slot. If the node is the slot's source then the HPU signals the corresponding Rx unit to discard the incoming slot and the corresponding switch to drop the slot. If the node is neither the slot's destination nor the slot's source then the HPU signals the corresponding Rx unit to discard the incoming slot and the switch to let the slot go. If the slot is empty then the HPU signals the corresponding Tx unit, or the Tx unit if there is only one, to transmit and the corresponding switch to add the signal modulated by that Tx unit.

A multiplexer multiplexes all the signals back into an aggregated signal, and a slowly tuneable λ-add inserts the information on the control channel into that aggregate signal.

Figure 2.6.3 depicts the node architecture in the FTx^C-FRx^C configuration.

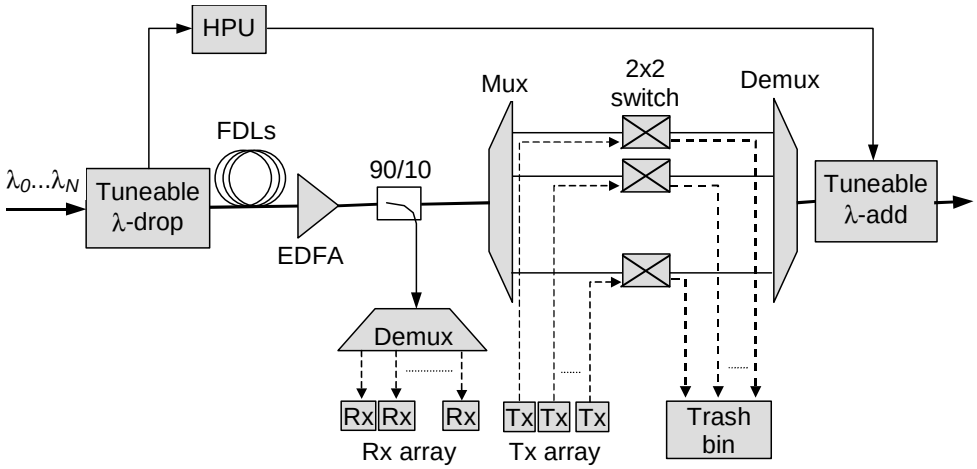


Figure 2.6.3 – Architecture of the Flamingo node

2.7 Discussion

The MOPS rings described in this chapter follow the same concept: transport of headers and corresponding payload data separately and use of FDLs to delay payload data while the corresponding header is obtained and processed. Based on

the outcome of the header processing nodes configure the OADM node to either forward, or receive the payload.

The differences among the MOPS rings are in the physical implementation of the signalling mechanism, the transceiver configuration of the nodes, and the OADM technology. Such differences influence not only the functionality, but also the performance, the scalability, the complexity, and the cost of the network.

For instance, only the Flamingo network can support optical multicast. All the other networks either use unicast to support multicast, or use multi-hop forwarding.

Consider MAWSON, RINGO, and TT-Pipeline. In these networks a node can drop only one wavelength, which is assigned by design. Thus, to deliver a multicast packet to all the destination nodes a source node has to transmit on each wavelength on which there is at least one destination node receiving (see Figure 2.7.1).

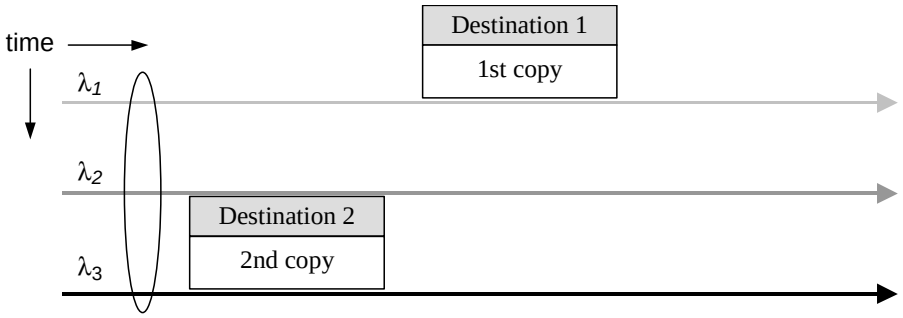


Figure 2.7.1 - Multicast via optical unicast in MAWSON, RINGO, and TT-Pipeline

Consider now BORN. In this network a node can drop all the wavelengths simultaneously, but it cannot drop and forward a given channel simultaneously. Thus, to deliver a multicast packet to two or destination nodes a source node has to unicast one copy of the packet to each destination node (see Figure 2.7.2).

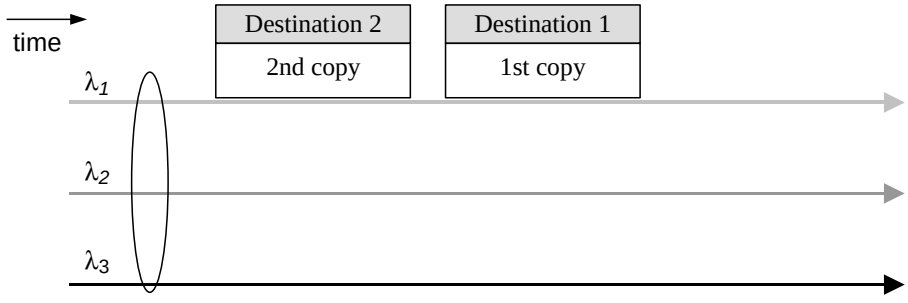


Figure 2.7.2 - Multicast via optical unicast in BORN

For either alternative to work though, a source node should know the identity of the destination nodes and their reception wavelengths. This could be the case in networks running Multicast extensions to Open Shortest Path First (MOSPF)

routing protocol [Moy1994] extended to carry wavelength information [Wang2000], for instance. Nevertheless, in networks running routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP) [Waitz1988], Protocol Independent Multicast – Dense Mode (PIM-DM) [Adams2002], and PIM – Sparse Mode (PIM-SM) [Estri1998], a source node does not know the identity and the location of the destination nodes.

Thus, unless some mechanism is designed to discover the identity of the destination nodes, their logical segment and their reception wavelength, the only way to assure that each incoming multicast packet will reach each destination node is to unicast one copy of the packet to each node in the network. In other words, the only way is to flood the network with copies of the packet.

Furthermore, each copy is transmitted on a different time slot. At high workload conditions, access delays increase and, consequently, destination nodes may receive the same information at very distinct instants of time. Needless to say that this is a problem not only to certain delay-intolerant applications (e.g., cooperative work), but also to certain network protocols upon which the network rely to function properly.

To support of optical multicast though, destination nodes in Flamingo cannot reuse slots that they release since add operations take place first than drop operations. Therefore, the Flamingo network cannot use to entire ring capacity; given the same ring length and the same number of slots, the waste of capacity increases as the number of nodes decreases.

Another example is fault tolerance. For instance, in MAWSON and in RINGO each node is assigned a unique channel, and a node that wishes to communicate with another node has to transmit on the reception channel of the latter. Since a node always drops its reception channel, that source node does not need to transmit any separate header containing forwarding information; consequently, packet misdelivery cannot occur. Packet live-lock (that is, packet rotates forever) can still occur if the destination node goes out of work, but that does not affect the functioning of the network because the entire channel becomes useless anyway.

As in electronics networks, in BORN and in FLAMINGO packet misdelivery and packet live-lock can occur as a result of transmission errors, digital mishandling, or node failure. What is more, their occurrences prejudice the network performance since channels are not bound to specific nodes and nodes need to evaluate slot headers before deciding what to do with their corresponding payload data. Therefore, mechanisms are required to cope with such events when they occur.

Chapter 3
Existing MAC protocols for MOPS rings

This chapter describes some of the existing MAC protocols for MOPS rings. As it has been the norm [vanAs1994a] the protocols described in this chapter derive from techniques developed in the early 1990s.

Note that this chapter focuses on protocols for single-fibre rings. For information on protocols for interconnected rings refer to [Chlam1999].

3.1 Empty slot contention/collision avoidance (ESCA)

Empty slot contention/collision avoidance (ESCA) [Chlam1995] is the MAC protocol designed for the Pipeline network. ESCA is a slotted ring with destination removal protocol, and it uses slot-synchronization across the channels.

There is one version of ESCA for each version of Pipeline, and both assume that packets and slots have the same size.

The ESCA protocol for TT-Pipeline is very simple. Since each node can receive on a single unique pre-assigned channel, destination contention and access contention become the same thing. Therefore, ESCA allows a node to transmit whenever an empty slot arrives.

The ESCA protocol for TR-Pipeline is slightly more complex. Since each node transmits on a unique pre-assigned channel, but receives on a single arbitrary channel at a time, destination contention and access contention become distinct issues. Therefore, ESCA allows a node to transmit if both conditions are met: i) an empty slot arrives; and ii) no busy slot on any channel goes to the same destination as the packet selected for transmission.

Reference [Chlam1995] does not mention how the protocol copes with unfairness.

3.2 BORN

Due to the FTx^{c}-FRx^{c} configuration of the BORN network, destination contention cannot occur, and there is no need for tuning Tx units or Rx units. For this reason, and because packets and slots are assumed to be of the same size, the MAC protocol is the typical slotted ring with destination removal protocol: transmission can occur only if an empty slot arrives.

Reference [Kang1995] does not mention how the protocol copes with unfairness.

3.3 Synchronous round-robin (SRR)

Designed for networks that follow the TTx[1]-FRx[1] configuration, synchronous round robin (SRR) [Marsan1996, Marsan1997] builds on slotted ring with destination removal to provide random, collision-free access to the medium. To achieve almost optimal access, SRR employs a global scheduling that, under heavy load traffic conditions, forces the behaviour of the network to that of time division multiplexing access (TDMA) networks with static assignment of slots.

Let N be the number of network nodes and C be the number of wavelength channels. Assuming that N is an integer multiple of C, $D = N / C$ nodes share the same reception channel.

Each channel can be divided into D logical partitions, each comprising $N / D = C$ adjacent nodes. The nodes are equally spaced and disposed within each partition in a sequence that reflects their reception channel. That is, the D nodes sharing the i-th reception channel, with $0 \le i < C$, are in positions $|i + dC|_N$, with $0 \le d < D$. Only the first node of a partition can receive and transmit on the channel associated to that partition; the others node can transmit, but not receive on that channel.

Figure 3.3.1a illustrates the SRR network topology of an eight-node, four-channel ring, and Figure 3.3.1b illustrates the logical partitioning of channel 0, the reception channel assigned to node 0 and node 4.

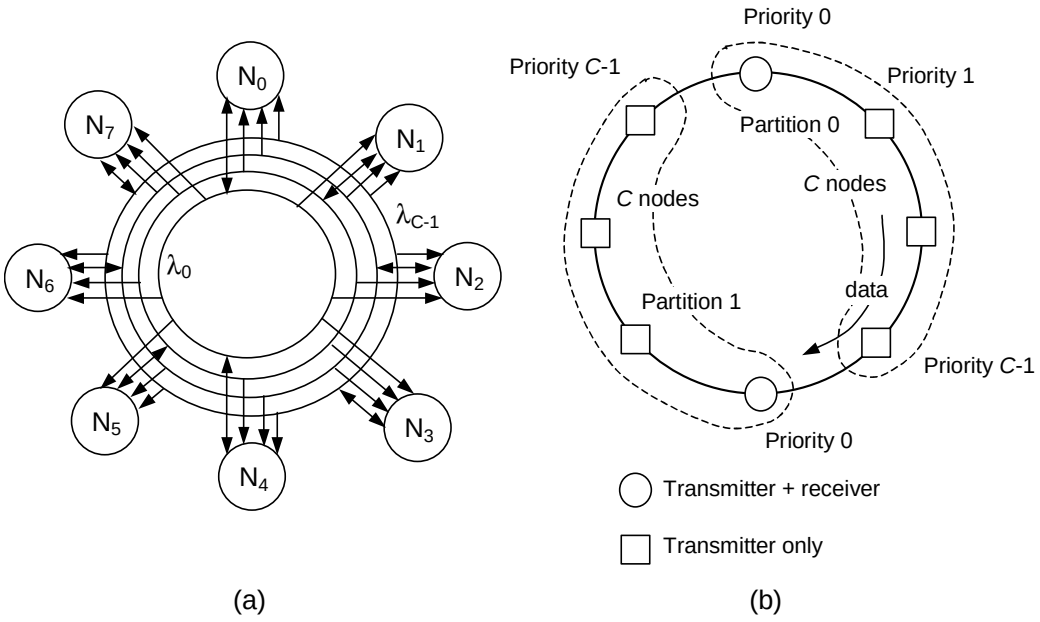


Figure 3.3.1 - (a) Network topology; (b) Logical partitioning

As a consequence of ring symmetry, some nodes have better-than-average access to the ring, while some nodes have worse-than-average access to the ring. In other words, when transmitting to a node j, a node i has priority $|i - j|_N$, whereas 1 is the highest priority and N-1 is the lowest priority.

SRR uses virtual output queuing (VOQ) to avoid head-of-line (HOL) blocking. Each node maintains one queue per each possible destination node. In an arbitrary time slot identified by a label *s*, node *i* schedules for transmission the HOL packet from the queue destined to node $|i+k+1|_N$, where $k = |s|_{N-1}$. If the corresponding queue is empty, the scheduler attempts transmission of the longest queue's HOL packet. If two or more longest queues exist, the scheduler selects the lowest priority queue. In either case, transmission can occur only if the slot is empty.

Figure 3.3.2 illustrates the principle of operation of SRR in a ring with four nodes, four channels, four time slots, and a scheduling cycle (or frame) of three ring rotations or three access opportunities per node. Note that each slot shows three packets, each with a label identifying the node that transmitted it. The order of these packets indicates that their transmission occurred in subsequent ring rotations.

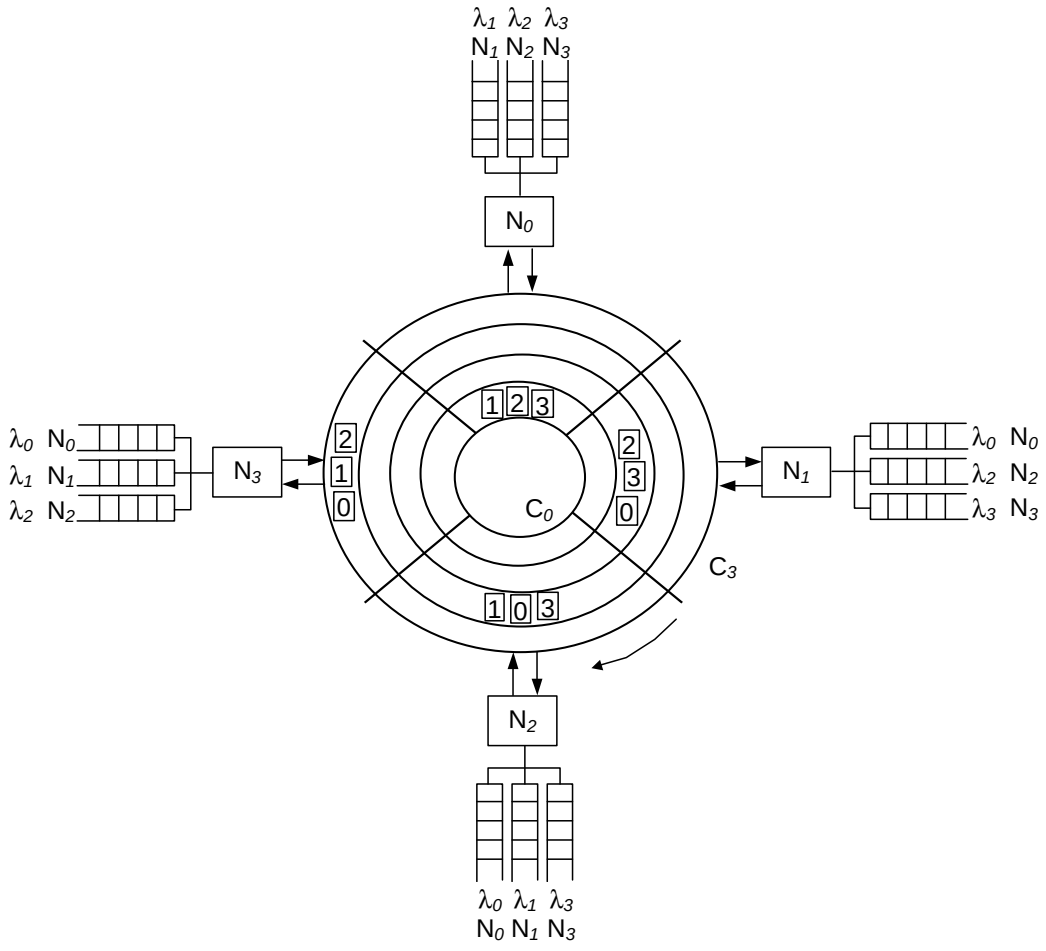


Figure 3.3.2 - Principle of operation of SRR with N = C = 4 nodes and four time slots

SRR uses a static global fairness protocol called SAT [Ofek1994] to ensure access fairness. Chapter 5 describes the SAT algorithm in detail.

3.4 Request/Allocation protocol (RAP)

Designed to regulate access in the MAWSON network, Request/Allocation protocol (RAP) [Summe1997, Frans1998] is a contention-free slotted ring protocol. A source node that wishes to transmit to a particular destination node must request bandwidth from that destination explicitly. Only after an allocation confirmation returns, the source node can transmit.

The total capacity of each channel is divided into slots of equal, fixed length. The structure of the slots consists of a header section followed by a data section. The data section consists of M data minislots (DMSs). The header section contains a synchronization minislot and N-1 request/allocation (R/A) minislots, each pre-assigned to a particular node that can transmit on the channel of that slot. More specifically, R/A minislot $i \neq j$ can be either used by node i to request DMSs on channel j, or used to allocate DMSs on channel i to node j.

Figure 3.4.1 depicts the layout of a frame (that is, a slot).

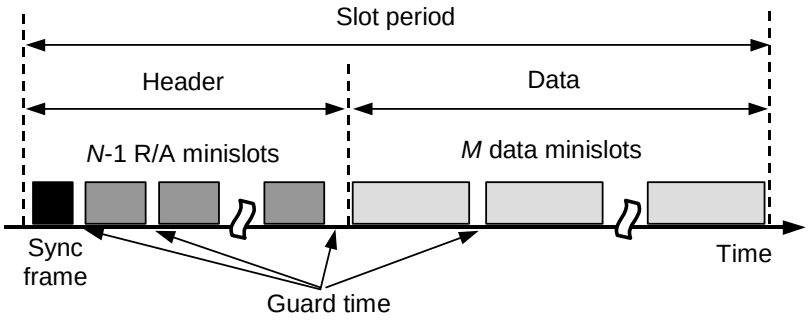


Figure 3.4.1 - Structure of the slots in RAP

An R/A minislot contains the following fields: synchronization preamble, beginning of frame (BOF), request, allocation, and end of frame (EOF). The request field consists of $\lceil \log_2(M+1) \rceil$ bits, where $\lceil x \rceil$ denotes the ceil operator. The allocation field consists of M bits. The i-th allocation bit indicates to a node that the i-th DMS has been allocated to that node.

RAP assigns R/A minislots across all the channels to distinct nodes via a cyclic permutation to guarantee that a node is never allocated DMSs on distinct channels in the same time slot. This prevents waste of bandwidth because each node can transmit on only one channel at a time.

Slots are synchronized in parallel across the wavelengths so as to reach each node all at the same time, hence, avoiding transmission collisions.

Figure 3.4.2 illustrates the synchronization of slots in the network.

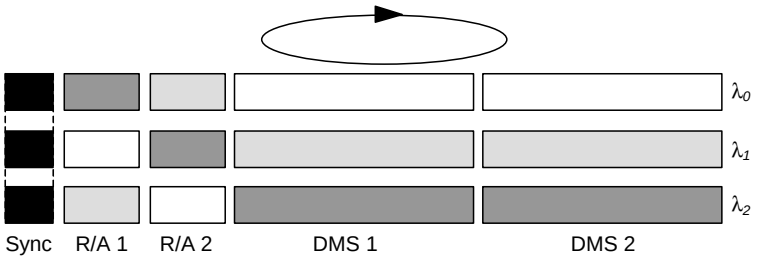


Figure 3.4.2 - Organization of slots across the channels; the shade of each R/A minislot shows which node this minislot is allocated to

To avoid HOL blocking, RAP uses VOQ. Each node maintains N - 1 queues, one for each possible destination. A node requests a number of DMSs that is sufficient to transmit a HOL packet from a particular queue. The maximum number of DMSs a node can request at a time is M.

To enforce access fairness, nodes perform round-robin allocation. That is, a (destination) node cyclically allocates DMSs to each (source) node in sequence until either all the requests are granted or all the DMSs are allocated. Allocations in the next slot start from the node that is next in sequence to the last node granted with allocations.

Since processing of requests consumes time, an allocation occurs in the slot following the one in which the request was received. For the same reason, actual transmission occurs in the slot following the one in which the allocation was received. Therefore, in RAP a successful transmission procedure consists of three stages and it spans over three slots.

3.5 Multitoken interarrival time (MTIT)

The multitoken interarrival time (MTIT) protocol [Cai2000] is designed to regulate access in single fibre, multi-channel ring networks consisting of nodes whose architecture follows that shown in Figure 3.5.1.

The architecture follows an FTx^{C}-TRx^{C} configuration. Therefore, a node can transmit and receive on all the data channels simultaneously.

Figure 3.5.1 depicts the node architecture considered by the MTIT protocol.

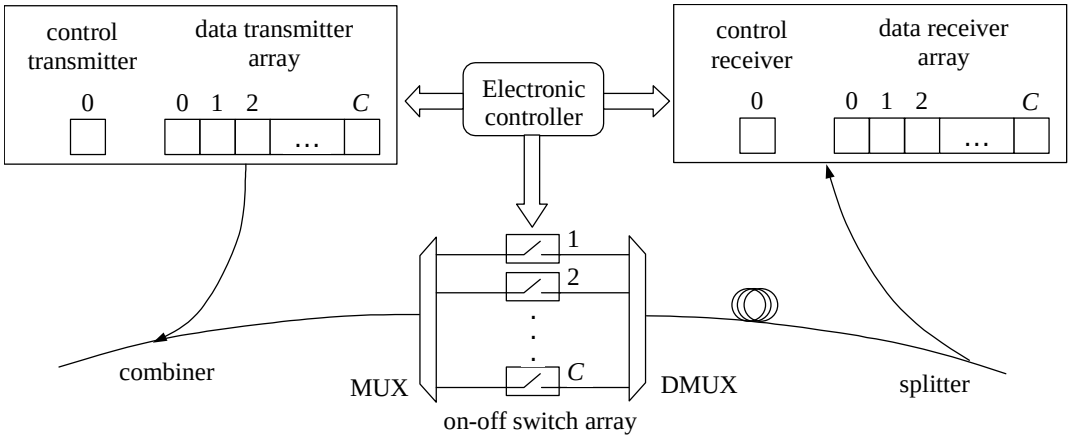


Figure 3.5.1 – Node architecture assumed in the MTIT protocol

Multiple tokens circulate in the ring, each associated with a particular channel. A token regulates access to the channel it is associated with.

A node can transmit on a given channel only if that node possesses the token that corresponds to that channel. A node that transmitted a packet is responsible for removing that packet from the ring.

A previously agreed target token interarrival time (TTIT) controls the THT dynamically. Upon arrival of a token, a node is allowed to hold that token for a period of time equal to TTIT - TIAT, where TIAT is the actual token interarrival time between that token's arrival time and the arrival time of the token held previously. If THT is up, then the node must finish the ongoing transmission and release the token. If TIAT exceeds TTIT, then the token is late and must be released immediately.

In any case, if a node has no packets to transmit, then that node must release the token immediately.

Since MTIT uses source release, it can guarantee fair access to the ring if the nodes' timers operate within a certain timing tolerance and the maximum packet length is bounded.

3.6 Carrier sense multiple access with collision avoidance (CSMA/CA)

Carrier sense multiple access with collision avoidance (CSMA/CA) [Shrik2000b] is the MAC protocol developed for the HORNET network. Two versions of CSMA/CA exist: slotted CSMA/CA with multiple slot sizes and unslotted CSMA/CA with back-off.

Slotted CSMA/CA with multiple slot sizes

The first scheme uses multiple slot sizes to match the packet size distribution of the network; a slot governor node performs the centralized time slotting function on each wavelength.

Besides the typical access rule of the slotted ring media technique -access can occur only if the slot is empty, transmission of a packet can occur only if that packet fits entirely in the incoming slot.

To avoid HOL blocking, the protocol uses VOQ. There are *C* queues, where *C* denotes the number of channels in the network, and each queue is associated with a particular channel.

To determine whether a packet fits into a slot, or does not, the protocol uses a common control channel to carry slot size information and define slot boundaries. Thus, the slots in parallel across the wavelengths have to be of the same size. Figure 3.6.1 illustrates such an organization.

An AP senses the carrier on each wavelength as usual. It also obtains the slot size information from the control channel. If the slot is empty, and the HOL packet of the corresponding queue is smaller than the slot size, or equal, then that packet is selected and transmitted.

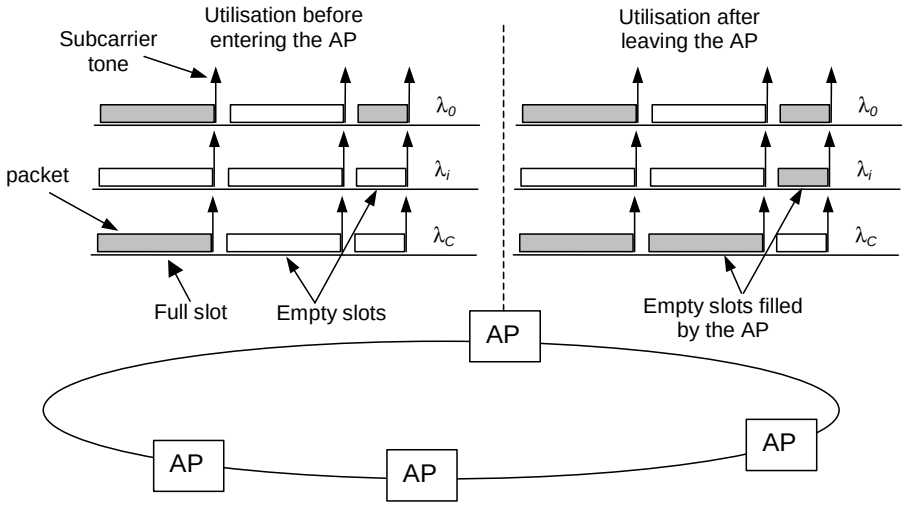


Figure 3.6.1 - Slotted CSMA/CA with multiple slot sizes

It is also possible to encode slot boundary and slot size information into the subcarrier of each channel separately. By doing so, slots across the parallel wavelengths could have different sizes.

Unslotted CSMA/CA with back-off

The second scheme aims at asynchronous access.

An AP senses the subcarrier of each channel as usual, and if the channel is idle then the AP selects the HOL packet from the corresponding queue, and transmits that packet.

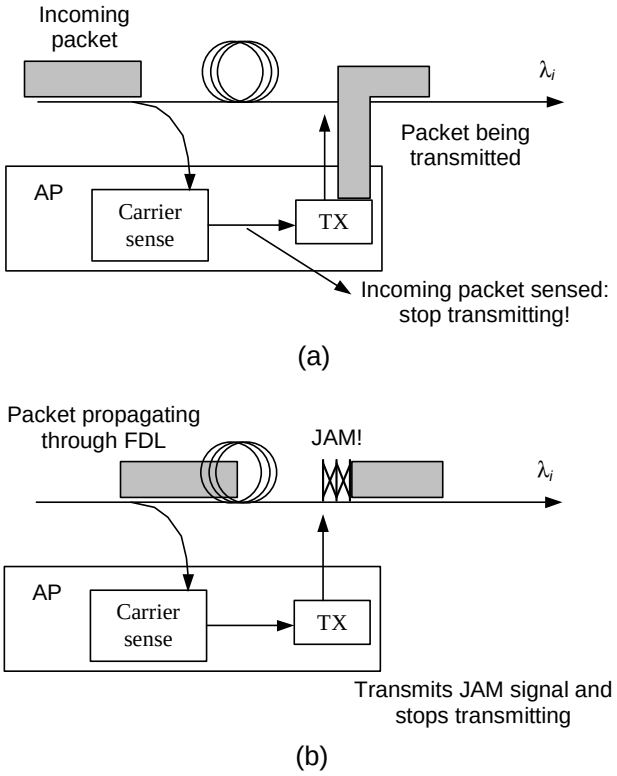


Figure 3.6.2 - Unslotted CSMA/CA; a) carrier sense; b) collision avoidance

Subcarrier sensing continues even during transmission though. If the AP detects transit traffic on the channel it is transmitting on, then that AP stops transmitting (see Figure 3.6.2a), and sends a jamming signal to the packet's destination AP to inform that the transmission has been interrupted, and the received fragment should be discarded (see Figure 3.6.2b). The source AP attempts retransmission of that packet later, following the access rules as usual.

The jamming signal could be a unique bit pattern, either at baseband or on the subcarrier of the corresponding channel.

Reference [Shrik2000b] does not specify any fairness algorithm, for either protocol, to cope with unfairness.

3.7 Discussion

Each of the protocols described in this chapter has been designed with a particular node architecture and signalling technique in mind. The main advantage of such an approach is optimisation. For instance, RAP does not need mechanisms to prevent packet misdelivery or packet live-lock because such events do not affect the MAWSON network. The main disadvantage of this approach though is that it limits the use of each protocol to the homogeneous transceiver configuration and the signalling technique assumed in the design of the protocol.

Because of such a protocol design approach it is difficult to compare the protocols described in this chapter, in particular with respect to performance.

RAP, MTIT, and the two variants of CSMA/CA aim to transmit variable size packets entirely, as single units. Nevertheless, RAP and MTIT use source removal, which prevents them from exploiting the total capacity of the medium, or in other words, which limits the maximum achievable throughput to 1.

Figure 3.7.1 [Frans1998] shows the achievable aggregate throughput of RAP under uniform traffic load as a function of number of nodes and ring lengths using the achievable aggregate throughput of the slotted ALOHA [Klein1973] as comparison; the channel bit rate is 100Mb/s.

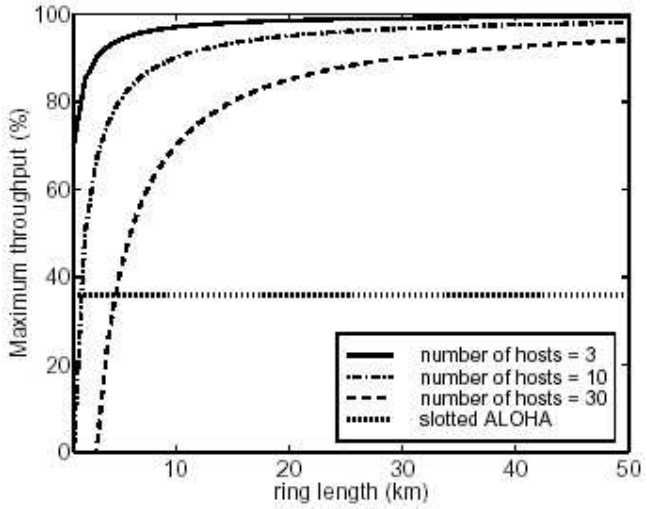


Figure 3.7.1 – Achievable aggregate throughput in RAP for different ring lengths and number of nodes, and for slotted ALOHA under uniform traffic load and using 100Mb/s capacity channels [Frans1998]

Figure 3.7.2 [Cai2000] shows the achievable throughput of MTIT for an 8-node, 80km-ring with 80 Gb/s total capacity and various numbers of channels under uniform traffic load consisting of packets with exponentially distributed lengths.

The obtained results show that bandwidth efficiency and access delay improve with the number of channels in the ring. Nevertheless, such scalability comes at high cost and extra complexity since MTIT assumes an FTx^{C}-TRx^{C} node configuration.

It remains to be seen how RAP and MTIT behave under non-uniform traffic loads. Intuition points to lower performances, but intuition can be very tricky, so no comment can be done on this matter until concrete results become available.

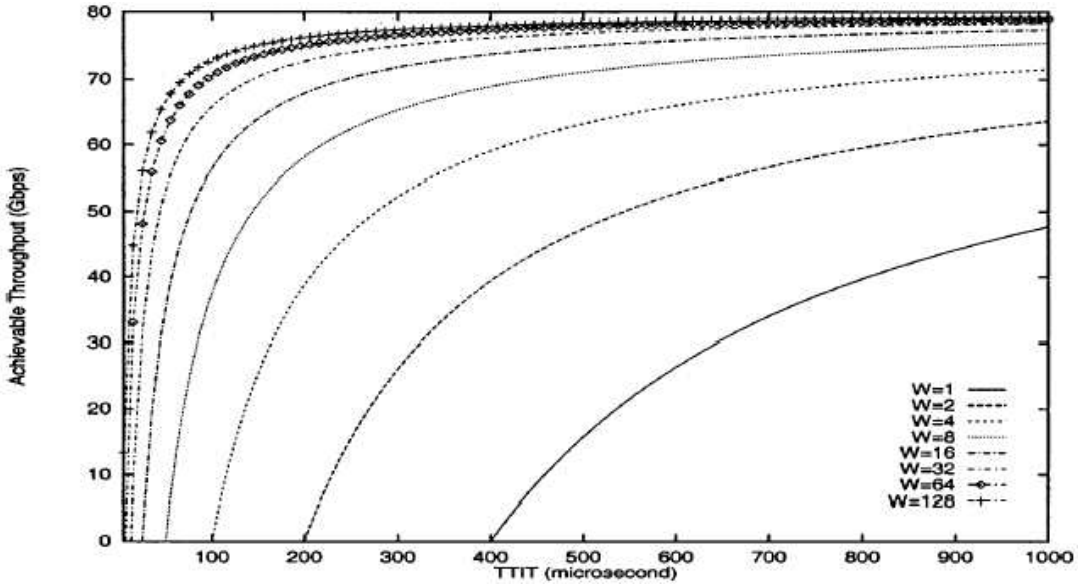


Figure 3.7.2 - Achievable throughput of MTIT for an 8-node, 80km-ring with 80 Gb/s total capacity and various numbers of channels under uniform traffic load consisting of packets with exponentially distributed lengths [Cai2000]

The two variants of CSMA/CA use destination removal and, therefore, they have the potential to exploit the total capacity of the medium.

Shrikhande et al. [Shrik2000b] shows performance results collected by simulating both slotted CSMA/CA and unslotted CSMA/CA, and comparing the results with those of slotted ATM CSMA/CA [Shrik2000a] –ATM CSMA/CA performs SAR operations to force packets to fit into slots.

The simulation set-up consists of an AP through which passes ten 2.5Gb/s channels. Markov modulated sources, whose probabilities are shown in Figure 3.7.4, insert bursty traffic into each channel, and into each AP's queues.

Figure 3.7.3 depicts the scenario used in the event-driven simulations.

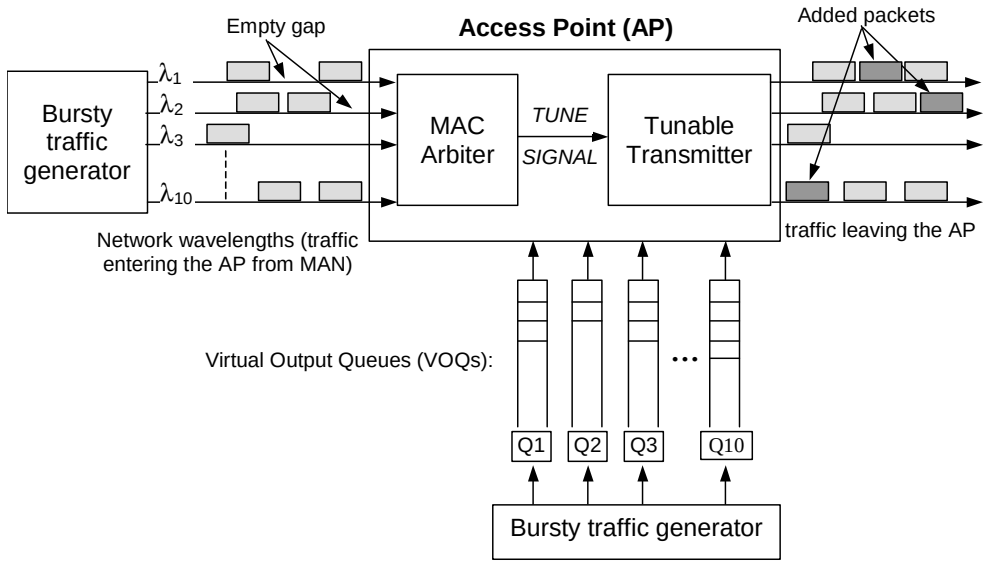


Figure 3.7.3 - Functional model of the AP used in the simulations

For the ATM CSMA/CA protocols, all the slots are 53B long. For the slotted CSMA/CA, slots are 40B, 552B, and 1500B long with a probability distribution that matches the packet size distribution of the Internet.

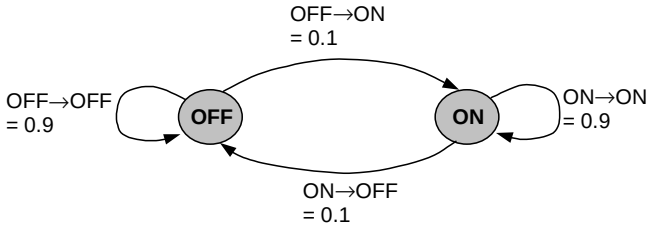


Figure 3.7.4 – Modulated Markov probabilities

Figure 3.7.5 [Shrik2000b] shows the AP throughput versus the load inserted into the channels when the queues are full, whereas throughput is the number of bits transmitted by the AP per second. Note that the graph uses the terms Slotted IP and Unslotted IP to represent the slotted CSMA/CA with multiple slot sizes and the unslotted CSMA/CA with back-off protocols.

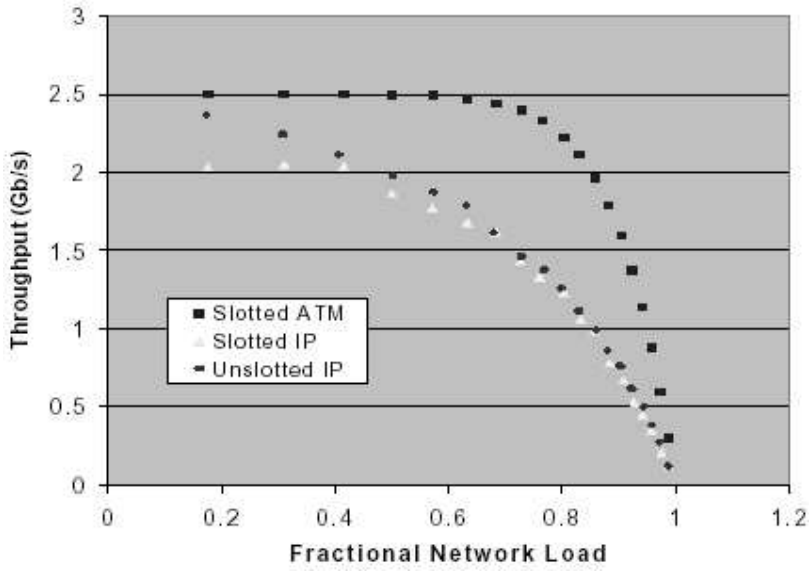


Figure 3.7.5 - Node throughput versus network load in CSMA/CA protocols [Shrik2000b]

As expected, ATM CSMA/CA achieves the best throughput. That is because packets fit exactly in the slots and, therefore, are never prevented from being transmitted if there is opportunity to do so. As the network load approximates 60% throughput starts to degrade until the load reaches 80%, when the throughput drops abruptly, meaning that the stability condition has been broken.

The curves of slotted CSMA/CA and unslotted CSMA/CA are similar, and show a reasonable smooth degradation as the network load increases. At low loads in slotted CSMA/CA, the likelihood of the AP find an empty slot that can transport a big packet is higher than the likelihood of find an empty slot that can transport a packet of the same size at a higher load. As the load increases, HOL blocking increases with higher probability for bigger packets, thus explaining why throughput drops as the load increases.

At low loads in unslotted CSMA/CA, the likelihood of the AP find a channel that stays idle for the time necessary to transmit a big packet is higher than the likelihood to find a channel that stays idle for the time necessary to transmit a packet of the same size at higher loads. Consequently, as the load increases, so does the likelihood of big packets being retransmitted.

It is important to note that the simulations did not consider real traffic scenarios, in which the access fairness protocol may have to act to cope with nodes seizing the medium under certain traffic patterns. Under such scenarios the performance obtained by the variants of CSMA/CA may be considerably different.

The other protocols described in this chapter are all derived from the slotted ring with spatial reuse media technique, and they assume packet size and slot size to be equal. Therefore, to transport variable size packets over fixed size slots these protocols require the use of SAR operations.

The great benefit of SAR is that it allows for the transport of variable size packets over fixed size slots without modifying the MAC protocols. Besides, SAR operations are simple and result in high multiplexing efficiency.

SAR can be accomplished either at the IP layer (that is, the network layer), or at the MAC layer. The advantage of doing SAR at the IP layer is that the latter is already supposed to do SAR to make sure that packets are never bigger than the maximum transmission unit (MTU) of the link. Thus, the only thing that the MAC layer has to do is to set the MTU to the slot size.

Despite its simplicity, such an approach has drawbacks. First, forcing the IP layer to segment packets into slot size fragments transfers the processing overhead of SAR operations from the MAC layer to the IP layer. Although in MOPS rings transmissions bypass intermediate nodes, it is unlikely that transmissions can go all the way in the optical domain. Rather, it is likely that transmissions undergo electronics processing at some interconnecting points. Since IP re-assembles packet fragments only at the end destination, those interconnecting points have to process all the fragments, hence experiencing higher processing overhead and achieving lower performances. Moreover, it may not be possible to implement network layer SAR operations in hardware for speedup.

Second, by segmenting packets at the IP layer each generated fragment contains a copy of the PCI contained in the original packet. When the MAC protocol receives a fragment it frames that fragment before transmitting it, and each frame may include additional PCI that the MAC protocol needs to work properly. Therefore, forcing the IP layer to segment packets into slot size fragments also results in additional network overhead.

Third, IP version 6 (IPv6) [Deeri1995] demands an MTU of at least 1280B, and it does not permit packet fragmentation at routers, only at source hosts. Therefore, forcing the upper layer to segment packets into slot size fragments may impact the network overall performance severely.

Clearly, SAR should take place at the MAC layer, and the MTU should be defined according to parameters such as loss probability and corruption probability.

Performing SAR at the MAC layer has drawbacks though:

- Destination nodes have to store fragments until they can be re-assembled and sent up to the upper layer. Given the number of possible simultaneous communication sessions between a given destination node and any other node, considerable memory has to be available, even if an average number of sessions rather than maximum is assumed to calculate memory requirements. Anyway, a destination node may run out of memory and, consequently, discard fragments. Such an action may impact the corresponding application, the corresponding protocol, or even the network overall performance;
- Depending on communication patterns and the access mechanism, it may happen that many re-assembly operations take place at the same time at a given node. The simultaneous execution of many re-assembly operations

may result in a burst of packets being sent to the upper layer. If the upper layer cannot process all the packets simultaneously then it applies packet discard, whose effects have already been mentioned.

For more detailed information on the issues related to SAR operations refer to [Moors1997].

Chapter 4
Access control protocols

This chapter introduces four access control protocols to transport variable-size packets in MOPS rings. The protocols aim at transceiver configuration independence and heterogeneity and do not assume any particular signalling technique.

4.1 Preliminaries

The protocols follow the conceptual node architecture shown in Figure 1.1.5. To achieve the goals mentioned above, the design of the protocols is based on an open system model and the description of the protocols is intentionally general.

Figure 4.1.1 depicts the open system model. The open system model relies on the global knowledge of the transceiver configuration of the network nodes. It decomposes the system into components that are grouped according to the layer and the domain to which they belong.

The PHY layer comprises the OADM, the Rx units and their RBs, and the Tx units and their TBs. The MAC layer comprises the following modules: add-drop decision making (ADDM), the reception and transmission decision making (RTDM), the channel selection, the packet scheduling, the queuing system, the transceivers information base, and the access control and fairness.

The ADDM decides upon either dropping or forwarding a packet based on the processing of the header of that packet. If a drop decision is made then the ADDM signals the OADM to drop the corresponding channel. The ADDM also decides upon adding a new packet upon arrival of a signal from the RTDM. If an add decision is made then the ADDM signals the corresponding Tx unit to transmit a packet from the corresponding TB and the OADM to add the corresponding channel.

The RTDM processes any packet arriving into an RB and decides upon transmitting upon arrival of an empty slot signal from the ADDM. To transmit the RTDM interacts with the other MAC modules as follows.

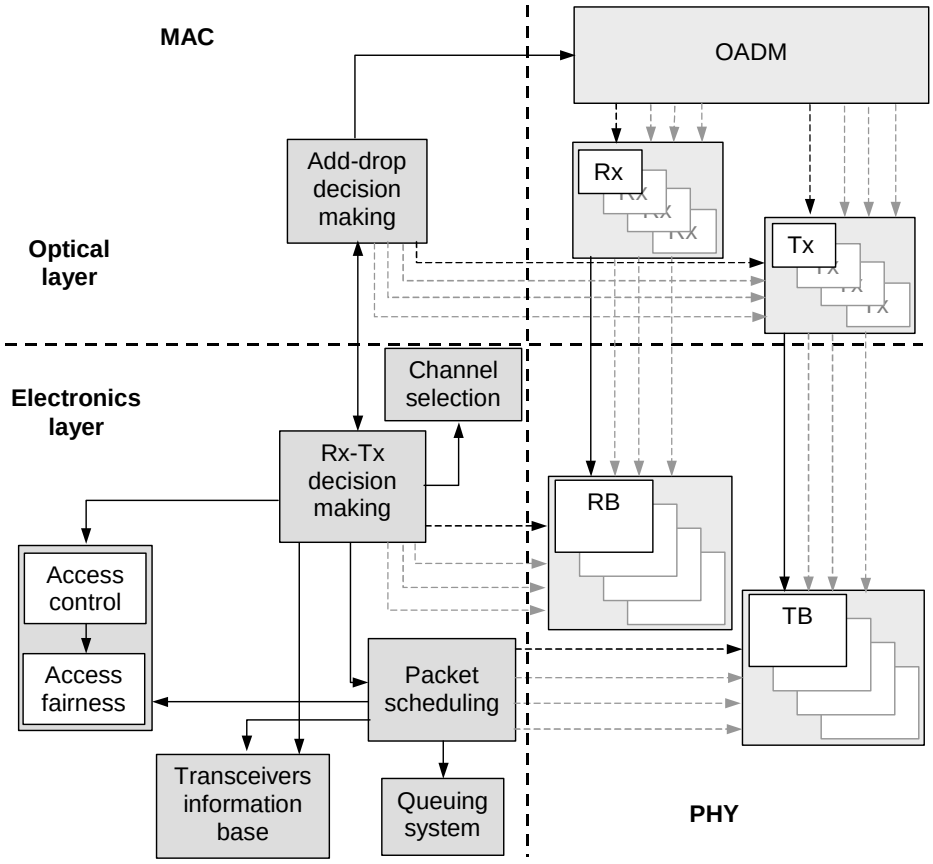


Figure 4.1.1 – Open system model

First, if the number of parallel empty trains is greater than the number of Tx units of that node, then the RTDM interacts with the channel selection module to select the trains (that is, channels) to access. Various selection strategies are possible, such as random and first find.

Second, the RTDM signals the packet scheduler to select a packet to be transmitted on the selected channel. Packet scheduling is an integrated a-posteriori process that involves a particular scheduling algorithm (e.g., random, round robin), the packet queue(s), the transceivers information base, and the fairness algorithm -the implementation of the scheduling process is implementation specific and out of the scope of this work. If the packet scheduler succeeds then it inserts the selected packet into the corresponding TB and signals the RTDM to inform the ADDM about the new packet.

The open system model allows for the relaxation of the dependence between the vertical and the horizontal domains, whereas the vertical domain is concerned with access across channels and the horizontal domain is concerned with access across time slots on a single channel. Figure 4.1.2 illustrates the relation between the vertical and the horizontal domains.

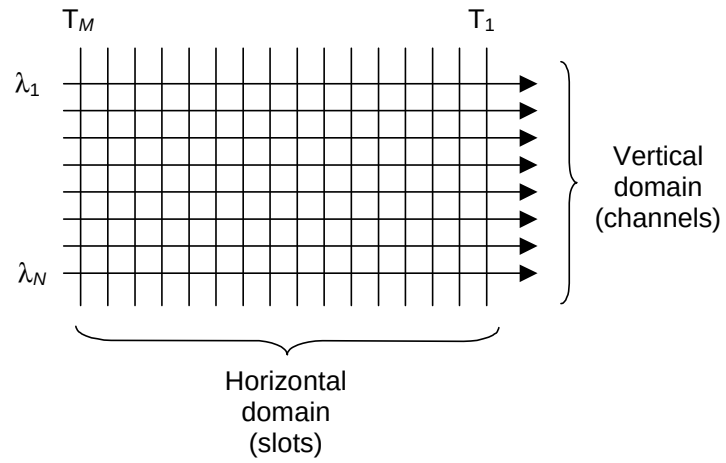


Figure 4.1.2 - Relationship between the vertical and the horizontal domains

The descriptions of the protocols assume a well-defined interface between the PHY layer and the MAC layer; they abstract the implementation details of the physical layer upon which the MAC layer relies. For the protocols what matters is whether a slot is either empty or busy, not how the slot status information is obtained; such concern is of the responsibility of the PHY layer.

Figure 4.1.3 shows an example of such interfaces. In the example, the PHY layer defines an interface with four primitives: PHY_HEADER.indication, PHY_HEADER.insertion, PHY_DATA.Tx, and PHY_DATA.Rx. The PHY layer issues a PHY_HEADER.indication upon detection of an incoming slot header, and that primitive may contain a slot header. The MAC layer issues a PHY_HEADER.insertion to update the header just processed. The MAC layer issues a PHY_DATA.Tx to transmit effectively -the primitive contains the data to be transmitted and informs the channel to add, and a PHY_DATA.Rx to inform the PHY layer to drop a specified channel.

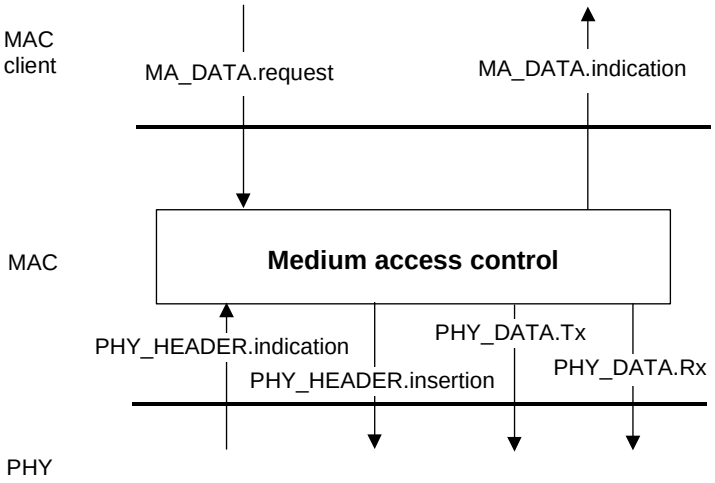


Figure 4.1.3 - Example of interface between the PHY and the MAC layers

To not prescribe any signalling technique implementation the description of the protocols does not define concrete slot header frame layouts. Concrete frame layouts are defined only for payload data, and they use 32-bit word alignment as recommended by the IETF.

For the same reason the description of the protocols uses the symbols BUSY, EMPTY, TRUE, FALSE, and NULL to represent certain signals and field values. BUSY and TRUE may be the equivalent to the presence of a signal in an implementation and to a certain field value in another implementation. EMPTY, FALSE, and NULL may be the equivalent to the absence of a signal in an implementation and to a certain field value in another implementation.

BUSY and EMPTY are used to indicate the status of slots, TRUE and FALSE are used with conditional fields that can assume only two values, and NULL is used with non-conditional fields.

It should be noted that payload data can have variable size, whereas the maximum size is limited either by the *MTU* of the link or by contention (see Figure 4.2.5 for an example). Therefore, whichever the chosen signalling technique, slot headers should always serve as synchronization references for payload slots.

The synchronization with a dropped payload occurs upon detection of both a BOF sequence that precedes every frame and an EOF sequence that follows every frame. To prevent interfaces from interpreting BOF and EOF sequences that might appear in payload data as indeed BOF and EOF sequences, the protocols use data stuffing [Peter2000]. Data stuffing inserts an escape signal in front of BOF or EOF sequences to mask them whenever they appear in the payload data. A source node inserts such an escape signal if necessary before transmitting, and a destination node removes such a signal when it receives the payload data.

The protocols follow the end-to-end argument in system design [Saltz1984, Blume2001]. The end-to-end argument says that applications have different requirements and that communication systems can implement certain functions correctly only with the knowledge of those applications standing at the end points. Because of the difficulty to meet all the applications requirements and since applications often implement the functions they need themselves, communication systems would better not to implement certain functions; doing so would result in the duplication of functions, with possible performance degradation at both the overall communication system and the applications that do not need such functions.

As recommended in [Carpe1996, Fairh2002, Karn2002] for networks with characteristics that are similar to those of MOPS rings, the protocols do not attempt to guarantee perfect communications. Nevertheless, for consistency the protocols include mechanisms to detect errors and data loss and to react to their occurrences promptly. For the sake of legibility such mechanisms are discussed only in Section 4.5.

The description of the protocols use the term slot train as an analogy to packet train [Jain1986] to define consecutive slots coming from the same source, going to the same destination, and carrying the same packet; a slot train may refer to

consecutive empty slots as well, but not the other way round. The first slot of a train is referred to as head and the last slot of a train is referred to as tail; note that in a slot train with a single slot the head slot is the tail slot too.

The next sections contain the description of the protocols. For legibility, they exploit the relaxation of the dependence between the vertical domain and the horizontal domain to focus on access from a single-channel perspective at first, hence concentrating on the horizontal contention domain, which is paramount to OPS. Section 4.4 deals with the effects of channel multiplicity on the protocols.

4.2 Conflict-free protocols

This section describes two conflict-free access control protocols: packet aggregate transmission (PAT) and slotted packet transmission with slot concatenation (SPT+SC).

4.2.1 PAT

PAT [Salva2002b] ensures that every packet is transmitted as an indivisible unit. Once a node starts transmitting a packet that node is guaranteed to transmit that packet entirely, using a single slot. Likewise, once a destination node starts receiving a packet that node is guaranteed to receive that packet entirely, in a single slot.

PAT follows [Karn2002], which advises that if frames have fixed size then they should be large for throughput, whereas the actual size depends on the bit error rate (BER) of the medium, the access delays, and the access delay variations. The larger the slot size, the lower the PCI overhead is; consequently, the higher the throughput is. Also, the larger the slot size, the lower the slot-forwarding rate is; consequently, the higher the achievable channel bit rates are.

Large slots also benefit signalling schemes such as that of Flamingo, in which the size of payload slots must equal the size of control slots, or be greater than that, for the network to function properly (recall that the number of slot headers is affected by the number of payload channels supported).

It is not possible to find an optimum slot size to transport Internet packets efficiently. Thus, to cope with the mismatch between the slot sizes and the packet sizes PAT multiplexes as many packets as possible into a slot.

Figure 4.2.1 illustrates how PAT works. In the example, a given node has a backlog of four packets, labelled *a*, *b*, *c*, and *d* to express the sequence in which they are scheduled for transmission. Packets *b*, *c*, and *d* have the same destination; packet *a* has a different destination.

Since a node can multiplex many packets into a single slot, and since a slot contains a single header, to prevent packet misdelivery in PAT only packets travelling to the same destination node can be transmitted in the same slot. That explains why the first slot carries only packet *a*.

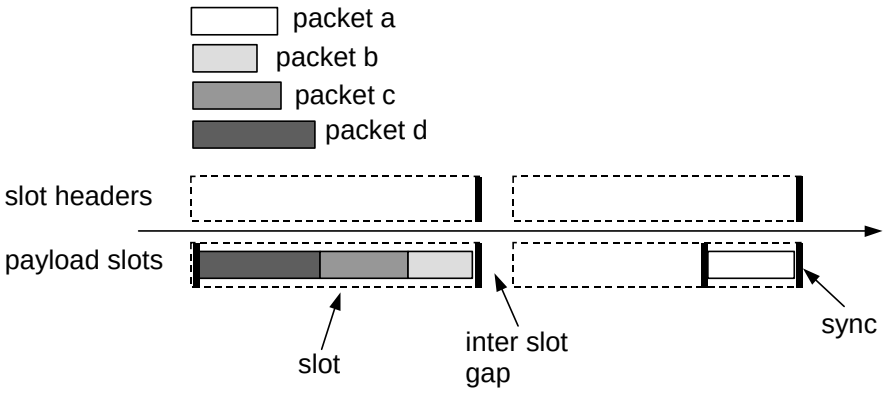


Figure 4.2.1 - Illustration of the functioning of PAT

Such a constraint can cause HOL blocking, and to avoid that PAT uses VOQ. Each node maintains *N*-1 queues, one per possible destination node, a single TB capable of storing one MTU-sized packet, and a single RB also capable of storing one MTU-sized packet. The packet scheduler serves the packets queue according to the first-come-first-serve (FCFS) discipline, and moves the HOL packet from the queue to the TB.

Figure 4.2.2 depicts the partial organization of a node.

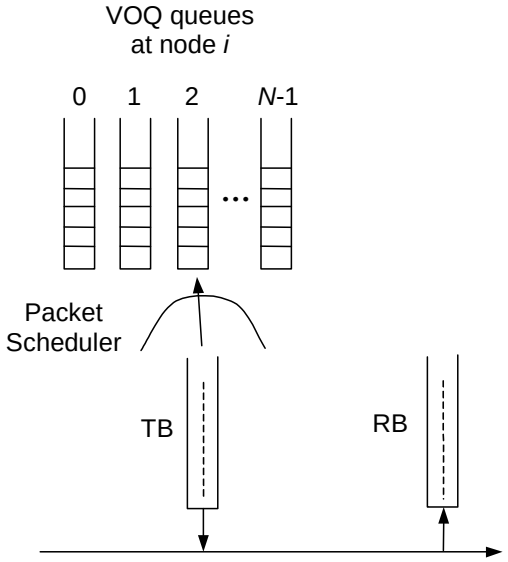


Figure 4.2.2 - Node organization in PAT

Whenever allocated a packet, the TB is bound to both that packet and the destination node of that packet. Hence, a new packet can be inserted into the TB only after the TB is released.

Whenever allocated a packet, the RB is bound to both that packet and the source node of that packet. Hence, a new packet can be inserted into the RB only after the RB is released.

As the upper layer protocol sends a packet down to the MAC layer, the latter frames the packet and stuffs escape signals into the frame to mask BOF sequences, EOF sequences, or both, if any -data stuffing is necessary at this stage to make sure that packets fit in the slots. Eventually the MAC layer inserts the resulting frame into the proper queue.

Frames

A packet frame includes the packet itself, the size of the packet, and the type of the upper layer protocol that generated the packet. The destination node needs such information to demultiplex packets and send them to the corresponding upper layer protocol.

Figure 4.2.3 depicts the packet frame layout, which contains the following fields:

- Protocol type (PType): 16-bit field that stores the upper layer protocol type;
- Payload length information (PLI): 16-bit field that stores the size (in octets) of the payload data;
- Payload data: variable size field that contains the payload data itself, and it can be as big as 64KB.

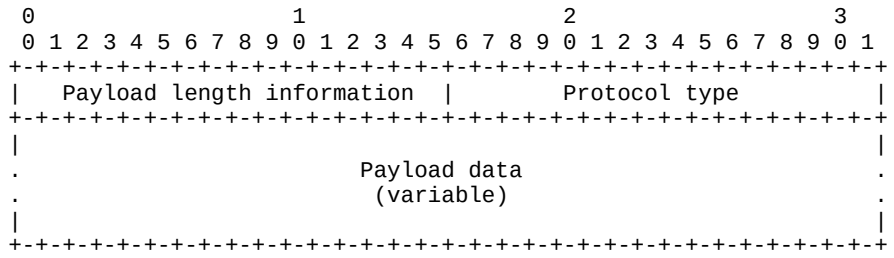


Figure 4.2.3 - Packet frame layout

To multiplex packets PAT uses the frame layout shown in Figure 4.2.4. The layout includes the following fields:

- Payload [i]: variable size field that contains packet frame *i*;
- PVL (Payload vector length): 8-bit field that describes the number of packets multiplexed in the slot.

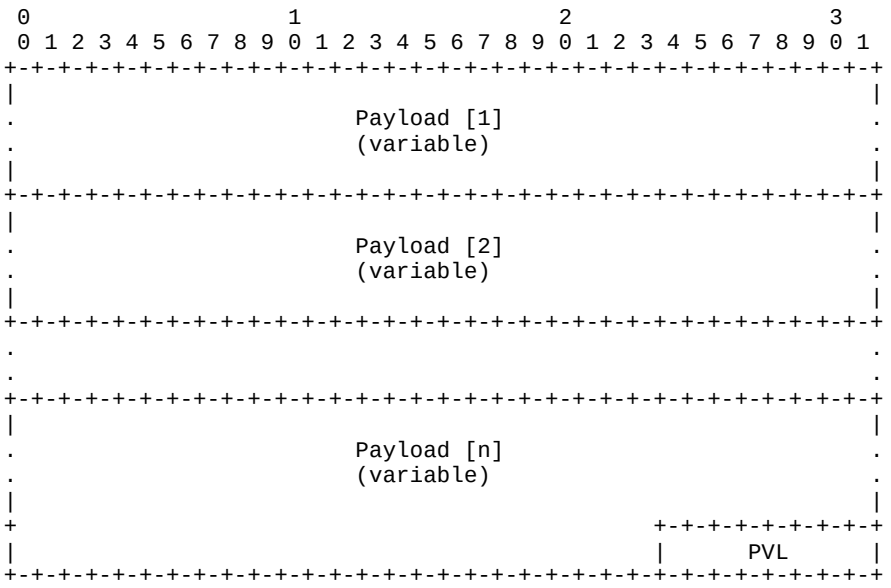


Figure 4.2.4 - Packet multiplexer frame layout

Regardless of the number of packets that a node can multiplex into a payload slot, there is always one slot header per payload slot.

The slot header contains the following fields:

- S: tells whether the corresponding payload slot is either EMPTY or BUSY;
- Destination Address (DA): contains the address of the destination of the slot if S is BUSY. If S is EMPTY then DA is set to NULL -nodes process DA only in the former;
- Source Address (SA): contains the address of the source node of the slot if S is BUSY. If S is EMPTY then SA is set to NULL -nodes process SA only in the former.

Behaviour

Upon arrival of a slot header, a node reads S to determine the status of the slot. If S is EMPTY then the node runs the transmission algorithm. If S is BUSY then the node matches its own address with the DA contained in the slot header. If the match fails then the node lets the slot go. If the match succeeds then the node drops the corresponding payload slot and runs the reception algorithm.

Note that a node always attempts reception first because that allows for that node to reuse the slots it releases, hence improving the network performance.

Transmission

If the node is idle then it lets the slot go unaltered. If the node has a traffic backlog then it triggers packet scheduling in an attempt to transmit. The packet scheduler selects a queue and moves the HOL packet from that queue into the TB.

If the packet scheduler fails then the node lets the slot go unaltered. If the packet scheduler succeeds then the node updates S to BUSY and assigns its own address and the destination address of the packet to SA and DA.

Then the node transmits the BOF sequence and, immediately after, the packet from the TB. While transmitting the packet, the node signals the packet scheduler to select the (new) HOL packet from the chosen queue.

To keep track of the space remaining in the slot for payload data, the node subtracts the size of each payload frame from the space available in the slot for payload frames. The size of a payload frame equals the value of PLI plus the size of the PLI field plus the size of the PType field; the space available in the slot for payload frames is the slot size minus the size of the PVL field.

Since PAT does not segment packets, the scheduler selects the HOL packet only if that packet fits entirely in the space remaining in the slot. If the packet scheduler succeeds then it inserts another packet into the TB as soon as the TB becomes empty. If the packet scheduler fails, then the node inserts the PVL field - a node increments PVL by 1 for each packet it transmits. Eventually the node transmits an EOF sequence.

Transmission can occur only from the beginning of a slot and continuously such that there is no gap between any pair of adjacent packets. Therefore, an idle node forwards an incoming free slot even if a burst of small packets arrives before that slot leaves the node entirely. The bandwidth usage efficiency gains that transmitting at any instant of time within a time slot might bring do not compensate for the complexity and the protocol-processing overhead that it introduces.

Reception

The node synchronizes with the BOF sequence and starts receiving the payload frame into the RB. While receiving, that node also removes any stuffed escape signals. Once the node synchronizes with the EOF sequence it starts to demultiplex the packet frames.

To determine how many packet frames the multiplexer frame contains the node uses PVL. Using the PLI field in each packet frame, the node extracts each packet frame from the multiplexer frame and forwards the embedded payload to the upper layer protocol, as indicated in PType.

To release the payload slot just received the node updates the slot header as follows: S to EMPTY and both DA and SA to NULL.

4.2.2 SPT+SC

SPT+SC [Salva2003a] guarantees that, under operational condition, packets are either transmitted entirely and continuously or not transmitted at all. To do so SPT+SC uses a technique called slot concatenation.

Figure 4.2.5 illustrates the functioning of SPT+SC. In the example a node has a backlog of five packets, labelled *a*, *b*, *c*, *d*, and *e* according to the order of arrival of those packets. Packet scheduling uses the longest delay as selection criterion; the destination of the packets is irrelevant.

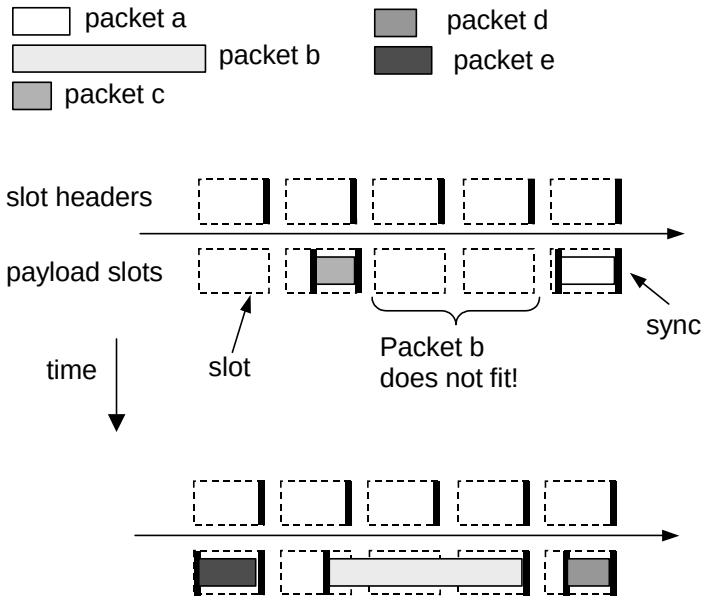


Figure 4.2.5 - Illustration of the functioning of SPT+SC

Slot concatenation is not a new idea. Pasch and Niemegeers [Pasch1991] describes the universal channel network (UCN) and a cyclic reservation access protocol that uses slot concatenation. Nevertheless, the protocol uses source removal, which limits the overall performance considerably.

SPT+SC builds on the distributed cycle protocol (DCP) [Dobos1993, Dobos1995]. DCP uses token removal rather than the traditional source removal or destination removal. Each node holds the token for exactly *THT* slots regardless of its traffic backlog status. A node that holds the token terminates the ring, that is, clears all the incoming slots indiscriminately; consequently, it has guaranteed access to the ring using such slots.

Right after releasing the token, each node starts the token rotation timer (*TRTr*). This timer plus the knowledge of the *THT* of each node plus the knowledge of the network topology allow for the determination of the exact location of the token at any time. Consequently, a node can always determine to which destination nodes it can transmit at any time.

SPT+SC enhances DCP in the following ways:

- Destination removal in addition to token removal;
- A node that holds the token is also responsible for the concatenation of slots;
- Multiple tokens to deal with multiple channels, whereas each token is associated with a distinct channel.

SPT+SC separates any pair of adjacent tokens (that is, subsequent nodes on adjacent channels) by intertoken distance (*ITD*) slots, where $ITD = S / C$[1], S denotes the number of slots in the ring, and *C* denotes the number of channels in the ring. Therefore, the smaller the ratio *N* / *C*, the more frequent a node possesses a token.

Figure 4.2.6 illustrates the distribution of tokens in a ring with eight nodes and four channels.

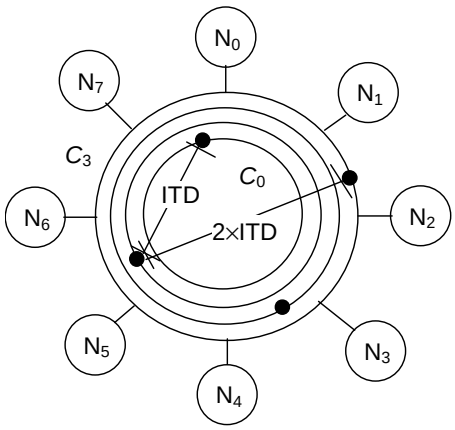


Figure 4.2.6 - Illustration of the distribution of multiple tokens across the ring

A node that receives the token performs slots concatenation over the incoming *THT* slots on the channel corresponding to that token, hence generating slot train(s) on that channel.

Organization

To avoid HOL blocking, which may happen as shall be seen later, SPT+SC uses VOQ and a-posteriori packet scheduling. Each node maintains *N*-1 queues, one per possible destination node, a single TB capable of storing one MTU-sized packet, and a single RB also capable of storing one MTU-sized packet. The packet scheduler serves the packets queue according to the FCFS discipline, and moves the HOL packet from the queue to the TB.

Figure 4.2.7 depicts the partial organization of a node.

[1] For the sake of simplicity, let S be an integer multiple of C.

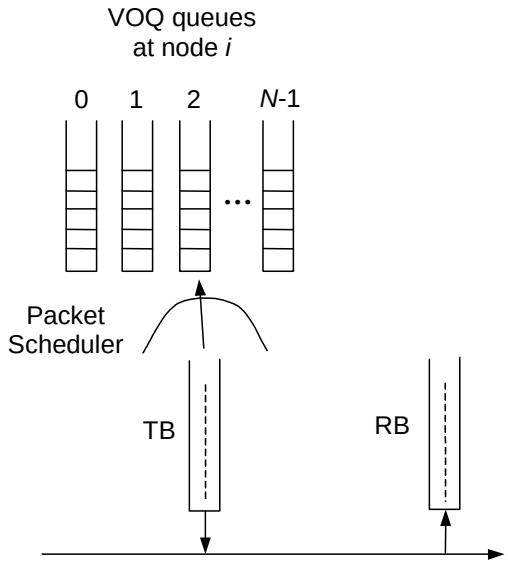


Figure 4.2.7 - Node organization in SPT+SC

Whenever allocated a packet, the TB is bound to both that packet and the destination node of that packet. Hence, a new packet can be inserted into the TB only after the TB is released.

Whenever allocated a packet, the RB is bound to both that packet and the source node of that packet. Hence, a new packet can be inserted into the RB only after the RB is released.

As the upper layer protocol sends a packet to the MAC layer, the latter frames the packet, stuffs escape signals into the frame to mask BOF sequences, EOF sequences, or both, if any; data stuffing is necessary at this stage to make sure that packets fit in the trains. Eventually the MAC layer inserts the resulting frame into the proper queue.

Frames

A packet frame includes the packet itself, the size of the packet, and the type of the upper layer protocol that generated the packet. The destination node needs such information to extract packets and send them to the corresponding upper layer protocol.

Figure 4.2.8 depicts the payload frame layout, which includes the following fields:

- PType: 16-bit field that describes the upper layer protocol that generated the payload data;
- PLI: 16-bit field that describes the size (in octets) of the payload data;
- Payload data: variable size field that carries payload data, and it can be as big as 64KB.

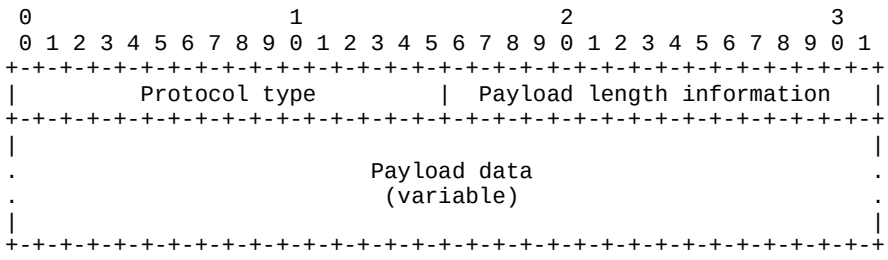


Figure 4.2.8 - Payload frame layout

The slot header contains the following fields:

- S: tells whether the corresponding payload slot is either EMPTY or BUSY;
- B: tells whether the slot either is the head of the train it belongs to, or is not. The slot is the head if B is TRUE, and the slot is not the head if B is FALSE;
- T: tells whether the token is either present or absent. The token is present if T is TRUE and the slot is absent if T is FALSE;
- Train length information (TLI): tells how many slots compose the train if B is TRUE. If B is FALSE then TLI is set to NULL. Nodes process TLI only in the former;
- Destination Address (DA): contains the address of the destination of the slot if S is BUSY. If S is EMPTY then DA is set to NULL. Nodes process DA only in the former;
- Source Address (SA): contains the address of the source node of the slot if S is BUSY. If S is EMPTY then SA is set to NULL. Nodes process SA only in the former.

Slot trains generation

Let n_i be the node that just received the token, c be the channel corresponding to that token, and R be the remaining THT corresponding to that token; n_i sets R to THT and, for each slot on channel c that passes by, it decrements R by 1. While possessing that token, n_i generates slot trains of size $T = min(MTU, R)$ on c, where MTU is the train size needed to carry one MTU-sized packet.

To generate a train of size T the node updates the header of each of the slots composing the train as follows:

- Head slot: set B to TRUE and TLI to T;
- Subsequent slots: set B to FALSE and TLI to null.

Figure 4.2.9 depicts the slot train generation process.

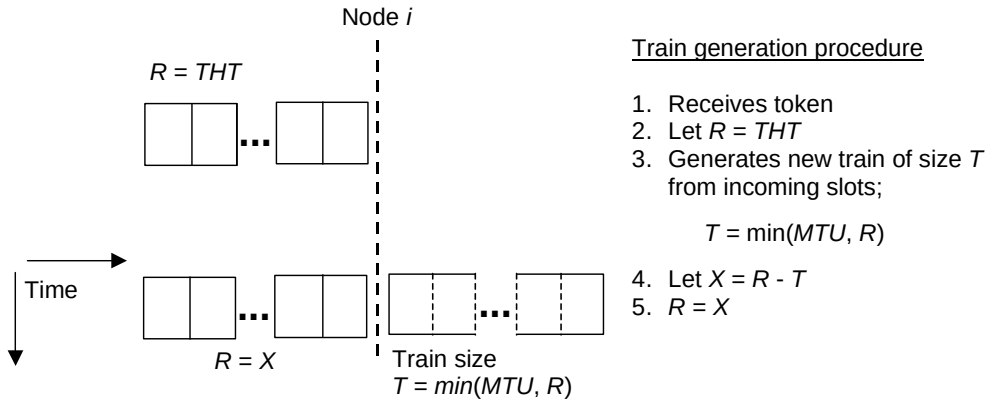


Figure 4.2.9 – Illustration of the slot train generation procedure

Note that slot train generation and transmission are separate functions. A node that is generating trains can also transmit using those trains though.

Behaviour

Upon arrival of a slot header, a node reads S to determine the status of the slot. If S is EMPTY then the node runs the transmission algorithm. If S is BUSY then the node matches its own address with the DA contained in the slot header. If the match fails then the node lets the slot go. If the match succeeds then the node drops the corresponding payload slot and runs the reception algorithm.

Note that a node always attempts reception first because that allows for that node to reuse the slots it releases, hence improving the network performance.

Transmission

If the node is idle then it lets the train go unaltered. If the node has a traffic backlog then it signals the packet scheduler to select a packet for transmission. A selection has to satisfy two conditions to be successful. First, the node holding the token is not along the path towards the destination node of the selected packet. Formally and assuming that nodes are assigned the same *THT*, a node is allowed to transmit a packet upon arrival of an empty train if

$$TRTr \geq (d - 1) \times THT \qquad (\mathbf{1})$$

where d denotes the distance in number of hops from the node attempting transmission to the packet's destination node.

If nodes are indexed in sequential order according to their topological position on the ring, then d is given $| j - i + N |_N$, where j denotes the index of the destination node, i denotes the index of the source node, and N denotes the number of nodes in the network; $| . |_N$ denotes the modulo N operation.

Second, the selected packet fits entirely in the train. Let T be the size of an incoming train at node i, and P be the size of a HOL packet at node i. Node i can transmit that packet only if

$$P \leq T \qquad (2)$$

If the packet selection meets both transmission constraints then the packet scheduler inserts that packet into the TB and starts transmitting.

As far as the transmission of payload data is concerned, the node transmits a BOF sequence and, immediately after, the payload. The node ends the transmission by transmitting an EOF sequence.

In parallel to the transmission of the payload data, the node updates the headers of the train allocated to the packet as follows: S to BUSY, the node address to SA, and the destination node address of the packet to DA. If the slot is the head then the node updates B to TRUE and TLI to the size of the train, that is, to the number of slots necessary to carry the packet. If the slot is not the head then the node sets B to FALSE and TLI to NULL.

Once transmission of the packet ends, if $T - P$ is greater than 0 then the node sets the next slot as head and sets the train as being of size $T - P$. In other words, the node splits the incoming train into two new trains using the slot train generation procedure explained previously. The first train carries the packet and, therefore, has a size P. The second train contains the remaining slots from the received train and, therefore, has a size of $T – P$; if $T – P$ is zero then the second train is not created.

Figure 4.2.10 illustrates how a node proceeds upon transmitting.

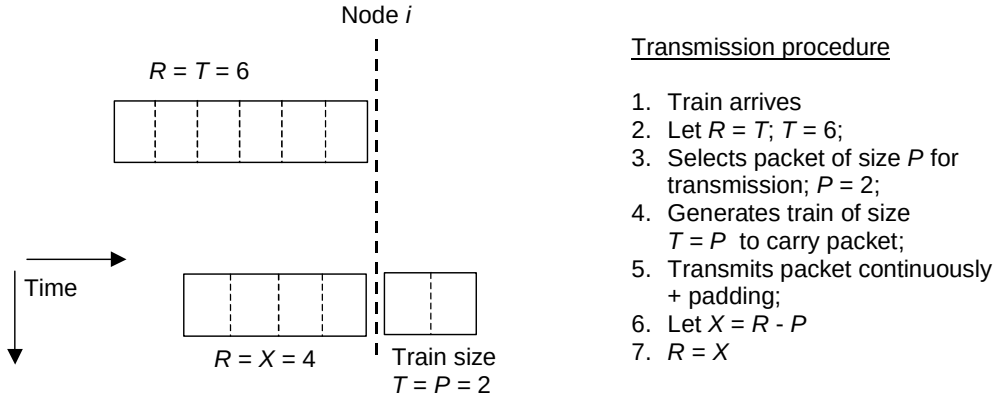


Figure 4.2.10 – Illustration of the transmission procedure in SPT+SC

SPT+SC allows transmission only from the beginning of a train. If a node cannot transmit at the beginning of a train for whatever reason (e.g., node is idle), then that node skips the entire train, even though transmission might become possible before the entire train is forwarded.

Such a transmission constraint ensures that nodes indeed perceive the length of a train as indicated in the head slot header of that train. If a node forwards the head slot header of an empty train and then transmits in the middle of that train, a collision might occur at a downstream node that has allocated that train entirely, based on the indication in the received head slot header.

Reception
A node matches its own address with the DA contained in the received header. If the match fails then the node lets the slot go. If the match succeeds then the node drops the corresponding payload slot.

The node synchronizes with the BOF sequence of the dropped frame to obtain the payload contained in that frame. While receiving, that node also removes any stuffed escape signals. Once the node synchronizes with the EOF sequence it starts to process the frame.

Using the value of PLI the node extracts the packet from the frame and sends the packet to upper layer protocol, as indicated in PType. Eventually the node empties the RB.

To release the train the node updates each header of the train as follows: S to EMPTY and both DA and SA to NULL; the value of the remaining fields remains unaltered.

4.3 Contention protocols

This section describes two contention access control protocols: slotted packet transmission with retransmission (SPT+R) and slotted packet transmission with pause (SPT+P).

4.3.1 SPT+R

SPT+R [Salva2003b] guarantees that packets reach their destination nodes as single units, using as many consecutive slots as necessary. Whenever a backlogged node detects an incoming empty slot that node selects a packet, and starts transmitting that packet. If during the transmission that node detects an incoming busy slot then that node stops transmitting to avoid collision. As soon as that node detects an incoming empty slot, that node restarts transmitting that same packet from the beginning.

Figure 4.3.1 illustrates how SPT+R works. In the example a node has a backlog of four packets, labelled *a*, *b*, *c*, and *d* according to the order of arrival of those packets. Packet scheduling uses the longest delay as selection criterion; the destination of the packets is irrelevant.

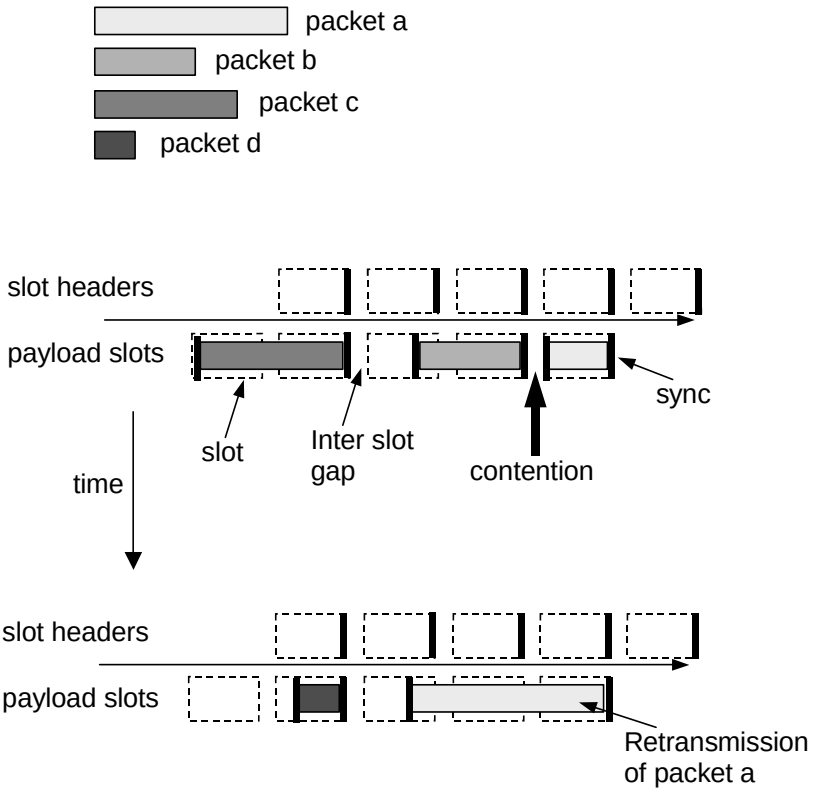


Figure 4.3.1 - Illustration of the functioning of SPT+R

Note that to cross the boundaries between slots SPT+R requires that slot headers be sent ahead of their corresponding payload slots, such that a node can detect the status of the next slot before reaching the end of the current payload slot.

Organization

Each node maintains a single packets queue, a single TB capable of storing one MTU-sized packet, and a single RB also capable of storing one MTU-sized packet. As the upper layer protocol sends a packet to the MAC layer, the latter frames the packet and inserts the frame into the queue. The packet scheduler serves the packets queue according to the FCFS discipline, and moves the HOL packet from the queue to the TB.

Figure 4.3.2 depicts the partial organization of a node.

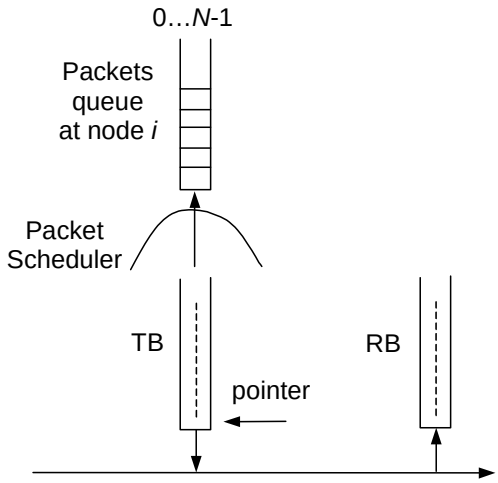


Figure 4.3.2 - Node organization in SPT+R

Whenever allocated a packet, the TB is bound to both that packet and the destination node of that packet. Hence, a new packet can be inserted into the TB only after the TB is released.

Whenever allocated a packet, the RB is bound to both that packet and the source node of that packet. Hence, a new packet can be inserted into the RB only after the RB is released.

Frames

The payload frame layout is shown in Figure 4.3.3, and it contains the following fields:

- PType: 16-bit field that identifies the upper layer protocol that generated the payload data;
- PLI: 16-bit field that defines the size of the payload data;
- Payload data: variable size field that carries payload data, and it can be as big as 64KB.

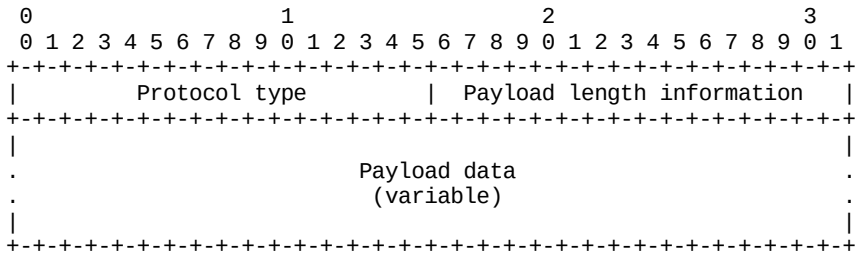


Figure 4.3.3 - Payload frame layout

The slot header includes the following fields:

- S: tells whether the corresponding payload slot is either EMPTY or BUSY;

- B: tells whether the slot either is the head of the train it belongs to, or is not. The slot is the head if B is TRUE, and the slot is not the head if B is FALSE;
- Destination Address (DA): contains the address of the destination of the slot if S is BUSY. If S is EMPTY then DA is set to NULL. Nodes process DA only in the former;
- Source Address (SA): contains the address of the source node of the slot if S is BUSY. If S is EMPTY then SA is set to NULL. Nodes process SA only in the former.

Behaviour

Upon arrival of a slot header, a node reads S to determine the status of the slot. If S is EMPTY then the node runs the transmission algorithm. If S is BUSY then the node matches its own address with the DA contained in the slot header. If the match fails then the node lets the slot go. If the match succeeds then the node drops the corresponding payload slot and runs the reception algorithm.

Note that a node always attempts reception first because that allows for that node to reuse the slots it releases, hence improving the network performance.

Transmission

A node always checks first if there is any pending transmission. If there is, then the node uses that slot to resolve that transmission. If there is not and the node is idle, then that node lets the slot go unaltered. If there is not, but the node has a traffic backlog then that node triggers packet scheduling in an attempt to transmit a new packet.

Assuming that there is no pending transmission and that the node has a traffic backlog, the packet scheduler selects a packet, inserts that packet into the TB, and starts transmitting.

As far as the transmission of payload data is concerned, the node transmits a BOF sequence and, immediately after, the payload. As the transmission progresses, the node inserts escape signals into any field that contains BOF sequences, EOF sequences, or both.

In parallel to the transmission of the payload data the node updates the headers of the train allocated to the packet as follows: S to BUSY, the node address to SA, and the destination node address of the packet to DA. If the slot is the head then the node updates B to TRUE. If the slot is not the head then the node sets B to FALSE.

If the node transmits the entire packet successfully then it transmits an EOF sequence. If, however, the node detects an incoming busy slot before the transmission is complete, then that node stops transmitting and inserts an EOF sequence immediately.

As the next empty slot arrives, that node attempts to retransmit that packet, and the whole transmission process restarts.

Eventually that node transmits the entire packet and releases the TB.

Reception

A node matches its own address with the DA contained in the received slot header. If the match fails then the node let the slot go. If the match succeeds then the node drops the corresponding payload slot.

The node synchronizes with the BOF sequence and starts receiving the payload data. While receiving, the node removes stuffed escape signals, and also counts the received data -the counting does not consider escape signals.

Upon detection of an EOF sequence the node processes the received frame. If the value of the counter matches the size of PType plus the size of PLI plus the value of PLI then the node extracts the packet from the frame.

To determine to which upper layer protocol to send the packet the node uses PType; eventually the node empties the RB.

To release the train just received, the node updates headers of the train as follows: S to EMPTY and both DA and SA to NULL. Field B remains unaltered.

4.3.2 SPT+P

SPT+P [Salva2002a] tries its best to transmit packets continuously, using as many consecutive slots as necessary. Whenever a backlogged node detects an incoming empty slot that node selects a packet, and starts transmitting that packet. If during the transmission that node detects an incoming busy slot then that node stops transmitting to avoid collision. As soon as that node detects an incoming empty slot, that node continues to transmit from the point where it stopped.

Figure 4.3.4 illustrates how SPT+P works. In the example a node has a backlog of three packets, labelled *a*, *b*, and *c* according to the order of arrival of those packets. Packet scheduling uses the longest delay as selection criterion; the destination of the packets is irrelevant.

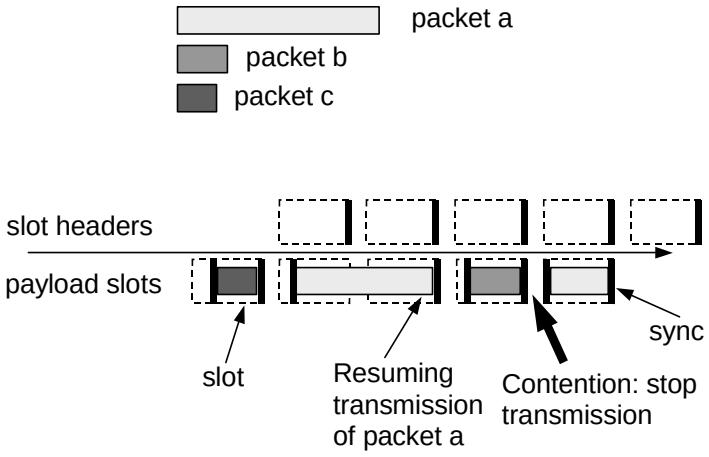


Figure 4.3.4 - Illustration of the functioning of SPT+P

Note that to cross the boundaries between slots the SPT+P requires that slot headers be sent ahead of their corresponding payload slots, such that a node can

detect the status of the next slot before reaching the end of the current payload slot.

Organization

Each node maintains a single packets queue and a single TB capable of storing one MTU-sized packet. As the upper layer protocol sends a packet to the MAC layer, the latter frames the packet and inserts the frame into the queue. The packets scheduler serves the packets queue according to the FCFS service discipline and moves the HOL packet from the queue to the TB.

Figure 4.3.5 depicts the organization of a node.

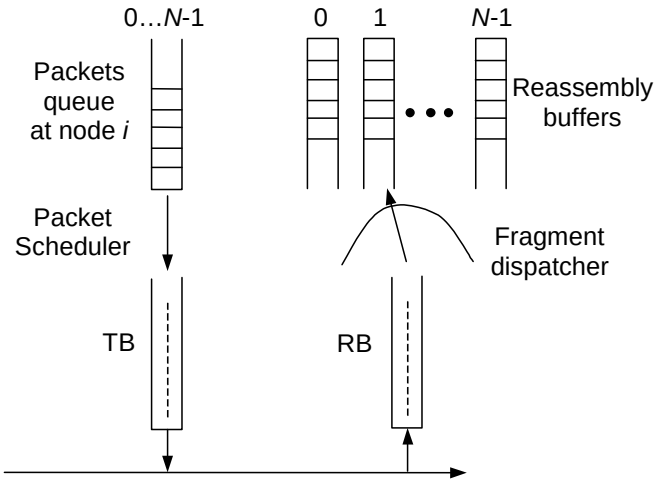


Figure 4.3.5 - Node organization in SPT+P

Whenever allocated a packet, the TB is bound to both that packet and the destination node of that packet. Hence, a new packet can be inserted into the TB only after the TB is released.

Each node maintains also a single RB capable of storing one MTU-sized packet and re-assembly buffers. In the worst-case, one re-assembly buffer per each possible source node is required. Note that various re-assembly buffer allocation strategies are possible, but they are out of the scope of this work.

Whenever allocated a packet, the RB is bound to both that packet and the source node of that packet. Hence, a new packet can be inserted into the RB only after the RB is released.

Whenever allocated a packet, a re-assembly buffer is bound to both that packet and the source node of that packet. Hence, a new packet can be inserted into that re-assembly buffer only after that buffer is released.

Frames

The payload frame layout is shown in Figure 4.3.6, and it contains the following fields:

- PType: 16-bit field that identifies the upper layer protocol that generated the payload data;
- PLI: 16-bit field that defines the size of the payload data;
- Payload data: variable size field that carries payload data, and it can be as big as 64KB.

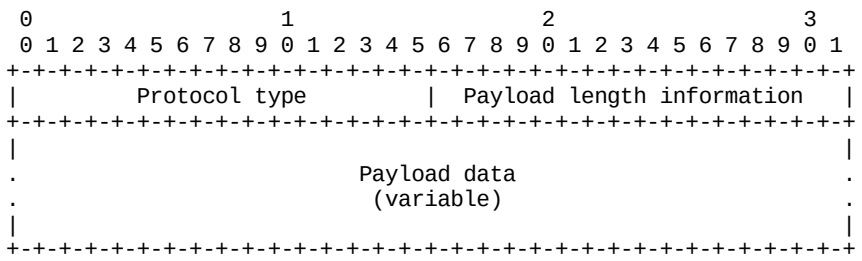


Figure 4.3.6 - Payload frame layout

The slot header includes the following fields:

- S: tells whether the corresponding payload slot is either EMPTY or BUSY;
- B: tells whether the slot either is the head of the train it belongs to, or is not. The slot is the head if B is TRUE, and the slot is not the head if B is FALSE;
- E: tells whether the slot either is the tail of the train, or is not;
- Destination Address (DA): contains the address of the destination of the slot if S is BUSY. If S is EMPTY then DA is set to NULL -nodes process DA only in the former;
- Source Address (SA): contains the address of the source node of the slot if S is BUSY. If S is EMPTY then SA is set to NULL -nodes process SA only in the former.

Behaviour

Upon arrival of a slot header, a node reads S to determine the status of the slot. If S is EMPTY then the node runs the transmission algorithm. If S is BUSY then the node matches its own address with the DA contained in the slot header. If the match fails then the node lets the slot go. If the match succeeds then the node drops the corresponding payload slot and runs the reception algorithm.

Note that a node always attempts reception first because that allows for that node to reuse the slots it releases, hence improving the network performance.

Transmission algorithm

A node always checks first if there is any pending transmission. If there is, then the node uses that slot to resolve that transmission. If there is not and the node is idle, then that node lets the slot go unaltered. If there is not, but the node has a traffic backlog then that node triggers packet scheduling in an attempt to transmit a new packet.

Assuming that there is no pending transmission and the node has a traffic backlog, the packet scheduler selects a packet, inserts that packet into the TB, and starts transmitting.

As far as the transmission of payload data is concerned, the node transmits a BOF sequence and, immediately after, the payload. As the transmission progresses, the node inserts escape signals into any field that contains BOF sequences, EOF sequences, or both.

In parallel to the transmission of the payload data the node updates the headers of the train allocated to the packet as follows: S to BUSY, the node address to SA, and the destination node address of the packet to DA. In addition, the node sets both B in the head slot header to TRUE and, if the slot carries the end of the packet, E in the tail slot header to TRUE; for all the other slots the node sets both B and E to FALSE.

If the node transmits the entire packet successfully then it transmits an EOF sequence. If, however, the node detects an incoming busy slot before the transmission is complete, then that node stops transmitting and inserts an EOF sequence immediately.

As the next empty slot arrives, that node attempts to transmit that packet from the point where transmission was interrupted.

Eventually that node transmits the entire packet and releases the TB.

Reception

A node matches its own address with the DA contained in the received slot header. If the match fails then the node let the slot go, but if the match succeeds then the node drops the corresponding payload slot.

The node synchronizes with the BOF sequence and starts receiving the payload data. While receiving the node also removes stuffed escape signals. Upon detection of an EOF sequence the node moves the received data to a re-assembly buffer and empties the RB.

If there is a re-assembly buffer already allocated that SA then the node inserts the packet into that buffer. If there is no re-assembly buffer allocated to that SA, then the node allocates one, inserts the packet into that buffer, and also starts an expiration timer for that buffer. At the expiration time the node releases the buffer indiscriminately.

The node triggers packet re-assembly upon arrival of a tail slot whose DA matches its own address and whose SA matches the SA of an allocated re-assembly buffer. If these conditions hold, then the node uses PLI to extract the packet and PType to determine to which upper layer protocol to send the packet.

To release the train the node updates the headers of the train as follows: S to FALSE and both DA and SA to NULL. Both B and E remain unaltered.

4.4 Channel multiplicity

The existence of multiple channels raises the concern of contention across channels at destination. From now on the term vertical contention is used to

describe the former and the term horizontal contention is used to describe contention across time slots on the same channel.

Vertical contention occurs whenever the number of trains across channels and overlapping in time destined to a certain node exceeds the number of Rx units available at that node. Vertical contention can occur in two situations:

- Destination node is equipped with one TRx unit or more, and source node cannot monitor all the channels to determine whether its own transmission will result in a vertical contention;
- Destination node is equipped with one FRx unit or more, and source node cannot determine the receivers from a multicast address, as usually; consequently, the source node cannot determine whether its own transmission will result in a vertical contention.

Two solutions to cope with vertical contention are as follows. In the first solution, a source node that can monitor all the channels schedules a packet for transmission only if the destination node of that packet has sufficient Rx units to receive the train carrying that packet plus the detected transit trains going to the same destination.

If while transmitting on the allocated train the source node detects the arrival of a transit train that is going to generate a vertical contention at the destination, then the source node stops transmitting; note that this type of contention does not occur in PAT since trains are all 1-slot long in this protocol.

If a destination node detects the simultaneous arrival of more parallel train heads than it can receive, then that node applies a channel selection algorithm to choose which trains to drop and which trains to let go.

If while receiving a train a destination node detects a vertical contention then that node aborts the ongoing reception, releases the RB and the re-assembly buffers, if applicable, and starts receiving the train that arrived last. Note that if a destination node is receiving two or more trains and it detects a vertical contention then that node has to select which receptions to abort first. Nevertheless, unless a transmission interrupted as a result of vertical contention is marked as such by its source, a destination node might abort the reception of a train whose transmission has not been interrupted.

In the second solution, a source node that can monitor all the channels schedules a packet for transmission only if the destination node of that packet has sufficient TRx units to receive the train carrying that packet plus the detected transit trains going to the same destination. Nevertheless, while transmitting that source node will continue to do so even if it detects a vertical contention.

If a destination node detects the simultaneous arrival of more train heads than it can receive, then that node applies a channel selection algorithm to choose which trains to drop and which trains to let go. If while receiving a train a destination node detects a vertical contention involving another train, then that node continues to receive the train that arrived first and lets the train that arrived last go unaltered.

In both solutions a train forwarded by its destination is eventually removed from the ring.

The main difference between both solutions is that in the first solution nodes respond to vertical contention in the same way as they respond to horizontal contention, although at the expense of higher complexity. The second solution considers contention from two different perspectives, but it is simpler. Nevertheless, it might result in higher packet loss because source nodes continue to transmit in the event of contention.

4.5 Error handling

As explained in Chapter 2, packet live-lock might occur depending on the transceiver configuration. To prevent that, all the protocols but SPT+SC should use the time-to-live (TTL) concept -packet live-lock does not occur in SPT+SC because a node that holds a token eventually removes an orphan packet from the ring. A TTL field counts how many hops a payload slot can still travel. Every node decrements TTL by 1 before forwarding the corresponding payload slot to the next node. If the resulting TTL is greater than 1 then the processing node forwards the payload slot to the next hop. If the resulting TTL is smaller than 1 then the processing node discards the corresponding payload slot.

Also explained in Chapter 2, erroneous slot headers may lead to packet misdelivery and even packet live-lock regardless of the presence of TTL. To detect erroneous slot headers the protocols add a frame check sequence (FCS) to each generated slot header. Typically, a FCS contains the cyclic redundancy check (CRC) polynomial that is calculated over the fields to be protected. A node processing a slot header always calculates its own FCS and matches it with the FCS contained in the header. If the match fails then the header is erroneous and proper measures should be taken.

In any of the protocols, a node that either detects an erroneous slot header or has to forward a slot that exceeds the minimum TTL should discard the corresponding payload data. Nevertheless, a node may be incapable of dropping on the channel that carries the payload. Consequently, if that node marks a slot header as empty without actually releasing the corresponding payload data, a collision will occur when a node attempts to transmit on that slot. Therefore, a node applies slot discard only if that node can drop that slot. If that is not possible then the node lets the slot go; eventually that slot reaches a node that can drop it.

A source node in SPT+SC may allocate a train to transmit a packet, but because of protection mechanisms the actual size of that train might not correspond to the size indicated in the PLI field of the head slot of that train. The source node copes with such an error by aborting the transmission.

Note that such an error may propagate through further downstream nodes, and only the node that holds the token on the corresponding channel can correct the error.

As a consequence of the execution of protection mechanisms or slot discard, destination nodes can receive packet fragments out-of-order or even packets missing fragments. Destination nodes react to such occurrences as follows:

- Incomplete packet arrival: the absence of fragments likely results in packets whose size fails to match the PLI field. Because both the end of the packet being received and the beginning of a new packet can be lost, it might happen that fragments from different packets but from the same source node succeed the PLI match. Nevertheless, the resulting packet is likely to fail to pass the FCS test. A packet that fails either test should be discarded, and the allocated buffers should be released;
- Out-of-order arrival: destination nodes cannot detect out-of-order arrivals until packets undergo the FCS test. A packet that fails the FCS test should be discarded, and the allocated buffers should be released;
- Erroneous packet: a complete in-order packet can contain errors caused by transmission impairments or, most likely, by electronic handling. Such errors can be detected with great likelihood by the FCS test. Again, packets that fail to pass the FCS test should be discarded, and the allocated buffers should be released;
- Packet arrival overlapping in SPT+P: given a certain channel, a destination node might receive a new packet from a node before it received the previous packet transmitted by that same node entirely. If that is true, then the destination node discards the received fragments, releases the re-assembly buffers allocated to those fragments, and starts receiving the new packet;
- Packet arrival overlapping in SPT+R and SPT+SC: given a certain channel, a destination node might either receive a train whose SA fails to match the SA of the packet being received, or receive a train whose SA matches the SA of the packet being received, but it carries another packet. A destination node reacts to this situation by discarding the packet fragment(s) already received, releasing the re-assembly buffers allocated to those fragments, and attempting to receive the new packet as usual;
- Orphan packet arrival in SPT+P, SPT+R, and SPT+SC: upon detection of a train that does not carry the beginning of a packet, if the destination node fails to match the SA of that train with the SA of any allocated re-assembly buffer, then that node should discard the train.

4.6 Discussion

This chapter introduced four access control protocols for MOPS rings based on the conceptual node architecture shown in Figure 1.1.5. Thanks to both the definition of an open system model upon which the protocols rely and the design concerned with openness, the protocol designs are not limited to particular transceiver configurations. What is more, the protocol designs can even support heterogeneous transceiver configurations. Because of the assumption about the existence of an interface between the MAC layer and the PHY layer, the protocol

designs do not assume any particular implementation of the signalling mechanism at the PHY layer.

Each protocol has advantages and disadvantages compared with one another. Amongst parameters of importance, this section discusses complexity, scalability, robustness, and efficiency. Performance is also important, but its discussion is left to Chapter 6.

SPT+SC is the most complex protocol. Its algorithm demands special care to maintain consistency and high processing overhead to schedule packets. SPT+P is simple, but its dependency on SAR operations and re-assembly buffers results in complexity at the system level.

The simplest protocol is SPT+R. SPT+R is similar to SPT+P, with the advantage that the former uses a retransmission contention resolution that is simple and dismisses SAR operations.

PAT is also simple, but the packet scheduling algorithm and the multiplexing and demultiplexing of packets at edge nodes introduce some complexity that puts the protocol in between SPT+R and SPT+P in complexity.

Scalability can be seen from different perspectives. As far as number of nodes is concerned, PAT is the less scalable. Given a certain ring length, PAT generates lesser slots than the other protocols. And as shown in [Zafir1988], destination removal slotted ring networks achieve better performances as the ratio between number of slots and number of nodes increases.

As far as (payload data) channel bit rate is concerned though, the most scalable is PAT. That is because of the use of large slots and the consequent reduced number of slot headers to process per second.

Since it transmits packets as atomic units, PAT does not require special care to cope with errors. It does require some care to cope with network failures, but not as much as the other protocols. Therefore, PAT is the most robust protocol.

SPT+SC suffers the most from failures since erroneous slot headers may lead to collision, and collision is not supposed to occur in the protocol.

As far as efficiency is concerned, figures may vary depending on the traffic pattern; therefore it is difficult to elaborate on efficiency. Nevertheless, overall SPT+P is the most efficient protocol since it exploits the fragmentation of the medium capacity. That does not hold with SPT+R, whose performance suffers as the fragmentation of the medium capacity increases.

SPT+SC also suffers from the same problem as SPT+R, but to a lesser extent, in particular, as the number of tokens increases, because of the assurance that once transmitted packets will reach the destination completely and successfully.

Because of the size of the slots, PAT can be the least efficient protocol, in particular under asymmetric traffic distribution patterns. Some nodes might have only a few short packets to transmit every time a slot arrives, which results in slots with considerable fractions of empty space and, consequently, waste of capacity.

Table 4.6-1 highlights the main aspects of the protocols.

Table 4.6-1 – Main aspects of the protocols

	SPT+P	**PAT**	**SPT+R**	**SPT+SC**
Efficiency	High	Depends on the traffic condition	Depends mainly on the traffic workload	Depends on the traffic condition
Complexity	Low	Medium	Low	High
Robustness	Medium	High	High	Low
Scalability	High	Depends mainly on the ratio between the number of slots and the number of nodes	Low	Depends mainly on the number of nodes and the traffic condition
Cost	High	Low	Low	Low

For information on synchronisation, group communication, or addressing please read Appendixes B, C, or D.

Chapter 5
Access fairness protocols

This chapter deals with access fairness control protocols for general high-speed ring networks, and it is divided in two parts. The first part elaborates on the definition of fairness and describes some of the existing protocols proposed in the literature. The second part introduces two new protocols, one global and one local. It also discusses how to integrate such protocols with the access control protocols described in Chapter 4.

5.1 Fairness definition

An access fairness protocol comprises rules and behaviours to constrain access under certain conditions to ensure that geographically distributed nodes competing for the medium get a fair share of that medium.

Before discussing fairness it is important to understand what fairness means. A definition of fairness that is commonly used in the literature is that which says:

> A network is fair if all nodes receive the same share of network resources.

According to this definition and assuming a performance measure P, a network is perfectly fair if P_i is constant for $i = 0,\ldots, N\text{-}1$, where i denotes node index and N denotes the number of network nodes.

Such a definition assumes that all nodes have the same demands, which is often not true. A more accurate definition of fairness is given by the max-min optimisation [Berts1987], which says:

> A network is fair if each node gets the largest possible share that does not impact nodes with lower shares.

Note that the max-min optimisation aims at throughput fairness. Thus, quoting from [Berts1987], "a max-min fair throughput point $v = \{v^1,\ldots,v^n\}$ on a ring with n nodes has the following property: for $i = 1,\ldots, n$, v^i cannot be increased while maintaining feasibility without decreasing v^j for some j with $v^j \leq v^i$, where v^i is the throughput of node i and v^j is the throughput of node j." By feasibility the definition means that the sum of the throughputs is below the ring saturation point.

Regardless of which definition one follows, the perception of fairness depends on the point of view one looks from. As pointed out in [Dobos1992], what is

considered fair from one point of view may turn out to be unfair from another point of view or vice-versa. For instance, a network may be fair with respect to node throughput and, at the same time, be unfair with respect to packet delay variations.

The perception of fairness also depends on the period of time one looks at it. Consider a network in which, at time *T*, a specific node *A* transmits large volumes of data over the network. All the other nodes are idle. At time 2*T*, node *B* becomes backlogged with large volumes of data. If one adopts fairness on long time scales, then node A must remain silent until node *B* achieves the same usage of network resources. On the other hand, if one adopts fairness on short time scales, then node *A* must reduce its transmissions to the point where both node *A* and node *B* achieve the same usage of network resources.

It is difficult to achieve fairness on very short time scales in networks with high bandwidth-latency products, such as high-speed networks, large networks, or both.

It is also difficult to enforce strict access fairness in such networks; QoS mechanisms are more efficient in enforcing strict access fairness.

5.2 Existing protocols

Existing access protocols can be classified into two major classes: global and local. Global protocols consider the network as a single shared communication resource. Consequently, every node sees the same transmission constraint. Local protocols consider each link as a communication resource and the whole network as a multiplicity of communication resources. Therefore, only nodes competing for the same subset of communication resources see the same transmission constraint.

5.2.1 Global protocols

Examples of global access protocols include [Cidon1993], for the MetaRing architecture [Ofek1994], and credit window reset, for the asynchronous transfer mode ring (ATMR) [Imai1994]. Both protocols aim at throughput fairness.

5.2.1.1 MetaRing

Commonly used in MOPS rings, the fairness protocol developed in the MetaRing architecture can work with either slotted ring networks, or buffer insertion networks. While the protocol requires a few minor changes to work with either network, this section describes only the details pertaining to the use of the protocol with slotted ring networks.

The fairness protocol uses distributed cyclic credit. A so-called SAT (short for satisfied) signal, which rotates continuously around the ring, grants every node a pre-defined transmission quota (that is, credit) of Q(*k*, *l*) data units, where $k \geq l \geq 0$.

Let a fairness cycle be the interval between two consecutive visits of the SAT signal. A node that has transmitted less than *l* data units within a fairness cycle is

termed UNSATISFIED. A node that has transmitted *x* data units within a fairness cycle, where $l \le x < k$, is termed SATISFIED. A node that has transmitted *k* data units within a fairness cycle is termed EXHAUSTED.

According to the protocol, a node may transmit as long as it is either UNSATISFIED, or SATISFIED.

Figure 5.2.1 depicts the finite state machine (FSM) of protocol. For the sake of simplicity and clarity, Figure 5.2.1 shows only the events that lead to a state transition.

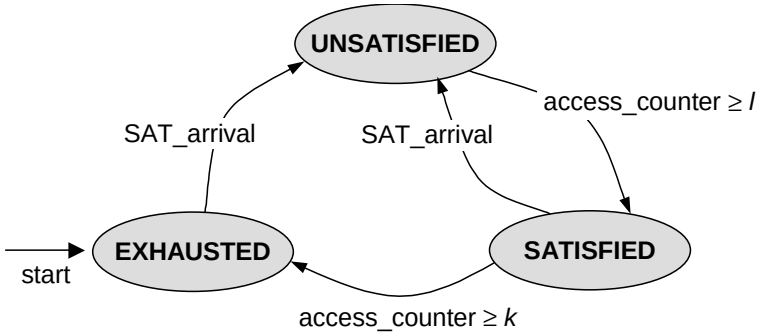


Figure 5.2.1 - FSM of the SAT protocol

Every node maintains a transmission counter that stores the number of data units transmitted in the current fairness cycle. A node increments such a counter by 1 every time it transmits one data unit.

A node that is either idle, SATISFIED, or EXHAUSTED and receives the SAT signal grants itself (that is, resets its transmission counter) a new transmission quota Q(*k*, *l*) and forwards the SAT signal to the next node immediately. An UNSATISFIED node that receives the SAT signal holds the signal until it becomes SATISFIED. The node then grants itself a new transmission quota Q(*k*, *l*) and forwards the SAT signal to the next node immediately.

The SAT signal can be implemented either by a single bit in the slot header, or by a specific control packet. Also, the SAT signal can rotate either in the data direction, or in the data's opposite direction.

Note that a few additional mechanisms exist to improve both the performance and the fairness of the SAT algorithm. For further information on such mechanisms refer to [Cidon1997].

Figure 5.2.2 depicts the flowchart of the SAT signal-forwarding algorithm.

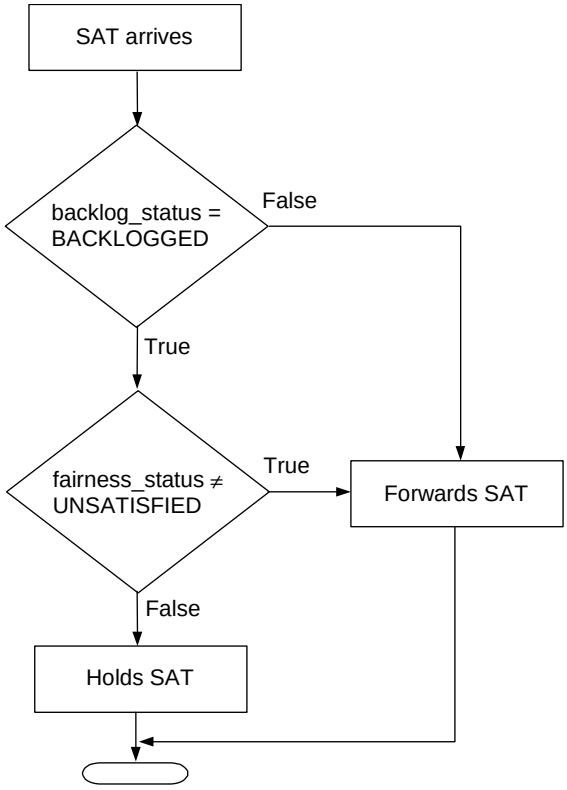


Figure 5.2.2 - Flowchart of the SAT signal-forwarding algorithm

5.2.1.2 ATMR

Based on Orwell [Falco1985], the fairness protocol developed for ATMR achieves global fairness through a cyclic credit reset procedure and a distributed window mechanism.

Each node maintains a window counter, or *windows size* in the ATMR terminology, that indicates how many cells that node may transmit within a fairness cycle, known as *reset period* in the protocol's terminology. The initial value of the window counter is set to a pre-defined credit.

A node decrements its window counter by 1 every time it transmits a cell. A node can transmit only if its window counter is greater than zero.

Every node with a window counter greater than zero and with traffic backlog to transmit, writes its own address (*busy address*) into the access control field (ACF) of each incoming cell regardless of the status of that cell. A node that finds its own address knows that all the other nodes have completed their transmissions.

A node that detects that all the other nodes have completed their transmissions issues a *reset cell*. A *reset cell* rotates around the ring resetting every node's window counter to its initial value. The node that issues a *reset cell* is responsible for removing the cell from the ring. The time interval between two consecutive visits of a reset cell defines a *reset period*.

Figure 5.2.3 depicts the FSM of the fairness protocol. For the sake of simplicity and clarity, Figure 5.2.3 shows only the events that lead to a state transition.

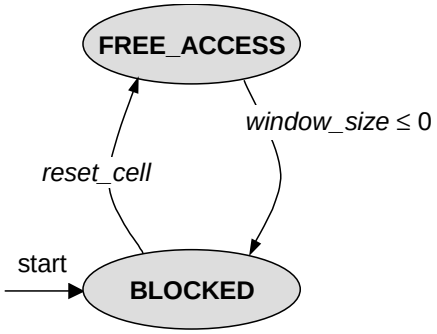


Figure 5.2.3 - FSM of the ATMR protocol

5.2.2 Local protocols

There are a few local fairness protocols proposed in the literature. This section describes the protocol presented by Chen, Cidon, Ofek, in [Chen1993], and the protocol presented by Mayer, Ofek, Yung, in [Mayer1996].

Note that both protocols assume the existence of two counter-rotating rings.

5.2.2.1 Fault-tolerant distributed local protocol

Chen, Cidon, and Ofek, in [Chen1993], describe a distributed local fairness protocol for gigabit LANs/MANs with spatial reuse. Like the global protocols described in Section 5.2.1, this local protocol uses transmission credits to regulate access to the medium. Unlike the global protocols though, which operate continuously, the local protocol operates only if necessary. Any node that detects potential starvation can trigger the protocol, at any arbitrary time.

Because starvation detection can be accomplished in different ways, the protocol can aim at either throughput fairness, or delay fairness, for instance.

The protocol can operate in the following modes:

- Non-restricted: in this mode, a node can transmit freely;
- Restricted: in this mode, a node can transmit up to a pre-defined quota of data units.

To trigger the transition from one operation mode to another the protocol uses the following two control signals:

- REQ: signal forwarded upstream over the counter-rotating ring to indicate that starvation has occurred;
- GNT: signal forwarded upstream over the counter-rotating ring to indicate that starvation has ended.

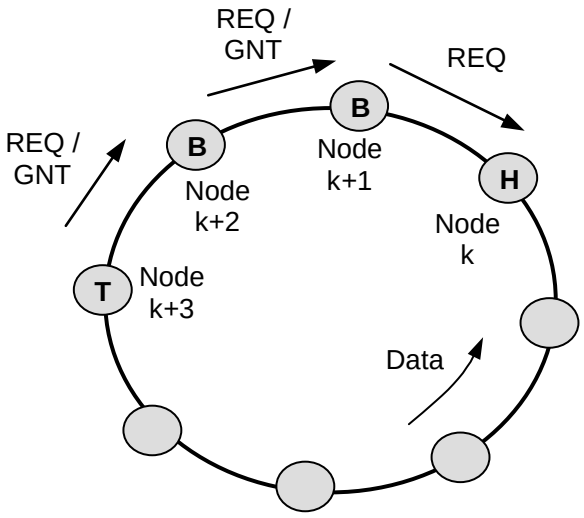


Figure 5.2.4 - Illustration of the functioning of the fault-tolerant protocol

In the initial state, all the nodes are in the unrestricted mode free access (FA) state. If a node in the FA state detects starvation, then that node transits to the restricted mode tail (T) state and issues a REQ message (see node *k*+3 in Figure 5.2.4).

To distinguish among several possible REQUEST PATHs, the protocol uses a request identifier (REQ_ID), a combination of node ID and a sequence number that uniquely identifies each REQUEST PATH. Each node maintains a REQ_ID variable, and each REQ message includes a REQ_ID that is generated by the tail node that originated the message.

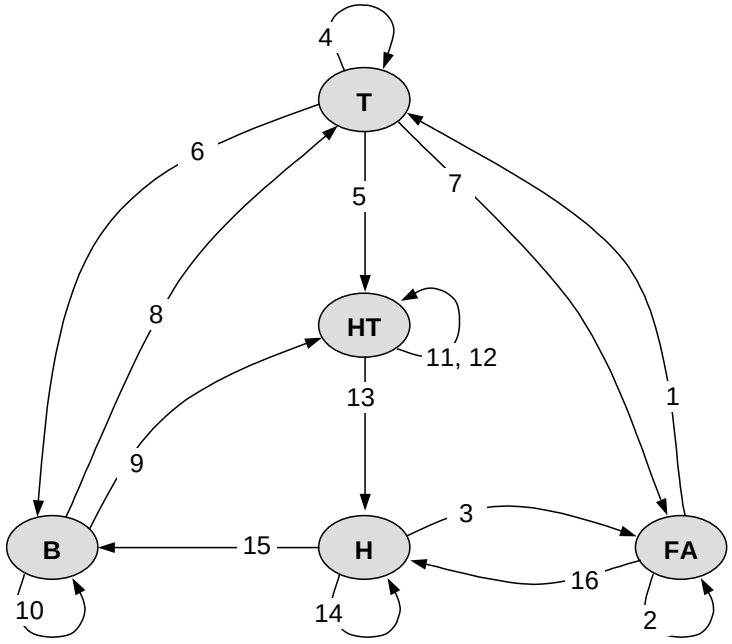
Upon reception of a REQ message, a node that is not involved in any other REQUEST PATH transits to the restricted mode as follows. First, it updates its REQ_ID with the message's REQ_ID. Second, either it transits to the restricted mode body (B) state and forwards the REQ message if it senses upstream incoming traffic (see nodes *k*+2 and *k*+1 in Figure 5.2.4), or it transits to the restricted mode head (H) state and discards the message if it does not sense upstream incoming traffic (see node *k* in Figure 5.2.4).

A node that is already involved in a REQUEST PATH, that is, that is in a state other than FA, and receives a REQ message, proceeds as follows. First, it compares the message's REQ_ID with its own REQ_ID. If a match is found, meaning that that node itself issued the message, then that node transits to the combined head-tail (HT) state and discards the message. If the message's REQ_ID is smaller than or equal to the node's REQ_ID, then the node transits to the body state and discards the message. In other words, the node merges both REQUEST PATHs. If the message's REQ_ID is greater than the node's REQ_ID, then the node replaces its REQ_ID with the message's REQ_ID and forwards the message upstream.

Upon satisfaction, that is, transmission quota has exhausted, a tail node issues a GNT signal upstream and transits back to the non-restricted FA state.

A node that receives a GNT message follows similar rules as a node that receives a REQ message. If that node is in the B state, then that node transits to the T state and forwards the message upstream. If that node is in the H state, then that node transits back to the non-restricted mode FA state, clears its REQ_ID variable and discards the message.

Figure 5.2.5 depicts the complete FSM of the protocol. Prefix R(eceived) means indicates an incoming event.



1 - Starvation
2 - R_GNT
3 - R_REQ(j)
4 - R_GNT
5 - R_REQ(j), j = REQ_ID
6 - R_REQ(j), j ≠ REQ_ID
7 - Satisfied
8 - R_GNT
9 - R_REQ(j), j = REQ_ID
10 - R_REQ(j), j ≠ REQ_ID
11 - R_GNT
12 - R_REQ(j)
13 - Satisfied
14 - R_REQ(j)
15 - Upstream busy
16 - R_GNT

Figure 5.2.5 - FSM of the fault-tolerant protocol

5.2.2.2 Distributed local scheduling with partial information

Mayer, Ofek, and Yung, in [Mayer1996], describe another distributed local fairness protocol. The protocol aims at throughput fairness and it follows the credit model to achieve its goal.

The idea of this protocol is to obtain rates that approximate those of the Max-Min fair throughput optimisation problem [Berts1987], but without the global knowledge that the latter requires.

The Max-Min optimisation algorithm works as follows.

1. For each link assign a residual capacity of 1, that is, the total capacity;

2. For each session associate the unassigned label;
3. Find all the bottleneck links. The bottleneck links are the links with the smallest residual capacity per unassigned session sharing them;
4. Divide the residual capacity of a bottleneck link by the number of unassigned sessions that share that link. Assign the result of such a division to be the Max-Min fair rate of those sessions and label them assigned;
5. Compute the residual capacity of all links by subtracting the rates which have been assigned to sessions in step 4;
6. Repeat 3-5 until all sessions are assigned.

Figure 5.2.6 illustrates a scenario containing a bottleneck link on a ring with four nodes. In the example, the nodes and the links are indexed according to their topological position. Each node generates a single full-capacity session, which is represented by a thick line and identified by the expression (*S*, *D*), where *S* contains the index of the source node and *D* contains the index of the destination node.

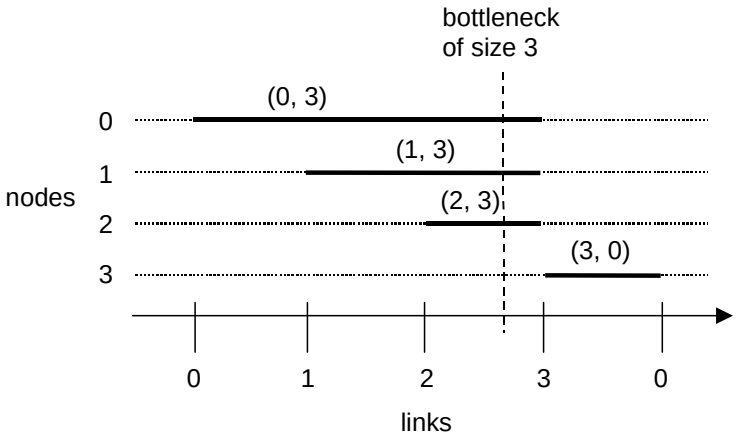


Figure 5.2.6 - Bottleneck link on a ring with four nodes

The feasibility condition is that for each link the sum of the rates allocated to sessions sharing that link should not exceed 1.

By applying the Max-Min algorithm the following set of bottleneck links is found: {(0,3), (1,3), (2,3)}. Therefore, each of the sessions belonging to the set gets a rate of 0.33; session (3,0) gets the residual rate of link 3, which is 1.

The local scheduling protocol performs local scheduling based on partial information that is collected through the exchange of signals. Every node maintains a table, referred to as mode, with an entry for every other node. Each entry *j* reflects the status of a node *i* with respect to node *j*. The statuses are:

- Unregulated: an entry *j* in the unregulated status means that node *i* can transmit through node *j* freely;

- Regulated: an entry *j* in the regulated status means that node *i* can transmit one more quota through node *j* before it becomes exhausted;
- Exhausted: an entry *j* in the exhausted status means that node *i* can no longer transmit through node *j*.

Figure 5.2.7 illustrates the table mode of a node *A*.

Destination ID	Status
B	Exhausted
C	Regulated
D	Irrevelant
E	Irrevelant
F	Irrevelant
G	Irrevelant
H	Irrevelant

Figure 5.2.7 – Example of table mode at node *A*

The following two control signals are responsible for updating table modes:

- R-signal: a node *j* issues an R-signal upstream to signal that it wants to start transmitting another quota;
- U-signal: a node *j* issues a U-signal upstream to signal that it has finished transmitting a quota.

If a node *i* is transmitting and it receives an R-signal from a node *j*, then node *i* knows that a new conflict has arisen and thus change entry *j* to regulated.

If a node *i* is in the regulated status with respect to *j* and receives a U-signal from node *j*, then node *i* changes entry *j* to unregulated.

Figure 5.2.8 depicts the FSM of the protocol.

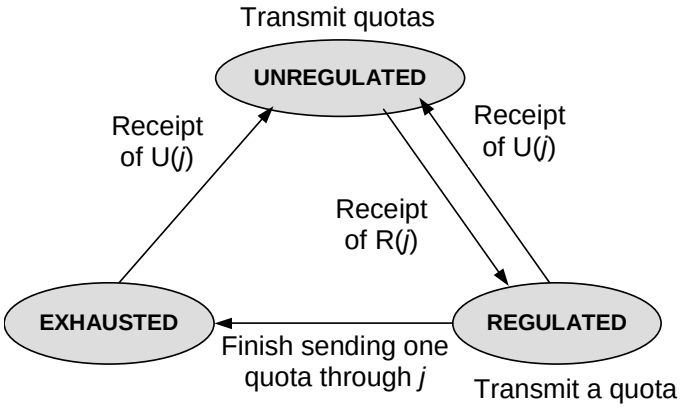


Figure 5.2.8 - FSM of the local scheduling protocol

5.2.3 Discussion

Each group of fairness protocols has advantages and disadvantages compared with the other. Even within each group each protocol has advantages and disadvantages compared with the other.

Under symmetric traffic conditions global protocols and local protocols achieve approximately the same performance. On the other hand, under asymmetric traffic conditions the local protocols outperform the global ones, and that is because global protocols may prevent a node from transmitting even if the intended transmission is not going to traverse a bottleneck link. Since they operate continuously, even if there is no starvation, global protocols may limit transmissions unnecessarily.

There is only one study that compares the performance fairness of both groups of protocols [Anast2001]. The study compares and analyses the (aggregate) throughput deviation from the optimum (aggregate) throughput, and the throughput fairness deviation of the SAT protocol, the fault-tolerant protocol, and local scheduling protocol. Three distinct traffic scenarios are considered: 1) disjoint groups and homogeneous workloads; 2) disjoint groups and non-homogeneous workloads; and 3) non-disjoint groups and non-homogeneous workloads.

The study shows that distributed local scheduling [Mayer1996] achieves higher performances than the other two protocols, in particular for scenarios belonging to scenario 3, which happens to be the most realistic situation. The study also shows that the SAT protocol achieves the worst fairness.

Because global protocols assign credits to nodes statically they are more susceptible to changes in traffic conditions than local protocols, and, consequently, tend to suffer more from dynamic changes in traffic conditions.

Although there is no study to prove it, local protocols should adapt better, but not necessarily well, to dynamic changes in traffic conditions because in such protocols nodes can trigger fairness messages, at arbitrary times, as needed.

Global protocols are simpler than local protocols. Global protocols rely on a continuous process rather than on event triggering, and continuous processes are simpler than event triggering -the FSM of each protocol makes this point clear.

All the described protocols forward control signals to upstream nodes using a counter-rotating ring to reduce propagation delays.

Actual networks use counter-rotating rings for protection, so forwarding control signals upstream using a counter-rotating ring is acceptable. Nevertheless, MOPS rings may consists of two, four or more counter-rotating rings, each of them carrying tens or hundreds of wavelengths. Maintaining the consistency between all the control-data relations may be difficult under such conditions.

5.3 Proactive fairness protocols

This section introduces two proactive fairness protocols. The first is a global fairness protocol. The second is a local fairness protocol that builds on the global

fairness protocol. Both protocols aim at throughput fairness, although at different levels.

The list below identifies the principles that drove the design of the protocols:

1. Fairness and efficiency: a node should obtain a share that is not less than what is fair and not more than what it needs;
2. Pro-action: the algorithm should be proactive, co-ordinating access to prevent starvation rather than taking measures to correct starvation after it occurred;
3. Automation and adaptation: the algorithm should require minimal manual configuration of input parameter values, and adapt quickly to dynamic fluctuations in traffic workloads and dynamic changes in traffic distributions;
4. Simplicity: the algorithm should be simple, exchanging the least number of messages, defining the least number of distinct behaviours, and requiring the least processing power;
5. Performance: the algorithm should provide network nodes with high throughputs and low delays.

The protocols are described next.

5.3.1 Global cyclic reservation (GCR)

The global cyclic reservation (GCR) fairness protocol combines cyclic reservations with credit allocations. It grants transmission rates to nodes periodically, according to their reservation requests, hence conforming partly to design principle 1, and entirely to design principle 2.

GCR comprises three main mechanisms:

- Cyclic reservation: the cyclic reservation mechanism deals with the calculation of reservation requests and distribution of such requests;
- Fair rate calculation: the fair rate calculation algorithm determines how much traffic a node can transmit during a fairness cycle, whereas a fairness cycle is the time difference between two consecutive departures of the control packet from the same node;
- Fairness enforcement: the fairness enforcement mechanism makes sure that transmission occurs only if it is in accordance with the calculated fair rate.

Cyclic reservation

The cyclic reservation mechanism uses a single, fixed frame layout control packet that circulates the ring in the data direction to collect and distribute reservation requests.

A reservation request expresses how much medium capacity a node needs to cope with its demand, and it can be absolute or a fraction of the nominal capacity available during a fairness cycle. Whichever option is chosen though, the fairness algorithm has to follow the same option throughout.

A node's demand is the sum of the entire traffic backlog at that node. Let N be the number of nodes in the ring, and n_i, for $i = 0,..., N-1$, be the i-th node on the

ring. Let also d_q, for $q = 0,..., N-1$, be the traffic backlog at queue q. Node i's total demand is given by

$$D_i = \sum_{q=0}^{N-1} d_q .$$

Assuming that fractional reservation is chosen, node i's demand is adjusted to

$$E_i = \min\left(\frac{D_i}{C_f}, 1\right),$$

where C_f denotes the capacity available during a fairness cycle. Let L_r be the ring latency, S be the medium bit rate, and W be the number of wavelength channels that the node can access simultaneously. The capacity of a fairness cycle is given by $C_f = L_r \times S \times W$.

The control packet carries the reservation requests vector $r = \{r_0,..., r_{N-1}\}$, where r_i, for $i = 0,..., N-1$, denotes node i's request. Initially, all elements of r are set to 0. As the control packet circulates the ring, each node inserts its request by updating the corresponding element in r.

To be able to calculate its own fair rate, a node has to calculate every other node's fair rate too, and to do so that node has to obtain the requests made by the other nodes during the current fairness cycle.

Each node maintains the local request vector $\alpha = \{\alpha_0,..., \alpha_{N-1}\}$. Upon arrival of the fairness control packet, a node j, amongst other things, executes the following ordered steps:

1. Save the requests made by the upstream nodes locally; that is, $\alpha_i = r_i$, for $i = 0,..., j$;
2. Calculate a new request E_j and inserts it into the control packet accordingly; that is, $r_j = E_j$;
3. Save the requests made by the downstream nodes locally; that is, $\alpha_i = r_i$, for $i = j+1,..., N-1$;
4. Forward the control packet.

Fair rate calculation

A node calculates its fair rate for the fairness cycle f at the end of the fairness cycle f-1, in between steps 2 and 3 of the cyclic reservation mechanism.

The execution of the fair rate calculation algorithm yields the fair rate vector $\beta = \{\beta_0,..., \beta_{N-1}\}$, where β_i, for $i = 0,..., N-1$, denotes the fair rate to be granted to node i. The calculated fair rate determines how much traffic a node can transmit during a fairness cycle, and it can express bits, bytes, or slots depending on whether the network is synchronous or asynchronous.

To comply to the design principle 1, the algorithm commits a rate to node i according to the following rule:

$$\beta_i = \begin{cases} \alpha_i & \text{, if } \alpha_i \le r_{fair} \\ & \\ r_{fair} & \text{, if } \alpha_i > r_{fair} \end{cases},$$

where r_{fair} denotes the fair rate at that time.

Let x be the number of nodes competing for the medium, or the number of requests, and c be the residual capacity of the medium. The fair rate is given by

$$r_{fair} = \frac{c}{x}.$$

The medium's residual capacity and the number of entities competing for such a capacity change as the algorithm evaluates the requests, and so does the fair rate. Therefore, given n_i and n_j, where $i < j$, n_j's committed rate depends on n_i's committed rate.

Given such a dependency, to achieve high efficiency and performance the algorithm evaluates the requests in ascendant order. How to achieve such an evaluation order is an implementation issue.

After calculating node i's final rate, for $i = 0,..., N-1$, the algorithm executes the following basic steps:

- Update the residual capacity by subtracting the calculated rate from it; that is, $c \leftarrow c - \beta_i$;
- Update the number of requests by decrementing it by 1; that is, $x \leftarrow x - 1$;
- Re-calculate the fair rate; and
- Move to the next request by incrementing i by 1; that is, $i \leftarrow i + 1$.

The flowchart in Figure 5.3.1 depicts the fairness algorithm. Observe that when the sum T of all requests is smaller than 1 there is no bottleneck, and, therefore, the algorithm grants each node its own request. Observe also that, to evaluate requests in ascendant order, the algorithm sorts the local requests vector before calculating the fair rates.

Sorting the request vector raises the question of how to keep track of which request corresponds to which node.

One implementation option is to index the request vector before sorting it, such that each index identifies the original position of the corresponding request. The flowchart in Figure 5.3.1 uses such an option.

The protocol assigns the calculated fair rate to the corresponding node right before forwarding the control packet to the next node.

Let tx be the transmission credit of node i. At the beginning of the next fairness cycle node i updates tx with the calculated fair rate; that is, $tx = \beta_i \times C_f$.

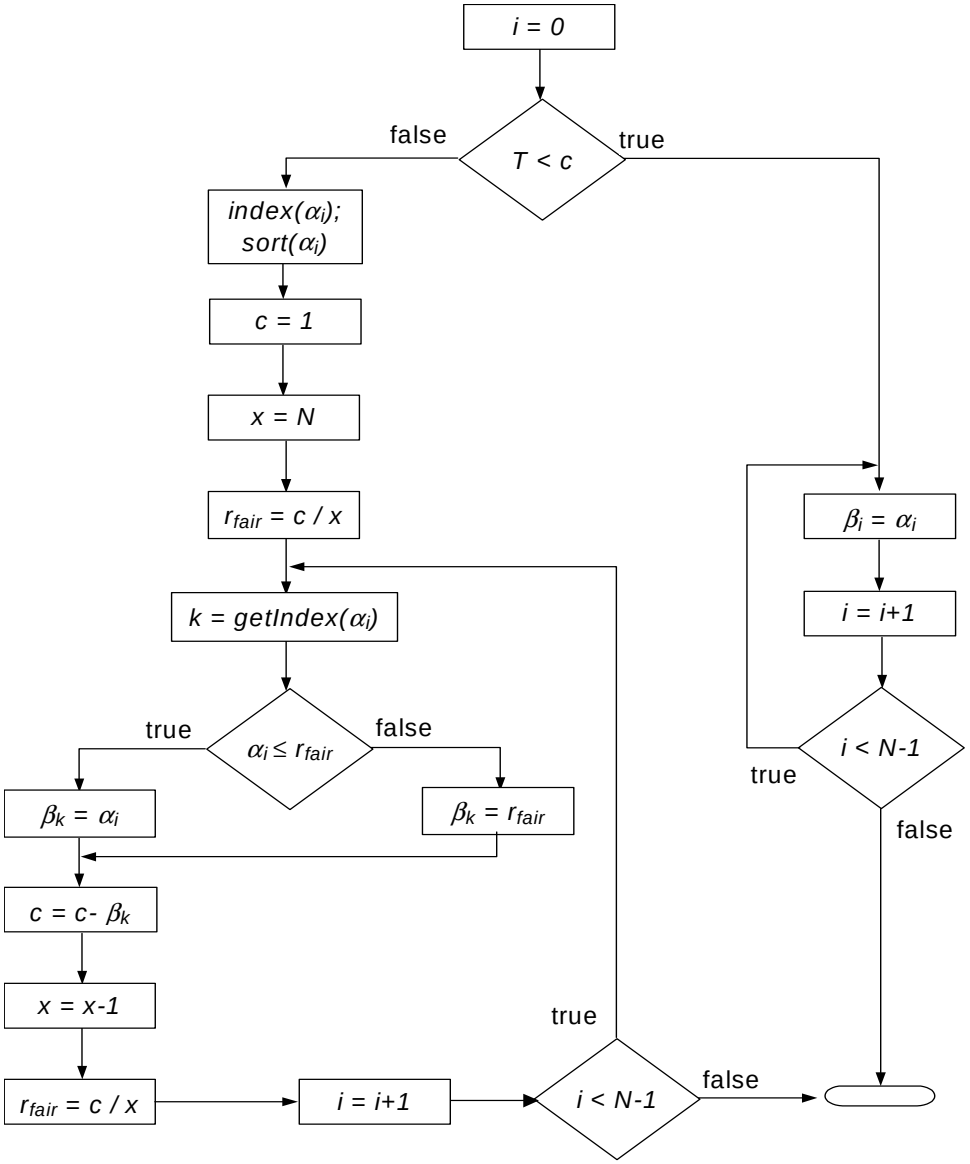


Figure 5.3.1 – Flowchart of the GCR fair rate calculation algorithm

Fairness enforcement

Let tx_i be node i's remaining transmission quota, and let s be the size of the packet selected for transmission. Transmission is allowed if tx_i is greater than s, or equal to s, that is, if $tx_i \geq s$.

To make it possible to enforce fairness, the fairness protocol has to account for each transmission made by the corresponding node. Thus, upon each transmission, the fairness protocol updates the remaining transmission credit; that is, $tx_i \leftarrow tx_i - s$.

Example

The example assumes a four-node ring and a request vector $r = \{1,0.3,0.7,0.2\}$. Figure 5.3.2 shows the pre-calculation phase, which indexes and sorts α.

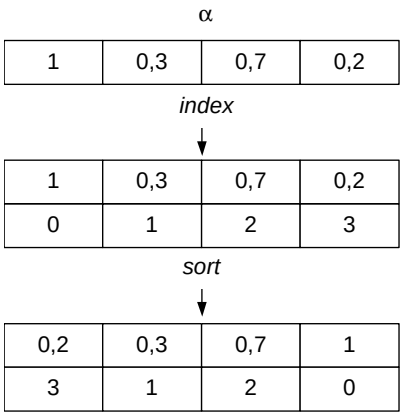


Figure 5.3.2 - Fair rate pre-calculation phase

Figure 5.3.3 shows the second phase, which is the core of the fairness algorithm and yields β. Note that the element in β to store a calculated fair rate is given by the index associated to the corresponding request in α and not the actual position of that request in α.

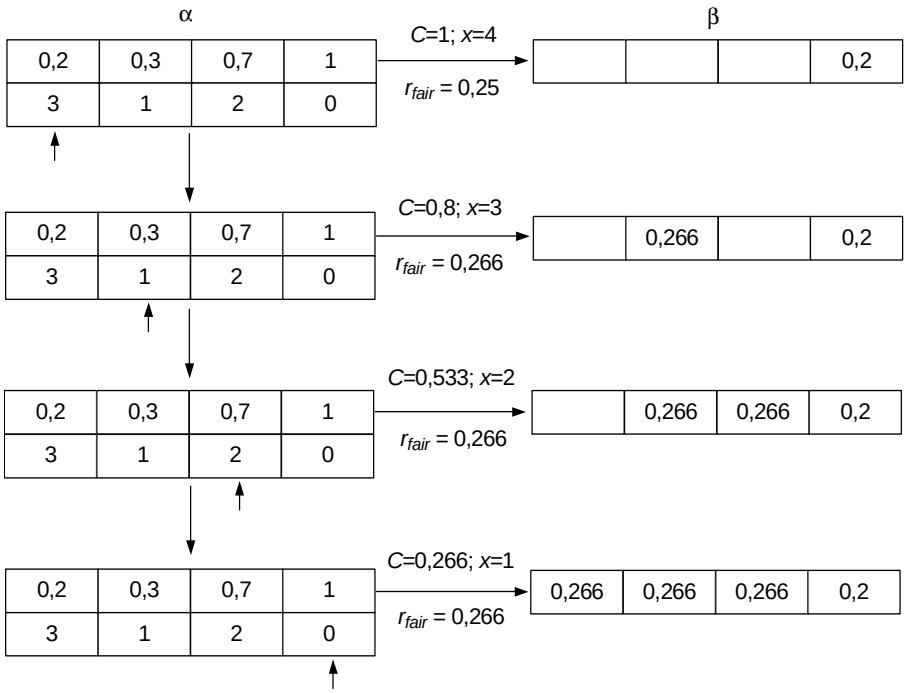


Figure 5.3.3 – Steps of the fair rate calculation phase

5.3.2 Local cyclic reservation (LCR)

The local cyclic reservation (LCR) fairness protocol [Salva2003c] builds on GCR. It combines cyclic reservations with credit allocations and assigns fair transmission quotas to nodes according to their requests.

The algorithm grants transmission rates to nodes periodically, according to their reservation requests, hence conforming partly to design principle 1, and entirely to design principle 2.

LCR comprises three main mechanisms:

- Cyclic reservation: the cyclic reservation mechanism deals with the calculation of reservation requests and distribution of such requests;
- Fair rate calculation: the fair rate calculation algorithm determines how much traffic a node can transmit over each link during a fairness cycle, whereas a fairness cycle is the time difference between two consecutive departures of the control packet from the same node;
- Fairness enforcement: the fairness enforcement mechanism makes sure that a transmission occurs only if it is in accordance with the corresponding calculated fair rate.

Cyclic reservation

The cyclic reservation mechanism uses a single, fixed frame layout control packet that circulates the ring in the data direction to collect and distribute reservation requests.

A reservation request expresses the link's capacity a node needs in the next fairness cycle to cope with its demands, and it can be absolute or a fraction of the nominal capacity available during a fairness cycle. Whichever option is chosen though, the fairness algorithm has to follow the same option throughout; for the sake of legibility, from now on fractional reservation is assumed.

A node's demand over a certain link is the sum of the traffic backlog at that node destined to that link and to the links downstream to that link. Let N be the number of nodes in the ring, and n_i, for $i = 0,..., N-1$, be the i-th node on the ring. Let also $L = N$ be the number of links in the ring, and l_k, for $k = 0,..., L-1$, be the k-th link of the ring such that link l_k connects node n_k to node n_{k+1}.

To avoid HOL blocking, LCR requires VOQ. Each node maintains N-1 queues, one per possible destination node, and these queues are organized in a sequence that reflects the logical topology as seen by that node. Thus, node i's demand over link k is given by

$$D_i^k = \sum_{q=|i+L-1|_L}^{k} d_i^q ,$$

where d_i^q, for $q = 0,..., N-1$, denotes the traffic backlog at node i's queue q; the symbol $|\,.\,|_V$ means the modulo V of ".".

To generate the definitive fractional demands, the algorithm observes whether node *i*'s total demand, which is given by D_i^k, for $k = i$, exceeds the total capacity of the link during the fairness cycle or not.

Let L_r be the ring latency, S be the bit rate of the medium, and W be the number of wavelength channels that the node can access simultaneously. The capacity of a fairness cycle is given by $C_f = L_r \times S \times W$.

Let E_i^k be the definitive fractional demand of node i over link k. If D_i^k, for $k = i$, is smaller than C_f or equal to it, then, for $k = 0,...,L-1$,

$$E_i^k = \frac{D_i^k}{C_f}.$$

If D_i^k, for $k = i$, is greater than C_f, then, for $k = 0,...,L-1$,

$$E_i^k = D_i^k \times f,$$

where $f = \frac{C_f}{D_i^k}$, for $k = i$, is the demand correction factor.

The control packet carries the reservation requests matrix

$$r = \begin{bmatrix} r_0^0 & \cdot & \cdot & r_0^{L-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ r_{N-1}^0 & \cdot & \cdot & r_{N-1}^{L-1} \end{bmatrix},$$

where r_i^k, for $k = 0,...,L-1$ and $i = 0,...,N-1$, denotes node *i*'s request over link *k*.

Initially, all elements of *r* are set to 0. As the control packet circulates the ring, each node inserts its request over each link by updating the corresponding elements in *r*.

To be able to calculate its own fair rate over a certain link, a node has to calculate every other node's fair rate over every link, and to do that a node has to obtain the requests made by all the other nodes.

A node stores such values in the local request matrix

$$\alpha = \begin{bmatrix} \alpha_0^0 & \cdot & \cdot & \alpha_0^{L-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \alpha_{N-1}^0 & \cdot & \cdot & \alpha_{N-1}^{L-1} \end{bmatrix}.$$

Initially, all elements of α are set to 0. As the control packet circulates the ring, a node updates each element of its local request matrix with the corresponding element in *r*.

Upon arrival of the fairness control packet, a node *j*, amongst other things, executes the following ordered steps:

1. Obtain the requests made by upstream nodes from the control packet; that is, $\alpha_i^k = r_i^k$, for $k = 0,...,L-1$ and $i = 0,..., j-1, j$;
2. Calculate new link requests and inserts them into the control packet accordingly; that is, for $k = 0,...,L-1$, calculate E_j^k and let $r_j^k = E_j^k$;
3. Obtain the requests made by downstream nodes from the control packet; that is, $\alpha_i^k = r_i^k$, for $k = 0,...,L-1$ and $i = j+1,...,N-1$;
4. Forward the control packet.

Fair rate calculation

A node calculates its own per link fair rate for the fairness cycle *f* at the end of the fairness cycle *f*-1, in between steps 2 and 3 of the cyclic reservation mechanism.

The execution of the algorithm yields the fair rate matrix

$$\beta = \begin{bmatrix} \beta_0^0 & . & . & \beta_0^{L-1} \\ . & . & . & . \\ . & . & . & . \\ \beta_{N-1}^0 & . & . & \beta_{N-1}^{L-1} \end{bmatrix},$$

where β_i^k, for $k = 0,...,L-1$ and $i = 0,...,N-1$, denotes the fair rate to be granted to node *i* over link *k*. The calculated fair rate determines how much traffic node *i* can transmit over link *k* during a fairness cycle, and it can express bits, bytes, or slots depending on whether the network is synchronous or asynchronous.

Let T^k be the sum of all the requests on link *k*. If T^k is greater than 1 then link *k* is a bottleneck. The greater T^k, the heavier the bottleneck is.

The algorithm starts from the heaviest bottleneck link towards the lightest bottleneck link, since solving a heavy bottleneck link may automatically solve a lighter bottleneck link.

Since solving a single bottleneck link is the same as solving an overloaded medium, the algorithm reuses the GCR fair rate calculation algorithm to solve each bottleneck link. Nevertheless, in LCR the fair rates calculated by GCR update α rather than generate β, as originally in that protocol.

Solving a bottleneck link may result in a situation in which a node's granted rate over a link *j*+1 is greater than that same node's granted rate over a link *j*. This excess rate on link *j*+1 cannot be used by that node, and consequently, by any other node. For efficiency, given an ex-bottleneck link of index *k*, for each node *i* with a request on link *k*, and for each downstream link *j* until node *i*'s request on link *j* is smaller than node *i*'s request on link *k*, or equal to it, then update the request on link *j* with the request on link *k*. That is, if $\alpha_i^j > \alpha_i^k$ then let $\alpha_i^j = \alpha_i^k$.

Figure 5.3.4 shows the flowchart of the algorithm to update the requests over downstream links. The algorithm uses the following auxiliary variables:

- *y*: stores the index of the farthest downstream link of node *i*;

- *z*: stores the index of a downstream link as the algorithm iterates through the downstream links towards link *y*;
- *w*: stores intermediate results for the calculation of *z*.

Note that how *y* and *z* are calculated is implementation specific; therefore they can be calculated in different ways.

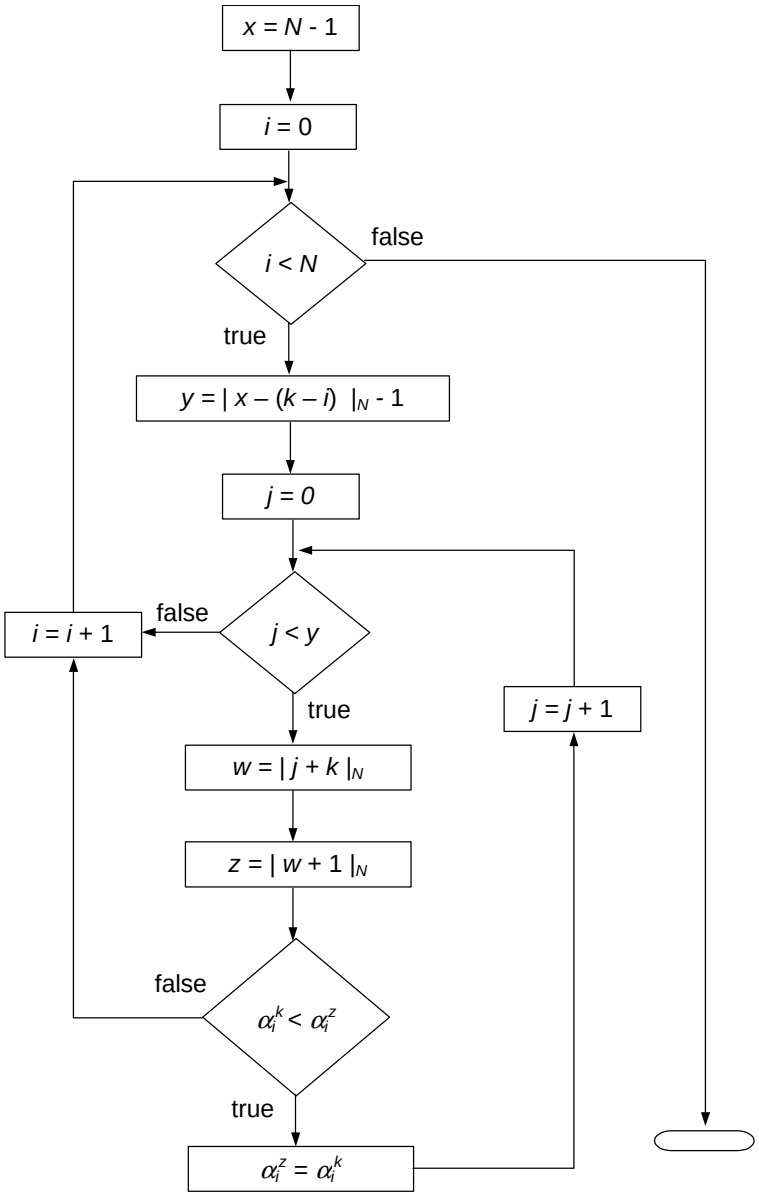


Figure 5.3.4 – Algorithm for updating downstream link requests

At a high level of abstraction, the complete fair rate calculation algorithm is as follows:

```
Select the heaviest bottleneck link
While there is a bottleneck link
   Solve that link
   Update downstream link requests
   Select the heaviest bottleneck link
End
```

After the algorithm calculates all the fair rates, it generates the matrix β and updates each element in β with the value of the corresponding element in α; that is, $\beta_i^k = \alpha_i^k$, for $i = 0,...,N-1$ and $k = 0,...,L-1$.

The algorithm then assigns the calculated fair rates to the node. Let tx_i^k be node *i*'s transmission credit over link *k*. At the beginning of the next fairness cycle node *i* updates tx_i^k, for $k = 0,...,L-1$, with the corresponding calculated fair rates; that is, $tx_i^k = \beta_i^k \times C_f$.

Eventually the node forwards the control packet to the next node.

Example

Figure 5.3.5 shows an example in which four nodes generate uniformly distributed traffic, and each of these nodes demands the total nominal capacity of the network distributed equally over the links to be traversed.

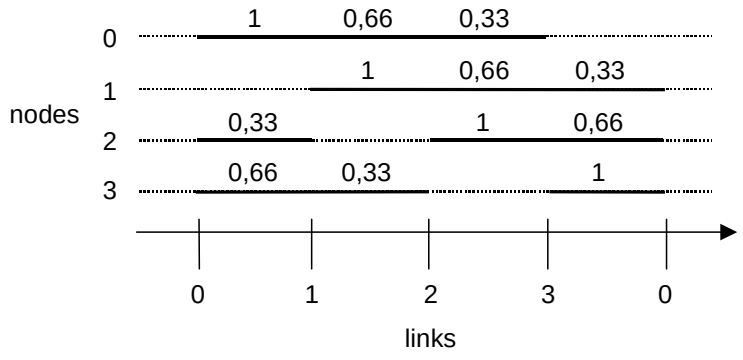


Figure 5.3.5 – Example of link requests on a four-node ring

Figure 5.3.6 shows the evolution of the algorithm as it solves one bottleneck link after another. Rates modified in the previous step are underlined. The sum of a link's requests is shown in bold.

	1	0,66	0,33	-
	-	1	0,66	0,33
start	0,33	-	1	0,66
	0,66	0,33	-	1
	1,99	**1,99**	**1,99**	**1,99**

⇓

	1	0,66	0,33	-
	-	1	0,66	0,33
1st. step	0,33	-	1	0,33
	0,33	0,33	-	0,33
	1,66	**1,99**	**1,99**	**0,99**

⇓

	1	0,66	0,33	-
	-	1	0,33	0,33
2nd. step	0,33	-	0,33	0,33
	0,33	0,33	-	0,33
	1,66	**1,99**	**0,99**	**0,99**

⇓

	1	0,33	0,33	-
	-	0,33	0,33	0,33
3rd. step	0,33	-	0,33	0,33
	0,33	0,33	-	0,33
	1,66	**0,99**	**0,99**	**0,99**

⇓

	0,33	0,33	0,33	-
	-	0,33	0,33	0,33
final step	0,33	-	0,33	0,33
	0,33	0,33	-	0,33
	0,99	**0,99**	**0,99**	**0,99**

Figure 5.3.6 –Step-by-step illustration of the LCR fair rate calculation phase

5.4 Discussion

This chapter introduced two fairness protocols called GCR and LCR. GCR aims at global fairness and, as such, can provide fairness only under certain traffic conditions. LCR, on the other hand, aims at local fairness and, as such, can provide fairness independent of the traffic distribution pattern.

Both protocols use cyclic reservations to enforce fairness proactively, adapting dynamic and automatically to the network demands and traffic distribution patterns.

The use of cyclic reservations introduces some complexity, but both protocols can be implemented with existing technologies.

Table 5.4-1 highlights the main aspects of the protocols in comparison with SAT, which is adopted in almost all the MAC protocols for MOPS rings.

Table 5.4-1 – Comparison among LCR, GCR, and SAT

	SAT	**GCR**	**LCR**
Fairness	Depends on the traffic condition	Depends on the traffic condition	High independent of the traffic condition
Dynamic adaptability	No	Yes	Yes
Complexity	Low	Medium	High
Manual configuration	Yes	No	No

The integration of GCR and LCR with the proposed access control protocols is an important step towards achieving an efficient, high-performance MAC protocol. Each access control protocol may require a particular form of integration.

As far as SAT is concerned, a single bit in the slot header can be used to represent the SAT signal. The integration of GCR and LCR is not as trivial though. Because of the limitations in the control channel of MOPS rings and the size of fairness control packets, a fairness control packet has to be transmitted using payload slot(s) just like data packets. Consequently, fairness control packets may experience access delays just as normal packets do, and such delays may have negative effects on performance, fairness, or both.

GCR has to assume a certain fairness cycle capacity to calculate fair rates that exploit spatial reuse. Nevertheless, the spatial reuse factor depends on the traffic conditions, which may change dynamically. Thus, if the protocol assumes a symmetric traffic condition, and therefore a reuse factor of 2, and during operation the traffic condition becomes asymmetric, hence changing the reuse

factor, the network nodes may experience unfairness and achieve poor performances.

To cope with traffic dynamics and possible unfairness, GCR uses fairness cycles of variable lengths rather than constant ones. Thus, a node that receives the global fairness control packet releases that packet only after that node used its transmission quota, hence, not granting new quotas to other nodes.

LCR does not depend on reuse factors because it considers each link isolated, and there is no reuse at link level. Thus, LCR uses fairness cycles of fixed length, that is, a node releases the control packet as soon as possible according to the access control protocol.

For high performance LCR requires VOQ. Thus, even though SPT+P and SPT+R work efficiently with single packets queuing, when integrated with LCR those protocols have to implement VOQ.

The integration of LCR and GCR with SPT+SC deserves special care. Because slot trains get fragmented into smaller trains as they travel, nodes farther from the node that possesses the token have more difficulty to transmit large packets. If the LCR rules apply to nodes regardless of whether the token is held then it might happen that a node transmits many small packets when it does not hold the token, but when that node obtains the token LCR forbids that node to transmit. Consequently, large packets experience large delays, and the node throughput degrades.

To avoid such a phenomenon, the LCR rules apply only to nodes that do not possess the token, and LCR does not account for transmissions when the token is held.

Both protocols are general and can be used in either synchronous, or asynchronous networks. Nevertheless, to work properly in synchronous networks both fairness enforcement mechanisms have to account for each transmission as sending multiples of the size of a slot rather than the actual number of bits transmitted.

Also, in both protocols the fairness control packet is sent as normal data, thus, when transmitting the fairness control packet a node should account for the transmission as usual.

Chapter 6 discusses the effects of such integration strategies on the performance of the integrated protocols.

Chapter 6
Performance evaluation

This chapter focuses on the performance of the protocols described in Chapters 4 and 5. First, it analyses the performance potential of the fairness control protocols, using a general slotted ring with destination removal network and three traffic distribution scenarios. Then, it analyses the performance of each access control protocol integrated with both SAT and LCR under two traffic distribution scenarios.

6.1 Goals

Performance evaluation may serve to different purposes. The main objectives of the performance evaluation carried out in this chapter are:

- To evaluate the fairness and performance potentials of the fairness protocols;
- To evaluate the performance of the access control protocols when combined with the fairness protocols;
- To understand the behaviour of both the access control protocols and the fairness protocols;
- To understand the impact of the fairness protocols on the performance of the access control protocols;
- To understand and evaluate how network conditions affect the performance and the effectiveness of the protocols.

6.2 Approach

The following two complementary performance evaluation techniques are recommended at initial stages of design: analytical modelling and computer simulation. Analytical modelling uses mathematics to construct a model of the target system, and it provides the best insight into the effects of various parameters and their interactions. Nevertheless, analytical modelling usually requires simplifications and assumptions, and the resulting model may become too distant from the original target model. Computer simulation uses a computer programming language to create a model of the target system that can be executed to imitate that system in the real world. Computer simulation provides the best accuracy and is closest to the reality, but it provides insight only to particular scenarios and requires considerable computation power, computation time, and memory.

Destination removal slotted ring networks are difficult to model analytically. There are only a few studies that use analytical modelling to evaluate the performance of such networks [Anast1997, Bengi2002, Bux1981, Mitra1986, Morri1984, Zafir1987, Zaffir1988, Zafir1999], but these studies either assume slots and packets to be of the same size, or consider a single traffic distribution pattern.

Because of the difficulty in modelling MOPS rings analytically, and given the number of access control and fairness protocols studied in this work, analytical modelling is unfeasible. Thus, this work uses computer simulation to evaluate the performance of the proposed protocols.

6.3 Simulators

The simulators are of discrete-event type. A discrete-event simulation can be described as a system representation in which events of interest occur at discrete points in time, and the ordered occurrence of such events together with the time at which they occur describe the system operations.

The simulators follow the next-event time-advance approach. A next-event time-advance simulator uses a simulation clock to keep track of the current value of simulated time as the simulation proceeds. Given a list of events to occur in the future, the simulation clock always advances to the time of occurrence of the most imminent event, hence triggering the occurrence of that event. As a result, the state of the system, the time of the next occurrence of that event, and the simulation clock are updated. Repetition of such a process occurs until a pre-defined stop condition is satisfied.

Figure 6.3.1 [Law2000] depicts the conceptual structure of next-event time-advance simulators.

The simulators use confidence interval as stop condition. The use of confidence interval as stop condition ensures that the accuracy of the output data falls below an acceptable threshold [Law2000, Jain1991]. Periodically the simulator checks if the stop condition is met. If yes then the simulation ends. If not, the simulation continues.

The design of the simulators follows object orientation principles, which model a system as a set of objects that interact via well-defined interfaces. The classes, the relationships between the classes, and the interfaces are the same for all the simulators, but the behaviour of the classes that implement the access control and access fairness functions are different.

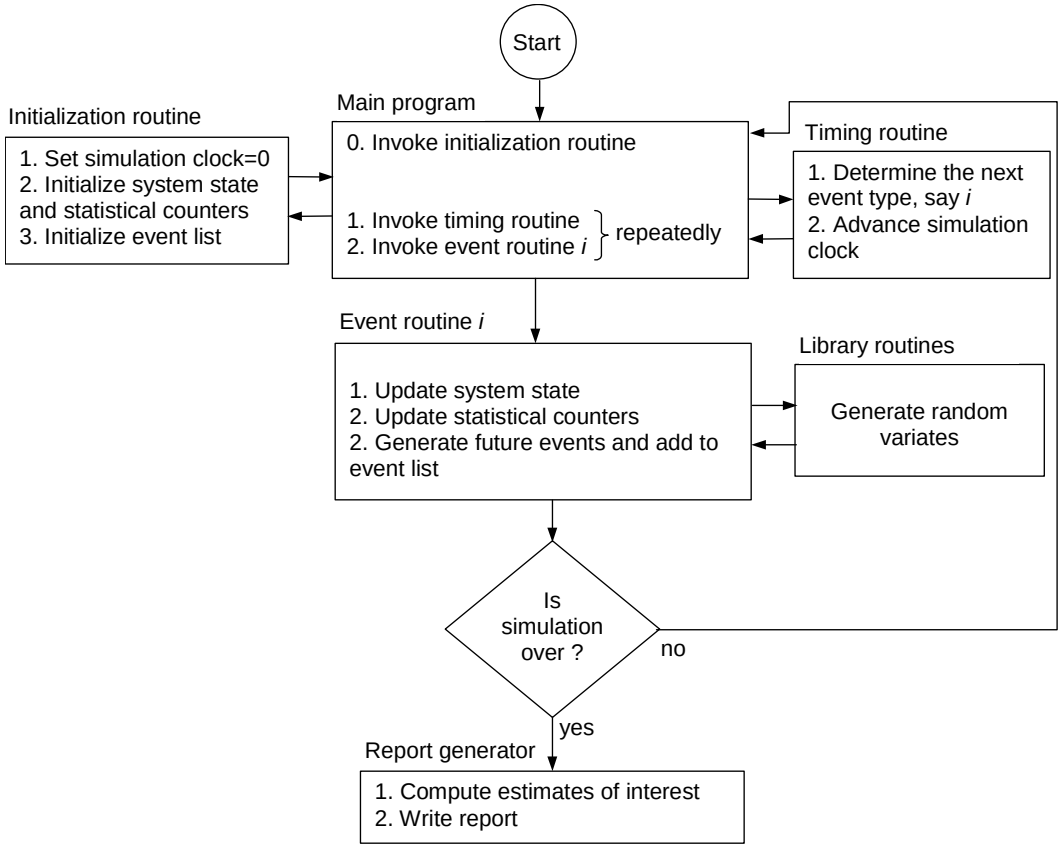


Figure 6.3.1 - Next-event time-advance approach

Figure 6.3.2 depicts the organization of the simulators using the unified method language (UML) [Fowle1999] conceptual class diagram -note that for the sake of clarity the classes contain no attributes or methods. UML is becoming the de-facto standard object modelling language, and its conceptual class diagram represents the concepts of the domain under study independently of implementation.

The diagram models a MOPS ring as a single fibre ring consisting of one channel or more. Each channel, in turn, is divided into one time slot or more.

The fibre ring connects two nodes or more. A node implements the access control protocol to be simulated, and it consists of a queuing system, one Rx unit or more and one Tx unit or more. An Rx unit can receive on one channel at a time, but the channel can be fixed-tuned or tuneable. Likewise, a Tx unit can transmit on one channel at a time, but the channel can be fixed-tuned or tuneable.

The queuing system consists of one packets queue or more, and it implements packet scheduling strategies for retrieving such packets.

The fairness class implements the access fairness algorithm and it superposes the access rules defined by the access control algorithm.

Each node is assigned a unique traffic generator. A traffic generator is a packet factory; it creates packets to the node it is associated with according to a pre-defined destination distribution.

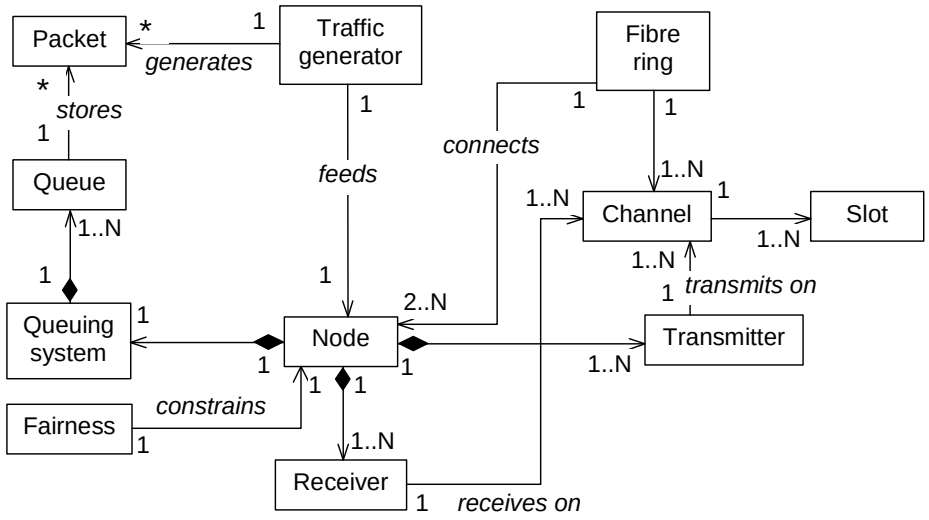


Figure 6.3.2 - UML conceptual class diagram of the simulators

The implementations of the simulators use Java. As source of randomness the simulators use the implementation of the Mersenne Twister [Matsu1998] random generator done at the European organisation for nuclear research (CERN). Mersenne Twister has a cycle of 2^{19937}-1 and virtual randomness in up to 623 dimensions, which ensures randomness integrity for very long simulation runs [Pawli2002].

6.4 Traffic characteristics

Traffic characteristics are paramount in any performance evaluation work. The parameters that describe the traffic characteristics are the packet arrival process, the packet lengths, and the traffic distribution.

Internet traffic characterization studies show that the Internet traffic has the following characteristics:

- Traffic distribution: mainly because of the world wide web (WWW), traffic distribution is often asymmetric (also known as unbalanced or non-uniform) [Balak2001], and sometimes it can be considerably volatile;
- Traffic arrival: packet arrivals occur in bursts, and such a burstiness persists over various time scales because of long-range dependence (LRD) among the arrivals [Lelan1994, Paxso1995, Thomp1997];
- Packet length: packets have variable size with peaks at 44B, 552B, 576B, and 1500B. What is more, while approximately 60% of the packets are 44B-long, 1500B-packets carry approximately half of the travelling bytes [Claff1998, Thomp1997].

Often performance evaluation works consider uniform traffic distribution, which is also known as balanced or symmetric traffic distribution. Nevertheless,

as shall be seen later in this chapter, performance results may change considerably depending on the traffic distribution. Thus, in addition to symmetric traffic distribution this work considers two other traffic distributions. The traffic distributions considered in this work are:

- Symmetric: every node generates the same workload, and packet destinations are evenly distributed;
- Asymmetric: nodes 0 to *N*-2, herein called clients, generate the same workload towards node *N*-1, herein called server, totalling 50% of the offered load. Node *N*-1 generates 50% of the traffic towards the other nodes, evenly;
- Worst case: nodes 0 to *N*-2 generate the same workload towards node *N*-1.

To evaluate the performance potential of the fairness protocols independently of the access control protocols proposed in this work the simulations assume packet size and slot size to be the same. To evaluate the performance characteristics of the access control protocols in conjunction with the fairness protocols, the simulations use the packet size distributions shown in Table 6.7-1; the table follows the measurements in [Claff1998, Thomp1997].

Traffic characterization studies show that self-similar traffic models Internet traffic better than Poisson because it captures LRD among arrivals; in Poisson packets inter-arrival times are independent and exponentially distributed.

A stochastic time series process is self-similar with the Hurst parameter *H* if the process has constant mean and finite constant variance, and the corresponding aggregated process have the same correlation functions as the original process or agree asymptotically with the correlation functions as the original process over large intervals. The Hurst parameter represents the degree of self-similarity; a value of approximately 0.5 represents the absence LRD, as in Poisson traffic, and values above 0.7 represent high LRD and high burstiness, with both increasing as *H* increases.

Nevertheless, some recent studies [Cao2001, Cao2002a, Cao2002b] argue that in high-speed links packet arrivals tend to Poisson and packet sizes tend toward independence as the load increases as a result of multiplexing gains. Also, simulation results of a destination removal MOPS ring using both Poisson traffic and self-similar traffic with $H = 0.95$ [Bengi2002] show that the impacts of such traffics on the mean queuing delay as a function of the aggregate throughput are similar. Similar results are also shown in [Rodel1999] for a multi-channel Ethernet network with $H = 0.8$. It should be noted though that both studies use the same fractal model to describe self-similar traffic: the superposition of fractal renewal processes (Sup-FRP) [Ryu1996]; the same findings might not apply to other fractal models.

Given the explanations above and because Sup-FRP is the only existing fractal model that generates inter-arrival times -the other fractal models generate arrival counts only, this work uses Poisson traffic only.

6.5 Performance parameters of interest

Amongst several performance parameters of interest, this chapter concentrates on the following ones:

- Network aggregate throughput;
- Throughput fairness; and
- Average network packet waiting time.

Network aggregate throughput is the sum of the throughput achieved by each node. Given an observation period, the throughput of a node is given by the ratio between the sum of the size of each packet successfully transmitted by that node and the capacity made available to that node.

Throughput fairness measures how fair the network really is concerning node throughput. The calculation of the throughput fairness degree depends on the traffic pattern.

In the symmetric traffic pattern the throughput fairness degree is given by the ratio between the minimum node throughput and the maximum node throughput. Let P be the vector containing the throughput of each node. The throughput fairness degree is given by

$$F = \frac{\min(P)}{\max(P)}$$

In the worst-case traffic pattern, the throughput fairness degree is given by the ratio between the minimum node throughput and the maximum node throughput, whereas only nodes 0 to N-2 are considered.

In the asymmetric traffic pattern, the throughput fairness degree is given by the ratio between the throughput of node N-1 and the sum of the throughput of all the other nodes. Let P_{N-1} be the throughput of node N-1 and P_i, for $i = 0,\ldots, N$-2, be the throughput of node i. The throughput fairness in the asymmetric traffic scenario is given by

$$F = \frac{P_{N-1}}{\sum_{i=0}^{N-2} P_i}$$

Regardless of the traffic pattern and its corresponding throughput fairness degree calculation, the closer to 1 the throughput fairness degree, the fairer the network is.

Average network packet waiting time is the mean of the average packet waiting time of each node. The average packet waiting time at a node is the mean of the waiting times experienced by the all packets transmitted by that node. Packet waiting time is the total time spent by a packet at a node, that is, the time difference from the arrival of a packet at a queue and the complete transmission

of the packet. In other words, packet-waiting time is the sum of time spent by that packet in the queue plus the time taken for that node to gain access to the medium plus the time necessary to transmit that packet completely.

6.6 Performance evaluation of the fairness protocols

This section analyses the potential performance of GCR and LCR and compares their performance with the performance of SAT. Specifically, performance metrics of interest are aggregate throughput and throughput fairness.

To make it possible to analyse the performance characteristics of the fairness protocols the simulations consider a general slotted ring with destination removal in which packets fit exactly in the slots, including the fairness control packets. Therefore, assuming that each request is 4B-long (that is, the standard size of float type data) and given that in LCR the control packet carries N^2 requests, where N denotes the number of nodes, the slot size has to be equal to $16^2 \times 4\text{B} = 1024\text{B}$, for a maximum of 16 nodes.

For the same reason, the simulations consider a single channel operating at 10Gb/s. The use of a single channel removes the influence of transceiver configurations and facilitates the analysis.

To understand the effects of the network parameters on the chosen performance metrics the simulations consider variable ring lengths and number of nodes.

The simulations consider the traffic distribution scenarios described previously, all under saturated condition; simulation of saturated traffic condition allows for the determination of the ability of each protocol to provide fair access.

For the selection of packets for transmission the simulations use the longest queuing delay packet scheduling. A node always selects the HOL packet with the longest queuing time that meets the access constrains of the fairness protocol under question.

Because of the use of saturated traffic workloads, the simulations use number of transmitted packets as stop condition. The execution of a simulation stops only after hundred thousand packets have been transmitted from each queue (actually receiving traffic).

Table 6.6-1 summarizes the parameters used in the simulations.

Table 6.6-1 – Input parameters used in the simulation

Parameter	**Value(s)**
Ring length	10km, 100km
Nr. of nodes	4, 16
Nr. of channels	1
Channel bit rate	10Gb/s
Slot size	1024B
Packet size	Slot size
$l = k$	Nr. of slots
Traffic workload	Saturated
Traffic distribution	Symmetric, asymmetric, and worst case

The performance evaluation is divided into three subsections, each focusing on one traffic distribution scenario.

6.6.1 Symmetric scenario

Figure 6.6.1 shows the network throughput against variable ring lengths and number of nodes for each of the protocols.

Figure 6.6.1 shows that LCR and SAT achieve throughputs close to the maximum, whereas with 16 nodes LCR achieves the highest throughputs; SAT achieves the highest throughputs with 4 nodes; GCR achieves the worst performances, and like SAT its achievable throughputs also degrade with 16 nodes.

The reason why LCR benefits from a high number of nodes and SAT and GCR benefit from a low number of nodes lies in the conception of these protocols.

In the three protocols the competition for the medium tends to increase with the number of nodes. Consequently, more bottlenecks tend to occur simultaneously. Nevertheless, in LCR the occurrence of a bottleneck on a given link does not affect the other links; therefore, nodes can exploit the traffic locality that the resulting increase in the number of links offers. In SAT and GCR nodes see the occurrence of a bottleneck on a given link as spreading over the entire network, and, hence, cannot profit from the traffic locality. Consequently, their performances suffer.

Both GCR and LCR achieve better performances in 100km-rings. That is because both protocols use payload slots to transmit the fairness control packet, but the transmission of the fairness packet does not account for throughput, only for fairness. Therefore, with the 10km-ring there are less slots and the slots used to transport the fairness packet represent a greater fraction of the ring capacity compared with the 100km-ring.

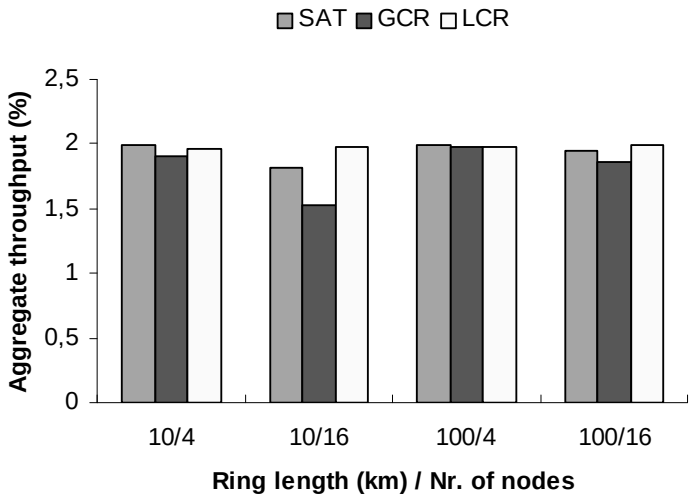


Figure 6.6.1- Aggregate throughput under the symmetric saturated traffic condition

Note that LCR never achieves the maximum throughput because of the mapping between the calculated fair rate and the corresponding transmission quota. To calculate the transmission quota of a node, LCR multiplies the calculated fair rate by the capacity of the ring. Nevertheless, fair rates may contain fractions, but it is not possible to allocate fractions of slots. Consequently, there is waste of capacity, and such a waste becomes worse as the number of nodes increases.

GCR suffers from the same problem as LCR, and that explains why the throughput achieved by GCR is worse than the throughput achieved by SAT.

Figure 6.6.2 shows the throughput fairness degree as both the ring length and the number of nodes vary. The three protocols achieve high degrees of fairness, with GCR achieving the highest and LCR achieving the lowest overall. GCR and SAT achieve higher degrees of fairness than LCR because they use fairness cycles of variable lengths. Hence, every node exhausts its transmission quota before a new cycle begins. In LCR fairness cycles are of fixed length. For this reason and because packet scheduling is not round-robin, some small traffic distribution asymmetries may happen during a fairness cycle, and when they do the degree of fairness degrades. Nevertheless, the degradation is negligible.

Typical of destination removal slotted rings, the nodes closer to the node that generates the fairness control packet or the SAT signal for the first time have a slight advantage over the other nodes, in particular when the number of nodes is small (see Figure 6.6.3). Given a certain capacity advantage, the more nodes there are in the network, the smoother the advantage of the first nodes is since there are more nodes to share the capacity advantage.

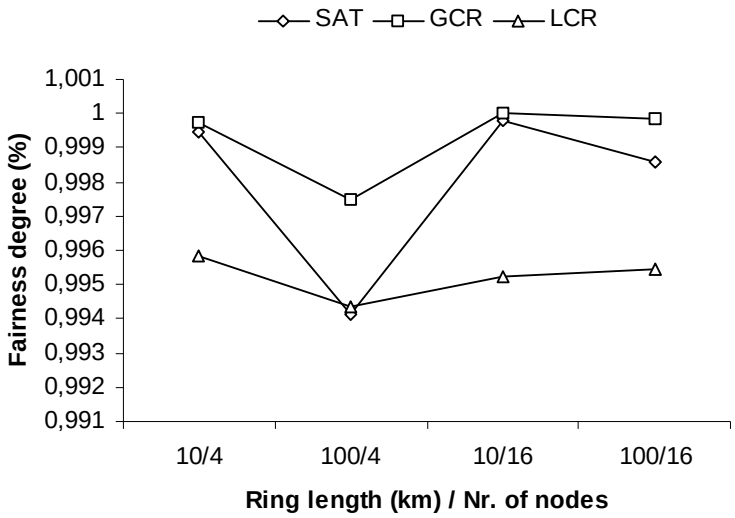


Figure 6.6.2 - Degree of throughput fairness under the symmetric saturated traffic condition.

Furthermore, the capacity advantage increases with the fairness cycle length, which explains why SAT suffers the most the influence of the ring length -the simulations assume transmission quotas to equal the ring length.

Figure 6.6.3 shows the per node throughput in a 100km-ring with 4 nodes.

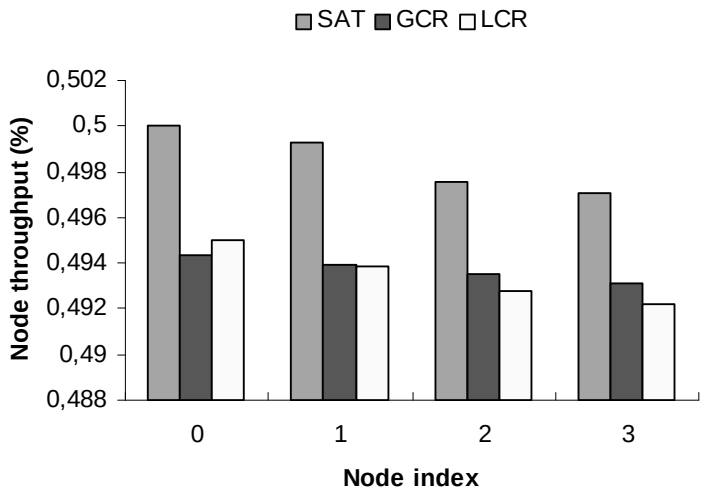


Figure 6.6.3 - Per node throughput in 100km-ring with four nodes under the symmetric saturated traffic condition.

6.6.2 Asymmetric scenario

Figure 6.6.4 shows the network throughput against varying ring lengths and varying number of nodes. The figure highlights the performance advantage of LCR over both SAT and GCR. As the traffic distribution becomes asymmetric it becomes even more important to exploit locality, and LCR does just that.

SAT and GCR cannot exploit traffic locality, and because all the nodes are assigned the same transmission quota, and because the nodes use their transmission quotas entirely, SAT and GCR reduce the achievable throughput of node *N*-1 to that of the other nodes. Consequently the throughput suffers.

As with the symmetric condition, the aggregate throughputs of both GCR and SAT degrade as the number of nodes increases. The reasons are the same as in the symmetric scenario.

Note that the number of nodes affects the aggregate throughput of all the three protocols. That is because under the asymmetric traffic distribution the fairness protocols have to force nodes to back off with more intensity as the number of nodes increases to guarantee fairness.

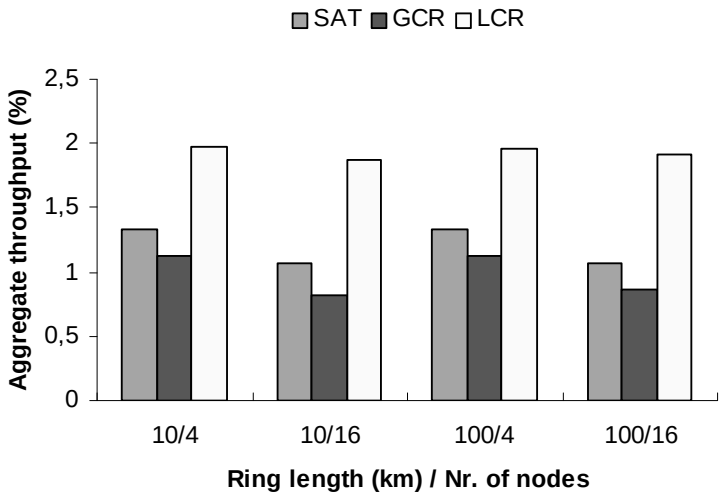


Figure 6.6.4 - Aggregate throughput under the asymmetric saturated traffic condition

Figure 6.6.5 shows the degree of throughput fairness obtained in the three protocols as the number of nodes and the ring length vary. The curves make it clear that only LCR is able to achieve high performances at high degrees of fairness under asymmetric traffic distributions. Because SAT and GCR reduce the achievable throughput of node *N*-1 to that of the other nodes, both protocols achieve the same poor degrees of fairness.

The effects of the number of nodes on the degree of throughput fairness achieved by all the three protocols can be seen in Figure 6.6.5. The deterioration of throughput fairness as the number of nodes increases shows that it is more difficult to ensure fairness under the asymmetric traffic distribution.

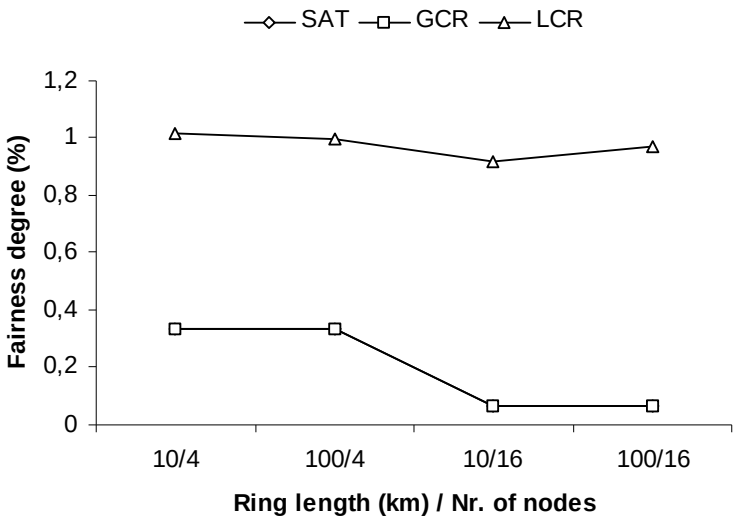


Figure 6.6.5 - Degree of fairness throughput under the asymmetric saturated traffic condition. Note that the SAT curve is hidden behind the GCR curve.

6.6.3 Worst-case scenario

Figure 6.6.6 shows the network throughput against varying ring lengths and varying number of nodes. As it can be seen, SAT achieves the best throughput, followed by LCR and, then, GCR.

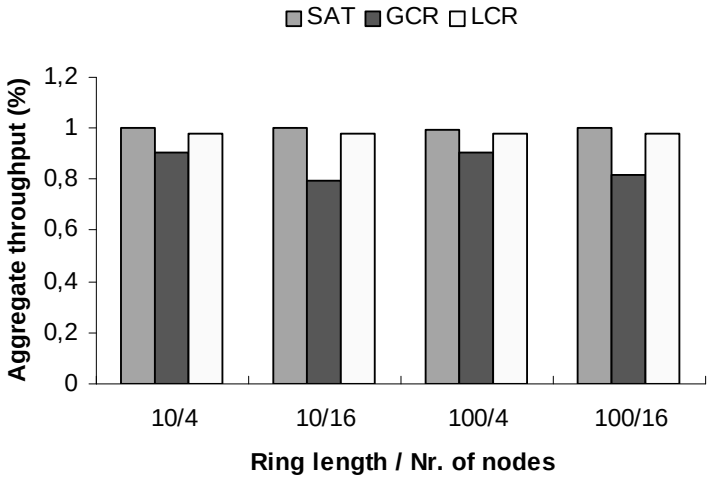


Figure 6.6.6 - Aggregate throughput under the worst-case saturated traffic condition

Both SAT and LCR achieve throughputs close to 1, which is the maximum achievable throughput. Nevertheless, LCR falls slightly behind SAT because it uses payload slots to transmit fairness control packets and, besides, it looses some capacity as a consequence of the mapping between the calculated fair rate and the allocated transmission quota, as explained previously.

Note that the curves have the same characteristic as the curves corresponding to the symmetric traffic condition, and for the same reasons.

Figure 6.6.7 shows the degree of throughput fairness obtained in the three protocols as the number of nodes and the ring length vary. The curves show that overall the three protocols achieve high degrees of fairness, and that the differences in the degrees are negligible.

As in the asymmetric traffic scenario, and for the same reasons, LCR provides the highest degree of throughput fairness.

Note that the first-nodes-advantage phenomenon detected in the symmetric scenario happens in the worst-case scenario too, and the arguments given to explain the phenomenon previously still hold.

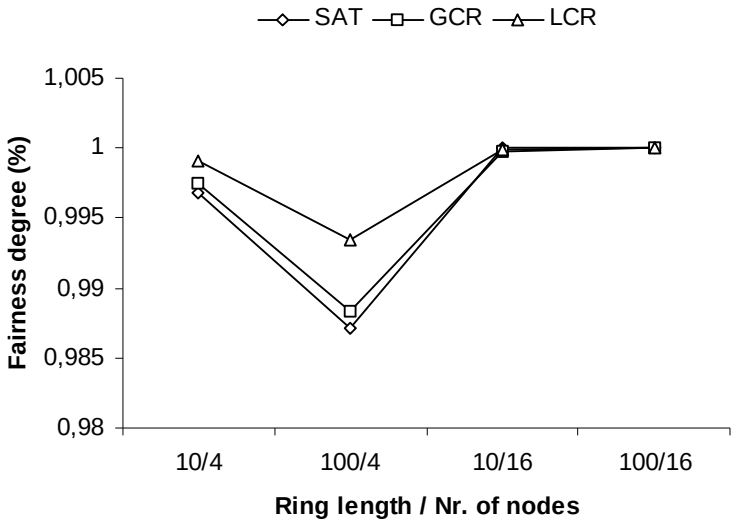


Figure 6.6.7 - Degree of throughput fairness under the worst-case saturated traffic condition.

6.7 Performance evaluation of the access control protocols

This section analyses the performance of the proposed access control protocols and the influence of SAT and LCR on the performance of those access protocols; GCR is not considered because it achieves the worst performances.

With the exception of PAT, all the simulations use 128B-slots. Such a size has been chosen because test simulations using slots of 64B, 128B, and 256B show that although 64B-slots offer the highest performance, and the performance differences between networks using 64B-slots and 128B-slots are small, the

processing overhead caused by 64B-slots is twice as much that caused by 128B-slots.

The simulations of PAT assume 4 kilobyte(KB)-slots. Again, such a size has been chosen because test simulations using slots of 2KB, 4KB, and 8KB show that 4KB-slots offer the best ratio between performance and processing overhead.

Table 6.7-1 shows the packet size distributions used in the simulations of the access control protocols. The distribution is based on that presented in [Claff1998, Thomp1997] with the difference that 20B are added to 44B- and 576B-packets to express IPv6 traffic –the IPv6 base header contains 40B rather than the 20B of the IPv4 header.

Table 6.7-1 - Packet size distribution adopted in the simulations

Packet size (Bytes)	**Probability (%)**
64	60
596	15
700	5
800	5
1100	7
1500	8

For the selection of packets for transmission the simulations use the longest queuing delay packet scheduling. A node always selects the HOL packet with the longest queuing time that meets the access constrains of the fairness protocol under question.

To remove the effects of transceiver configuration in the results, and, hence, show the potential of the protocols, the simulations consider a single-channel ring.

To understand the effects of the network parameters on the performance of the protocols, the simulations consider variable ring lengths and number of nodes. To understand the effects of traffic patterns on the performance of the protocols the simulations consider two traffic distribution scenarios: symmetric and asymmetric.

To analyse the access control protocols, and compare them, the discussions are divided into two parts: one considering SAT and one considering LCR.

All the results satisfy the 95% confidence interval, which is calculated over the average packet waiting times calculated by the nodes.

Table 6.7-2 summarizes the parameters used in the simulations.

Table 6.7-2 - Input parameters used in the simulations

Parameter	Value(s)
Ring length	10km and 100km
Nr. of nodes	4 and 16
Nr. of channels	1
Channel bit rate	10Gb/s
Slot size	4KB for PAT 128B for the others
$l = k$	Nr. of slots
THT	Nr. of slots / nr. of channels
Traffic workload	20%, 40%,…, 160%, 180%
Traffic distribution	Symmetric and asymmetric

6.7.1 SAT

Aggregate throughput under the symmetric traffic distribution

Figure 6.7.1 to Figure 6.7.4 show the aggregate throughputs achieved by the four protocols combined with SAT under the symmetric traffic distribution.

The curves show that SPT+P, PAT, and SPT+SC achieve aggregate throughputs that approximate 160% of the nominal capacity of the ring. Such high throughputs demonstrate that the protocols indeed achieve a spatial reuse factor of two approximately, considering that the slots carry considerable padding.

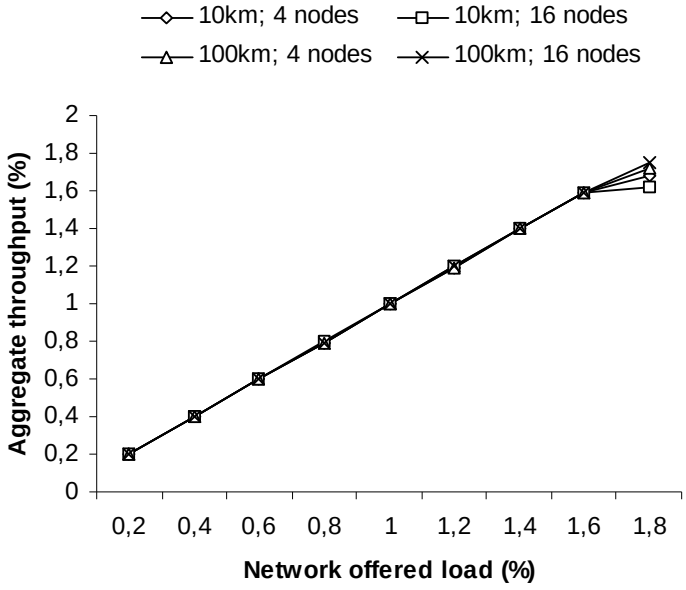


Figure 6.7.1- Aggregate throughput in SPT+P under SAT and the symmetric traffic condition

As it can be seen from the curves, the ring lengths and the number of nodes do not influence the aggregate throughputs as long as the offered load is below the saturation point. Above that point, longer rings achieve a slightly better performance because of the higher access opportunities caused by the greater number of slots.

The effects of the number of nodes on the aggregate throughput of SPT+R are evident. That is because contention increases with the number of nodes, and as contention increases more capacity is wasted with slots containing payload data that will be discarded at destination nodes for being incomplete. Furthermore, more retransmissions are required, and retransmissions in turn increase the chances of contention.

What is more, since SPT+R is a persistent protocol, which means that nodes keep trying to transmit a given packet until they succeed, contention may become severe to the point at which all nodes are prevented from transmitting. For this reason and because of the consequent long time necessary for the simulations to converge, the curves corresponding to 16 nodes do not go beyond 60% of offered load, the maximum achievable throughput.

The curves also show some influence, although minor, of the ring length on the achievable throughputs of SPT+R. That is because of the correspondence between ring length and number of slots. As the ring length increases nodes have more access opportunities, and this alleviates contention slightly.

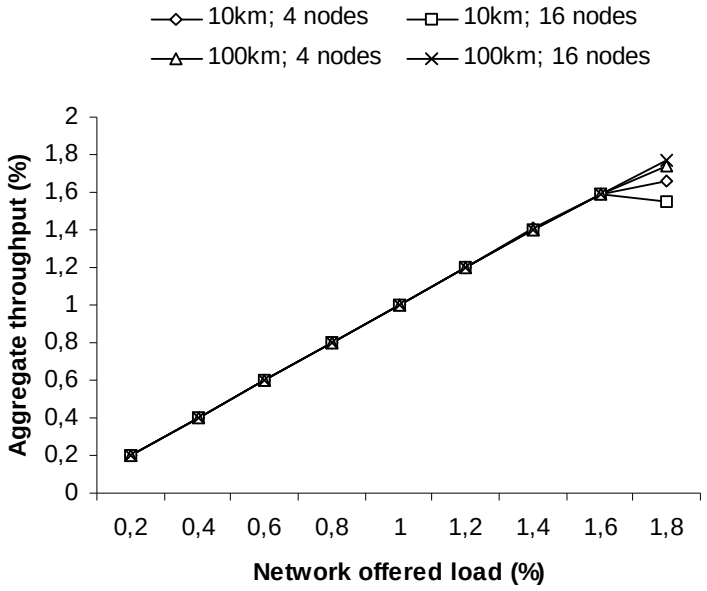


Figure 6.7.2 - Aggregate throughput in PAT under SAT and the symmetric traffic condition

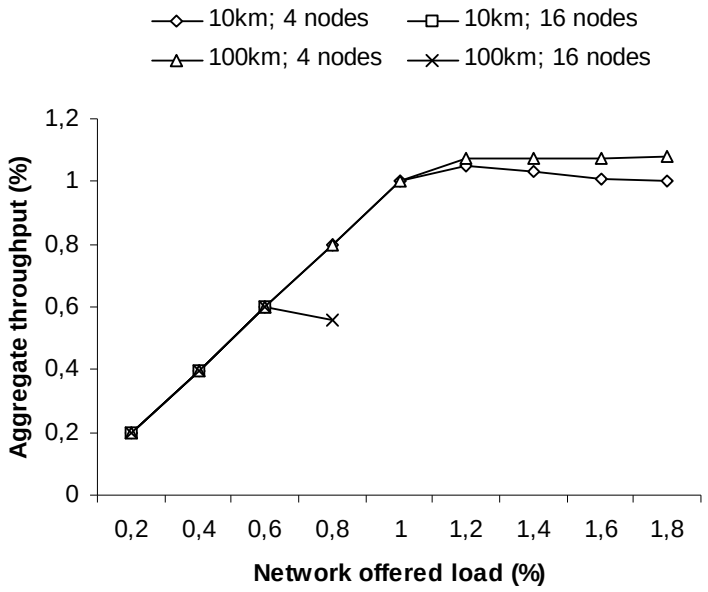


Figure 6.7.3 - Aggregate throughput in SPT+R under SAT and the symmetric traffic condition

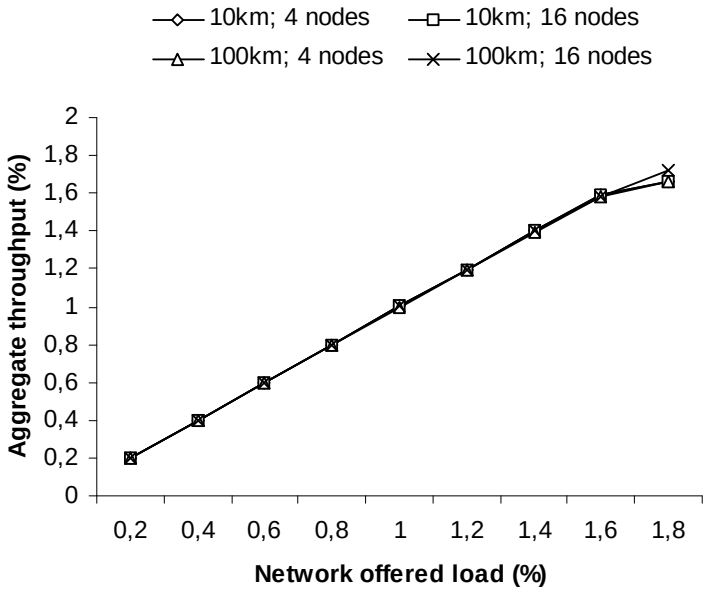


Figure 6.7.4 - Aggregate throughput in SPT+SC under SAT and the symmetric traffic condition

Throughput fairness under the symmetric traffic distribution

Figure 6.7.5 to Figure 6.7.8 show the degree of throughput fairness provided by the four protocols combined with SAT under the symmetric traffic distribution.

From the curves it can be concluded that all the protocols achieve high degrees of throughput fairness, which is expected since the traffic distribution is symmetric.

Slight variations in the degrees of fairness can be seen in the curve representing a 100km-ring with four nodes. That is because the nodes closer to the node that inserts the SAT signal for the first time get a slight throughput advantage over the other nodes.

As explained before, the smaller the number of nodes, the greater the advantage of the first nodes are, in particular with a great number of nodes since the first nodes have more slots to profit from; as shown in [Zafir1988], the correlation among node activities decreases as the number of nodes increases, hence relaxing the effects of SAT slightly.

The degree of fairness degrades as the load increases because at low loads the first nodes let slots go empty because they are idle. At high loads though, the first nodes use such slots, and the other nodes suffer from unfairness.

Nevertheless, the variations in the degree of fairness are negligible.

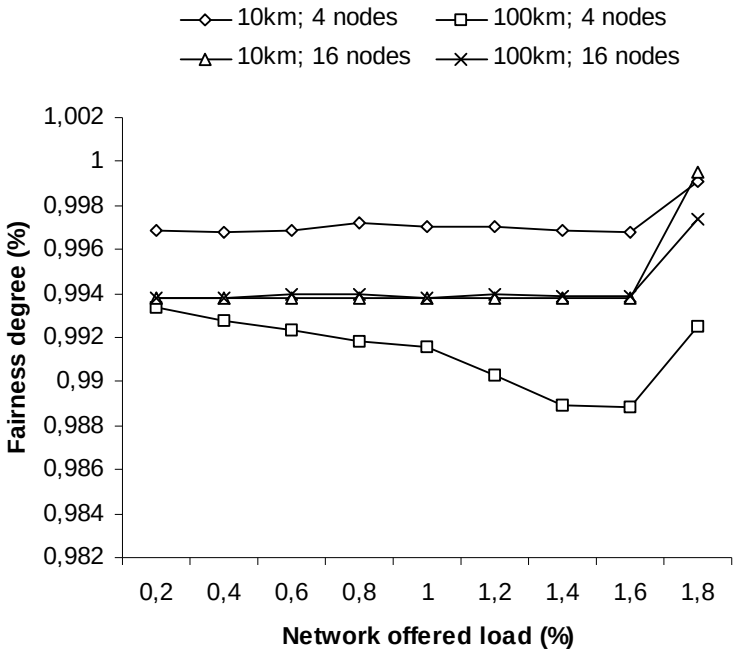


Figure 6.7.5 - Degree of throughput fairness in SPT+P under SAT and the symmetric traffic condition.

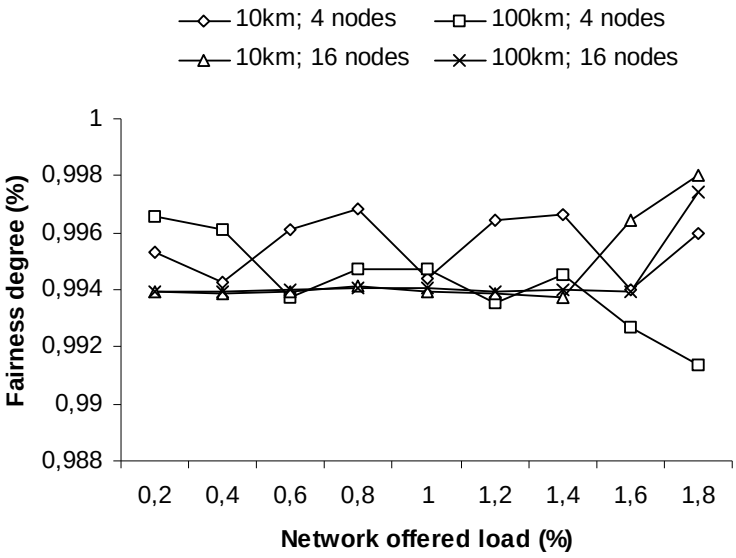


Figure 6.7.6 - Degree of throughput fairness in PAT under SAT and the symmetric traffic condition.

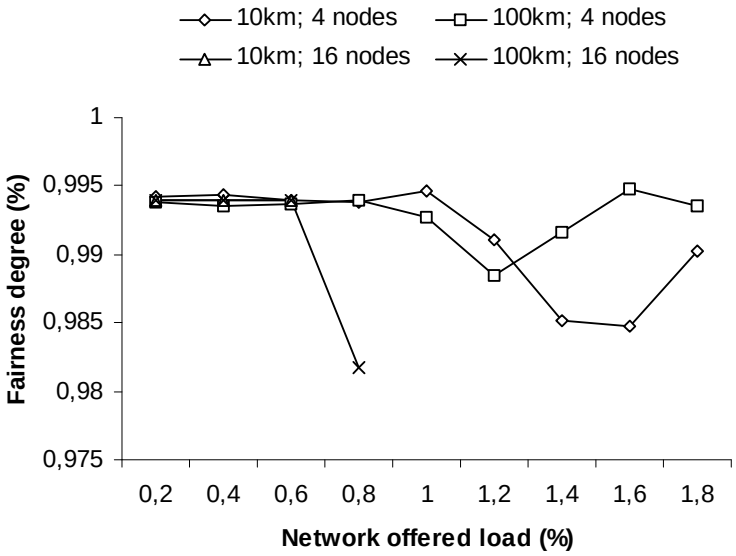


Figure 6.7.7 - Degree of throughput fairness in SPT+R under SAT and the symmetric traffic condition.

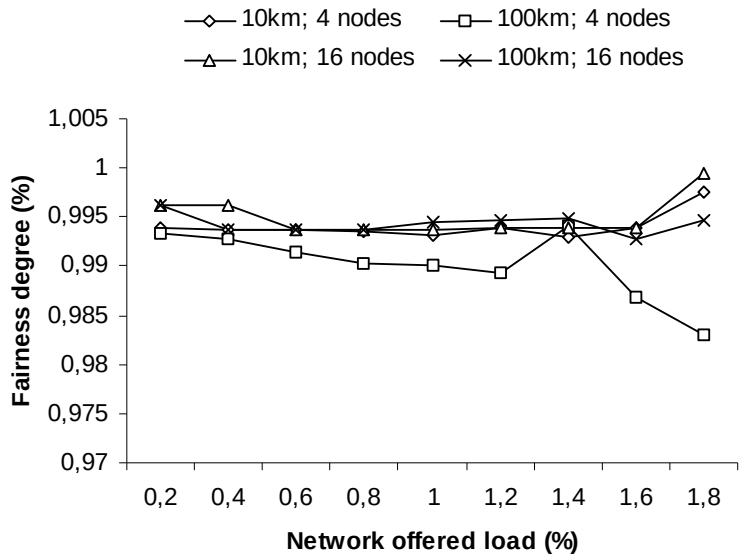


Figure 6.7.8 - Degree of throughput fairness in SPT+SC under SAT and the symmetric traffic condition.

Average network packet-waiting time under the symmetric traffic distribution

Figure 6.7.9 shows the average network packet waiting times under each access control protocol as a function of the network offered load for a 100km-ring with 4 nodes. Figure 6.7.10 shows the average network packet waiting times as a function of the network offered load for a 100km-ring with 16 nodes.

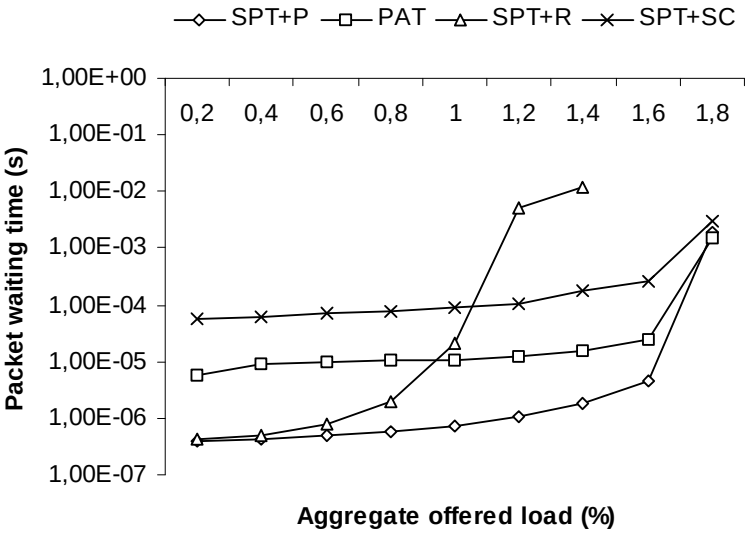


Figure 6.7.9 - Average network packet-waiting time (in log. scale) in a 100km-ring with 4 nodes under SAT and the symmetric traffic condition.

SPT+P achieves the lowest packet waiting times thanks to the exploitation of the fragmentation of the network capacity. Since the difference between SPT+P and SPT+R lies in the contention resolution, under low loads SPT+R achieves packet-waiting times that are as low as those of SPT+P. As the load increases though, the effects of contention on the packet waiting times experienced in SPT+R become evident.

Both PAT and SPT+SC achieve nearly flat packet waiting times, although in PAT waiting times are lower. The packet waiting times in PAT are nearly flat because as the load increases nodes start to use the capacity of the slots they allocate more efficiently, and doing so compensates for steady network utilisation; network utilisation is the ratio between the sum of the total number of traversed links by all the busy slots and the total number of links traversed by all the slots regardless of their status.

The packet waiting times in SPT+SC are nearly flat because as the load increases nodes start to use more of the slot trains they generate. Besides, even at low loads nodes are constrained by the transmission rules of SPT+SC.

With the exception of SPT+P, an increasing number of nodes has a negative effect on the packet waiting times. In particular, SPT+R suffers the most as a consequence of higher contention probability.

The reason why PAT experiences higher packet waiting times with an increasing number of nodes is that the ratio between number of slots and the number of nodes becomes smaller. When such a ratio becomes considerably small it takes longer for a node to find an empty slot, in particular because each node has a transmission quota that equals the number of slots

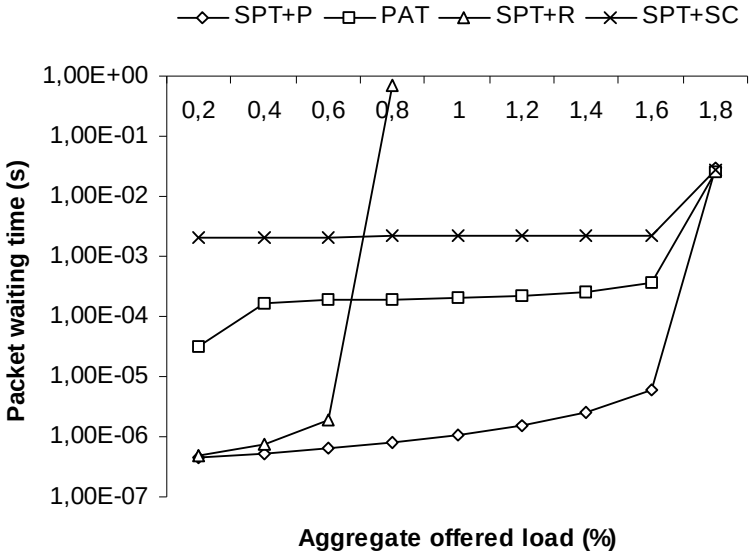


Figure 6.7.10 - Average network packet-waiting time (in log. scale) in a 100km-ring with 16 nodes under SAT and the symmetric traffic condition.

SPT+SC experiences the highest packet waiting times because there is a single token, and as the number of nodes increases it takes longer for a node to get unconstrained access to the ring.

It should also be noted that, as discussed previously, the performance of SAT degrades as the number of nodes increases because of the effects of local bottlenecks on the ring entire ring.

Aggregate throughput under the asymmetric traffic distribution

Figure 6.7.11 to Figure 6.7.14 show the aggregate throughputs achieved by the four protocols combined with SAT under the asymmetric traffic distribution.

With the exception of SPT+P, the performance of the protocols drops considerably under the asymmetric scenario, in particular as the number of nodes increases. SPT+P achieves high aggregate throughputs and suffers less from an increasing number of nodes. That is thanks to the ability of SPT+P to exploit the fragmentation of the capacity of the medium, which is more intense under the asymmetric traffic condition than under the symmetric traffic condition, in particular with high number of nodes.

The 4-node curves in Figure 6.7.12 can be explained as follows. At low loads PAT gains little from multiplexing packets inside slots. Consequently, nodes have to allocate more slots to cope with the load, resulting in a network utilisation of almost 100% at a 40%-offered load.

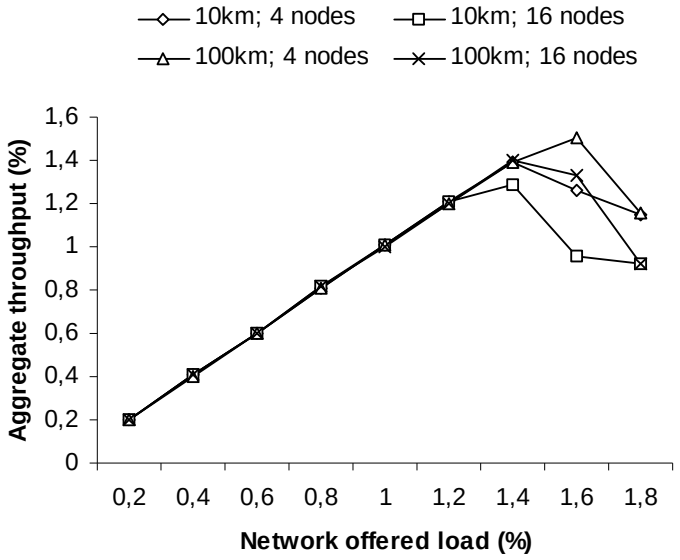


Figure 6.7.11 - Aggregate throughput in SPT+P under SAT and the asymmetric traffic condition

Still at low loads, multiplexing gains improve as the load increases. From a certain offered load, which corresponds to approximately 60% with 10km-rings and 80% with 100km-rings, multiplexing gains do not improve, and the client nodes start to detect starvation. As SAT acts to ensure that nodes get equal access

opportunities, the network utilisation starts to decrease, and it does so until it becomes steady. The decrease in the network utilisation stems from the server node being forbidden to transmit by SAT. As shall be seen in this chapter, such a phenomenon creates unfairness according to the definition used for the asymmetric traffic distribution.

From the point where the network utilisation becomes steady on the aggregate throughput bounces back and increases up to its maximum. That is because SAT transforms the asymmetric traffic distribution onto the symmetric traffic distribution, with the client nodes using the capacity "stolen" from the server node.

Similar phenomena explain the 16-node curves, with the exception that the client nodes detect starvation at a load slightly above 20% of the nominal capacity.

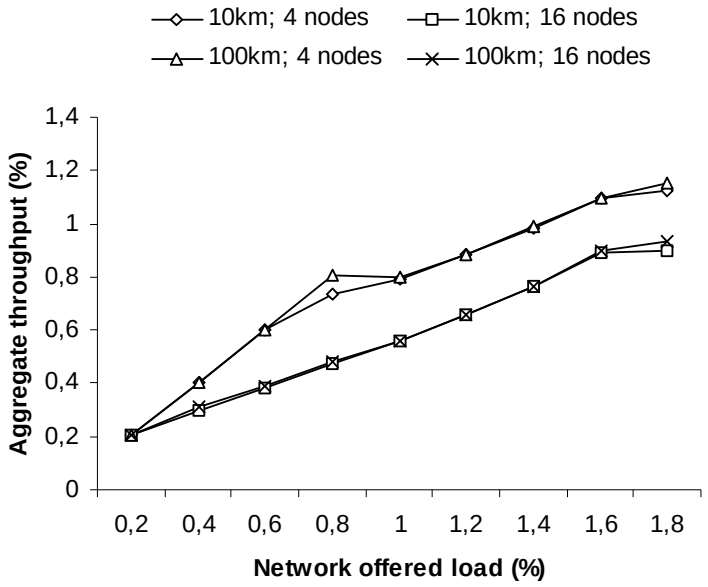


Figure 6.7.12 - Aggregate throughput in PAT under SAT and the asymmetric traffic condition

Figure 6.7.13 shows a phenomenon similar to the one just explained: a throughput decrease followed by a throughput increase. Nevertheless, in SPT+R the client nodes detect starvation at higher loads (that is, 80% with 16 nodes and 90% with 4 nodes) than do the client nodes in PAT; the performance drawback caused by the small number of slots in PAT becomes evident under the asymmetric traffic distribution.

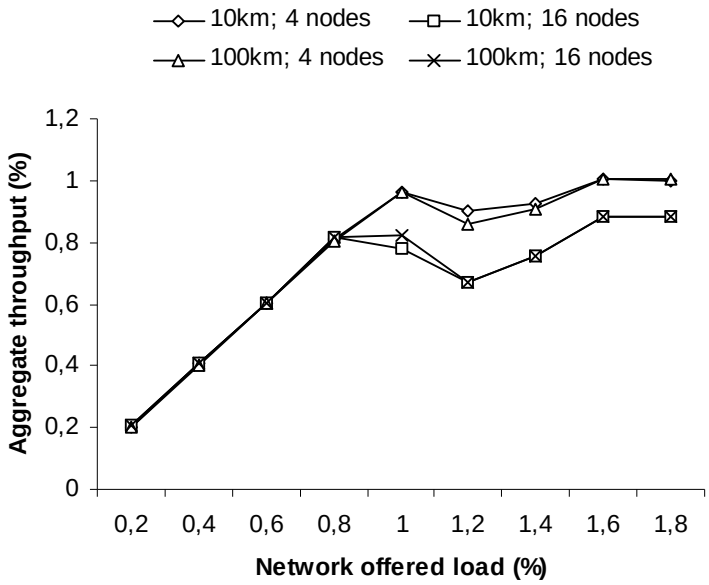


Figure 6.7.13 - Aggregate throughput in SPT+R under SAT and the asymmetric traffic condition.

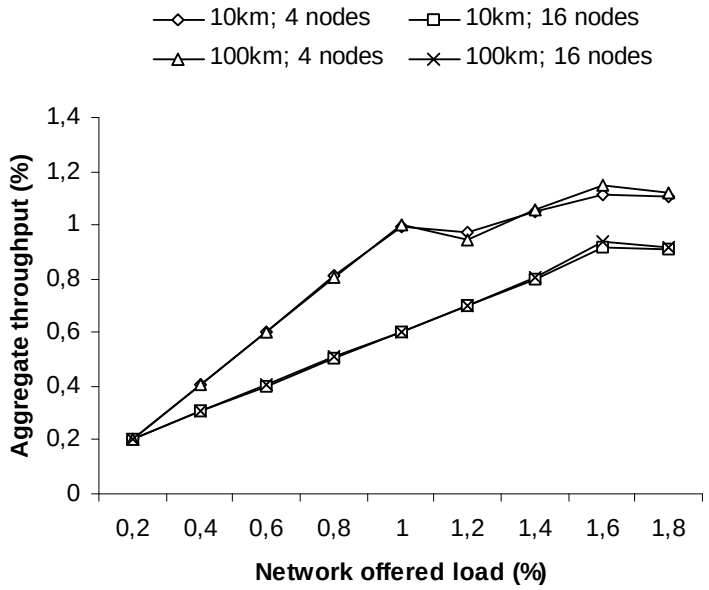


Figure 6.7.14 - Aggregate throughput in SPT+SC under SAT and the asymmetric traffic condition.

Note also that SPT+R achieves higher throughputs under the asymmetric traffic condition than under the symmetric traffic condition. That is because under

the asymmetric traffic condition access is more regulated and both fragmentation of the network capacity and, consequently, contentions are lower.

The same phenomenon of throughput decrease followed by a throughput increases occurs in SPT+SC as well, and that describes the typical behaviour of SAT under the asymmetric traffic distribution.

Throughput fairness under asymmetric traffic distribution

Figure 6.7.15 to Figure 6.7.18 show the degree of throughput fairness provided by the four protocols combined with SAT under the asymmetric traffic distribution.

The curves show that none of the protocols can guarantee fairness above the saturation point, and that is due to SAT. The protocols can guarantee fairness at lower loads because at those loads the nodes become satisfied without using their transmission quota entirely. As the load increases the nodes start to use more of their quota, until the moment where the nodes detect starvation. At this point the SAT protocol acts more strongly to assure that each node gets the share corresponding to its quota. Since all nodes are allocated the same quota, as the load increases all the nodes tend to achieve the same throughput, and that explains why the network becomes unfair. Figure 6.7.19 illustrates such a phenomenon.

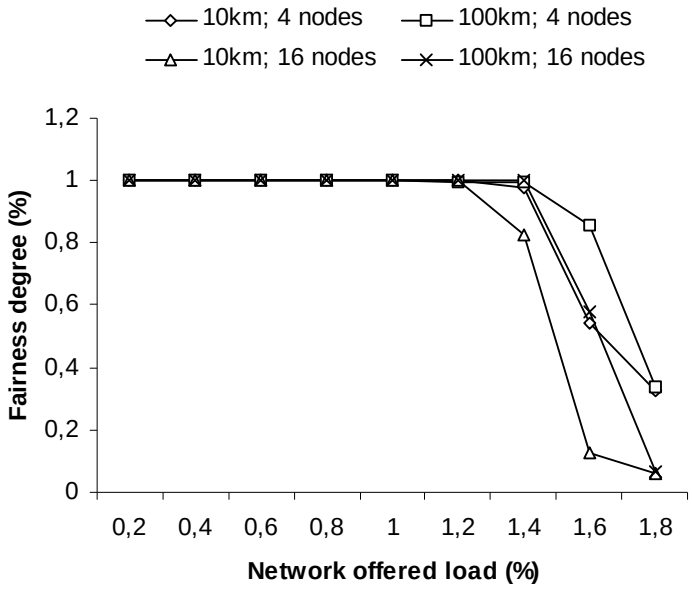


Figure 6.7.15 - Degree of throughput fairness in SPT+P under SAT and the asymmetric traffic condition.

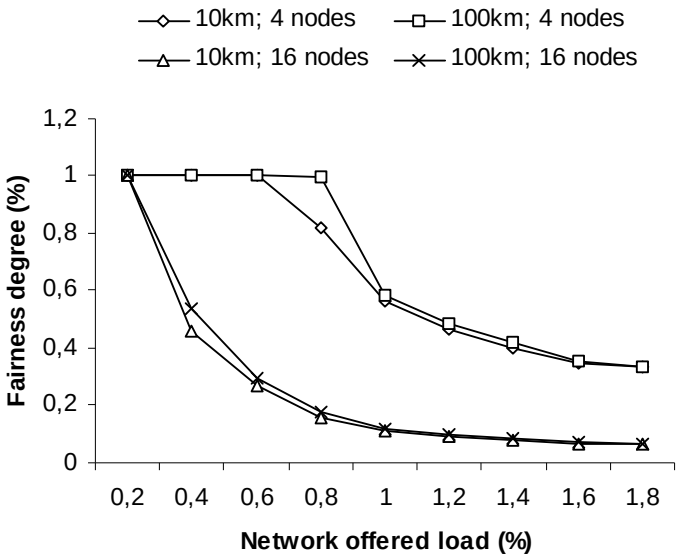


Figure 6.7.16 - Degree of throughput fairness in PAT under SAT and the asymmetric traffic condition.

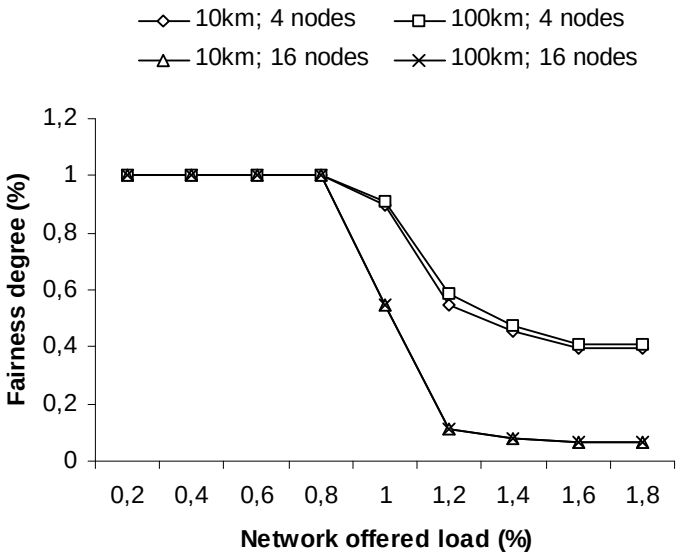


Figure 6.7.17 - Degree of throughput fairness in SPT+R under SAT and the asymmetric traffic condition.

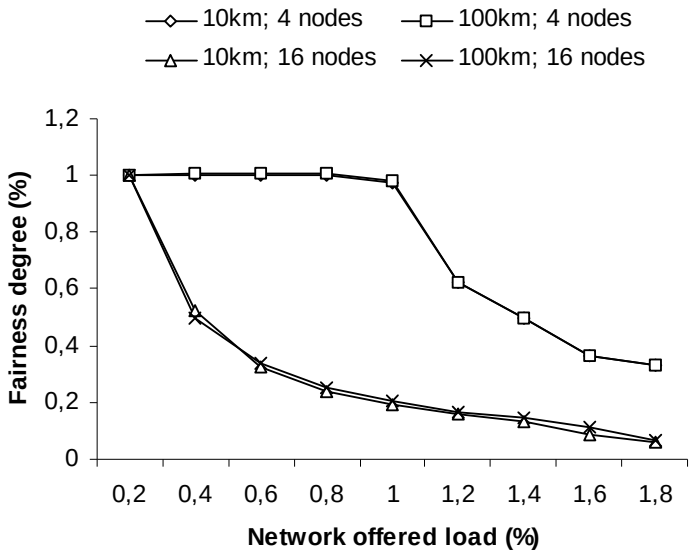


Figure 6.7.18 - Degree of throughput fairness in SPT+SC under SAT and the asymmetric traffic condition.

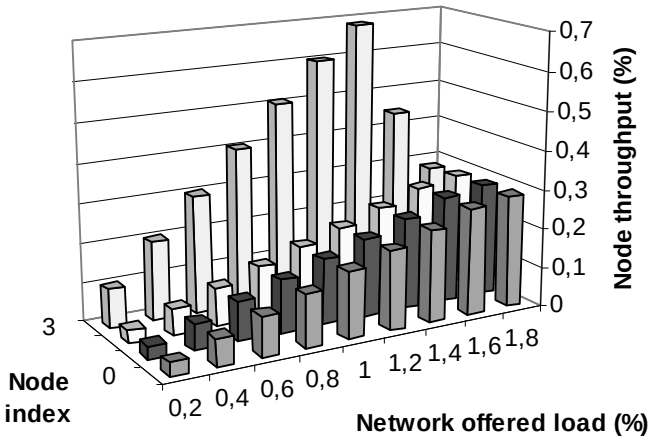


Figure 6.7.19 - Per node throughput in SPT+P in a 10km-ring with four nodes under SAT and the asymmetric traffic condition.

PAT and SPT+SC suffer the most from unfairness, but for different reasons. In PAT the cause is the small ratio between the number of slots and the number of nodes, which leads to starvation quickly as the load increases. In SPT+SC the causes are the SAT itself, as explained previously, and the implicit fairness of SPT+SC, which ensures free access whenever a node possesses a token and regardless of what SAT determines.

Average network waiting time under the asymmetric traffic distribution

Figure 6.7.20 shows the average network packet-waiting times experienced by client nodes under each access control protocol as a function of the network offered load for a 100km-ring with 4 nodes. Figure 6.7.21 shows the average network packet-waiting times experienced by the server node as a function of the network offered load for a 100km-ring with 4 nodes.

The curves show that below saturation all the protocols achieve low packet waiting times regardless of the role played by the nodes. Nevertheless, as the load increases the client nodes start to steal capacity from the server node, as explained previously, and this explains the more evident deterioration of the packet waiting times experienced by the server node compared with the deterioration experienced by the client nodes.

Figure 6.7.22 shows the average network packet-waiting times experienced by client nodes under each access control protocol as a function of the network offered load for a 100km-ring with 16 nodes. Figure 6.7.23 shows the average network packet-waiting times experienced by the server node as a function of the network offered load for a 100km-ring with 16 nodes.

The SPT+SC and PAT curves show once more the negative effects of an increasing number of nodes on the performance of these protocols. Again the server node suffers the most in both protocols. What is more, the packet waiting times experienced by the server node are too high for any use other than non-interactive bulk transfers –in interactive communications response times should not exceed the human factors limit of 100 to 200 milliseconds [Jacob1990] to be acceptable.

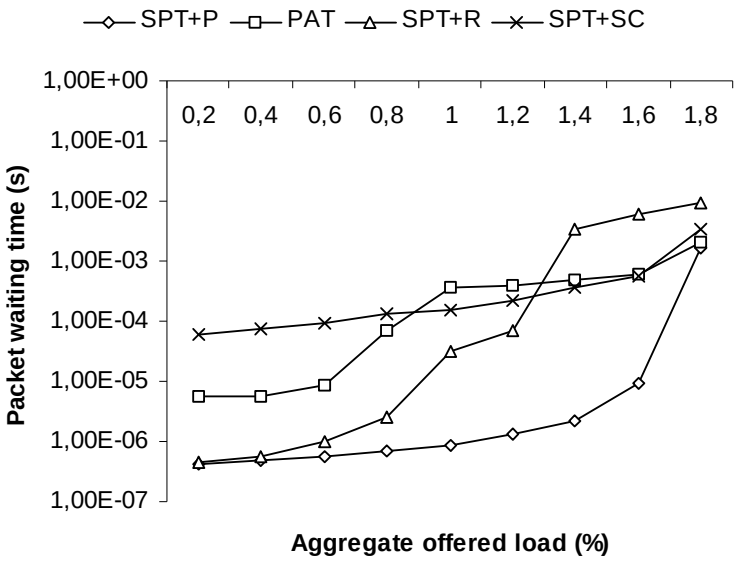


Figure 6.7.20 - Average packet-waiting time (in log. scale) experienced by the client nodes in a 100km-ring with 4 nodes under SAT and the asymmetric traffic condition.

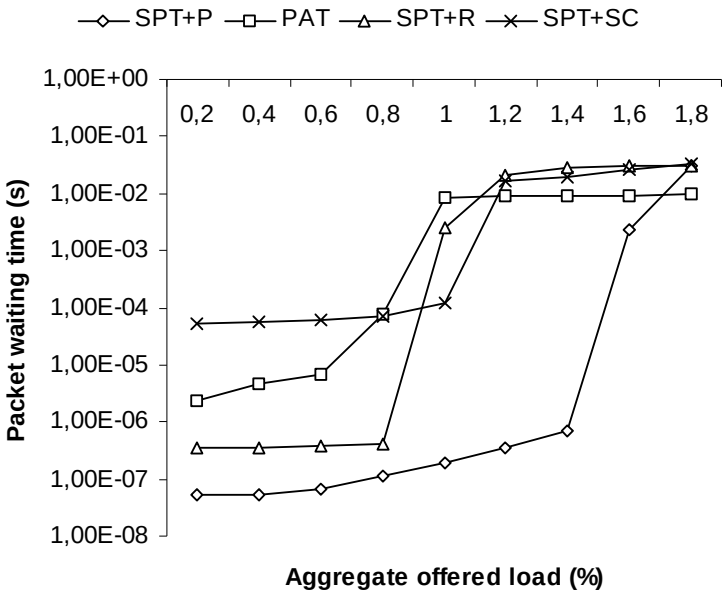


Figure 6.7.21 - Average packet-waiting time (in log. scale) experienced by the server node in a 100km-ring with 4 nodes under SAT and the asymmetric traffic condition.

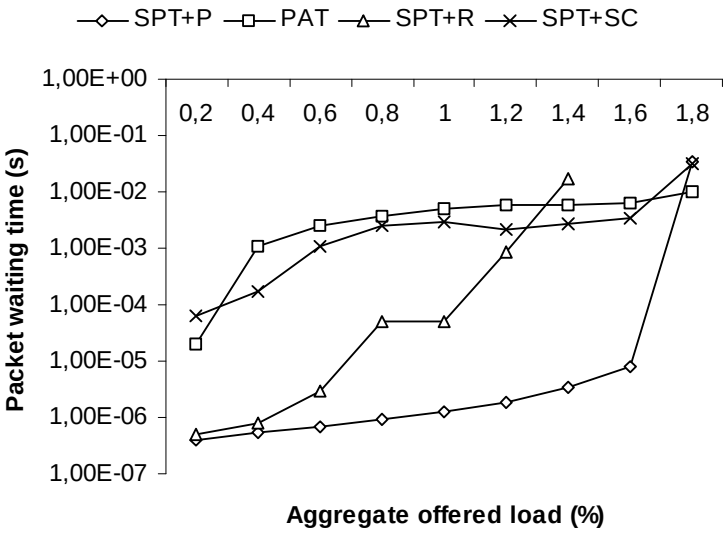


Figure 6.7.22 - Average packet-waiting time (in log. scale) experienced by the client nodes in a 100km-ring with 16 nodes under SAT and the asymmetric traffic condition.

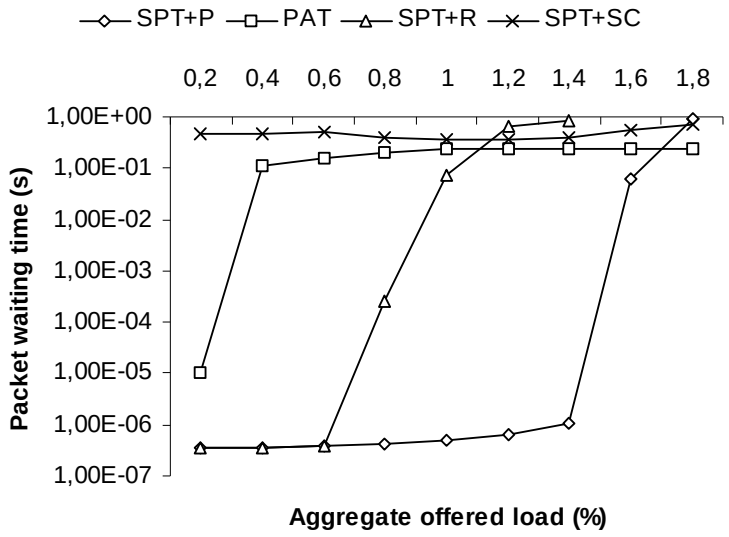


Figure 6.7.23 - Average packet-waiting time (in log. scale) experienced by the server node in a 100km-ring with 16 nodes under SAT and the asymmetric traffic condition.

6.7.2 LCR

Aggregate throughput under the symmetric traffic distribution

Figure 6.7.24 to Figure 6.7.27 show the aggregate throughputs achieved by the four protocols combined with LCR under the symmetric traffic distribution.

As the curves show, SPT+P achieves the highest aggregate throughput, and the aggregate throughput suffers influence from neither ring length nor number of nodes.

The same does not apply to the other protocols though. The aggregate throughput of both SPT+R and SPT+SC degrades as the number of nodes increases. The causes for such a degradation are i) increase in the fairness control packet size as the number of nodes increases; and ii) the difficulty of both protocols to forward large packets.

Although both protocols suffer from an increasing number of nodes, the ring length affects SPT+R and SPT+SC in opposite ways when the number of nodes is high.

Longer rings affect the throughput of SPT+SC positively because as the ring length increases *THT* also increases, and a greater *THT* results in the generation of more *MTU*-sized slot trains. Such large trains facilitate the transmission of large packets.

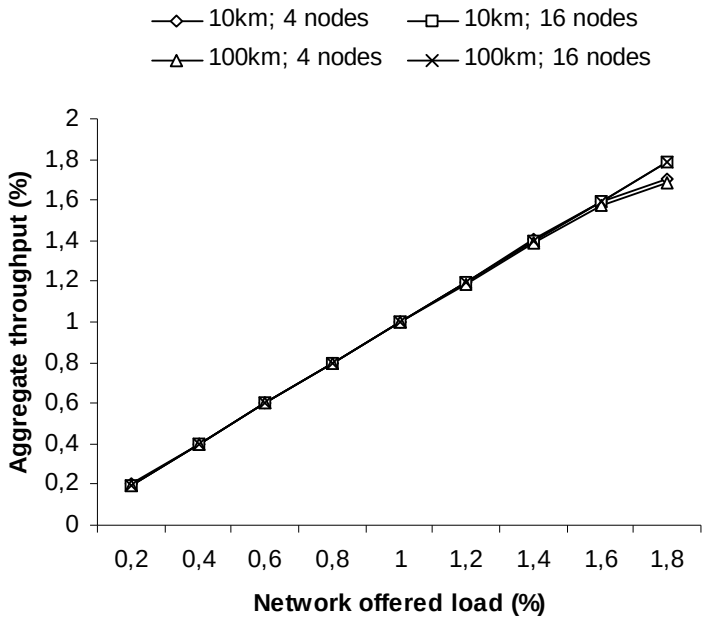


Figure 6.7.24 - Aggregate throughput in SPT+P under LCR and the symmetric traffic condition

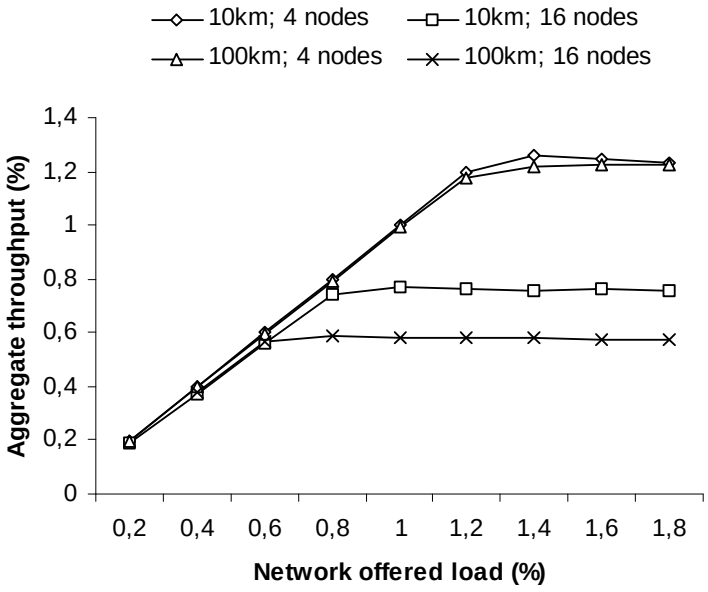


Figure 6.7.25 - Aggregate throughput in SPT+R under LCR and the symmetric traffic condition

Shorter rings affect the throughput of SPT+R positively because with shorter rings there is a stronger correlation between the node activities. That is, LCR acts more intensively, and nodes receive a higher number of consecutive empty slots.

Consequently, contention, retransmission, and network utilisation decrease, and the aggregate throughput increases.

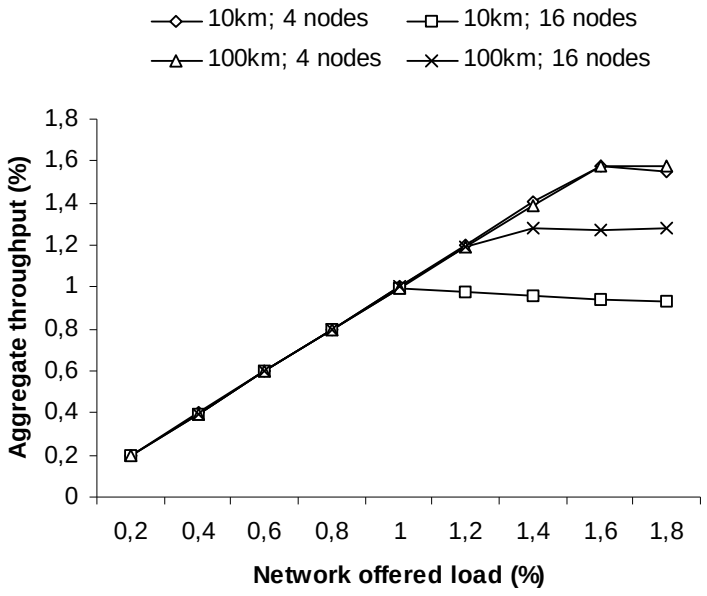


Figure 6.7.26 - Aggregate throughput in SPT+SC under LCR and the symmetric traffic condition

Figure 6.7.27 shows the effects of the number of nodes and, in particular, the ring length on the aggregate throughputs that PAT can achieve. Specifically, the throughput of PAT improves as the ring length increases and degrades as ring length decreases. A decreasing number of nodes helps improve the throughput too, but to a lesser extent.

The cause for such behaviour is the combination of large slot sizes with the transmission quota allocation mechanism. The fair rates calculated by LCR may be fractions, in particular as the number of nodes increases, and the multiplication of such fractions by the ring length capacity to determine the transmission quota may result in the waste of one slot per node. Therefore, the shorter the ring length and the higher the number of nodes, the heavier the effect of the wasted slots on the aggregate throughput is.

Note that Figure 6.7.27 lacks the curve corresponding to a 10km-ring length with 16 nodes. That is because a 10km-ring provides only 15 slots, and with less slots than nodes all the nodes get zero transmission quota at every fairness cycle.

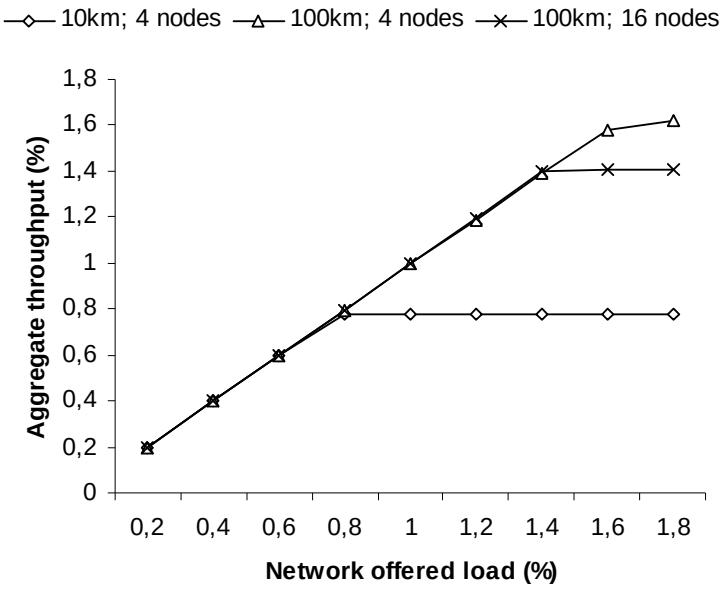


Figure 6.7.27 - Aggregate throughput in PAT under LCR and the symmetric traffic condition

Throughput fairness under the symmetric traffic distribution

Figure 6.7.28 to Figure 6.7.31 show the degree of throughput fairness provided by the four protocols combined with LCR under the symmetric traffic distribution.

It can be concluded from the curves that, with the exception of SPT+R, all the protocols achieve high degrees of throughput fairness. Again the problem is the difficulty in forwarding the fairness control packet. Since the fairness cycle has fixed length, the longer it takes for a node to forward the fairness packet, the longer the fairness cycle perceived by the other nodes gets. Consequently, the higher are the chances that the other nodes will exhaust their transmission quotas before they receive the control packet.

Since the same phenomenon might occur at every node, the network suffers from higher unfairness overall as the load increases, in particular as the number of nodes increases, as this causes higher contentions.

To illustrate how fairness affects performance Figure 6.7.32 shows the per node throughput achieved in SPT+R in a 10km ring with 16 nodes.

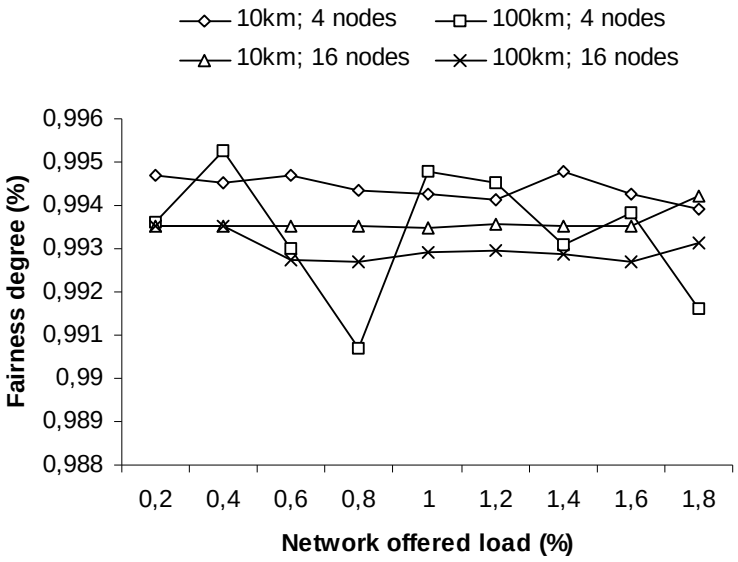


Figure 6.7.28 - Degree of throughput fairness in SPT+P under LCR and the symmetric traffic condition

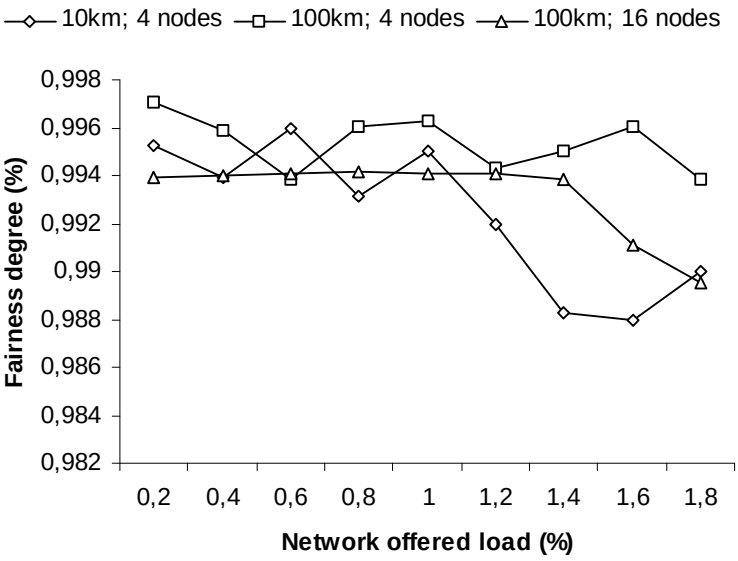


Figure 6.7.29 - Degree of throughput fairness in PAT under LCR and the symmetric traffic condition

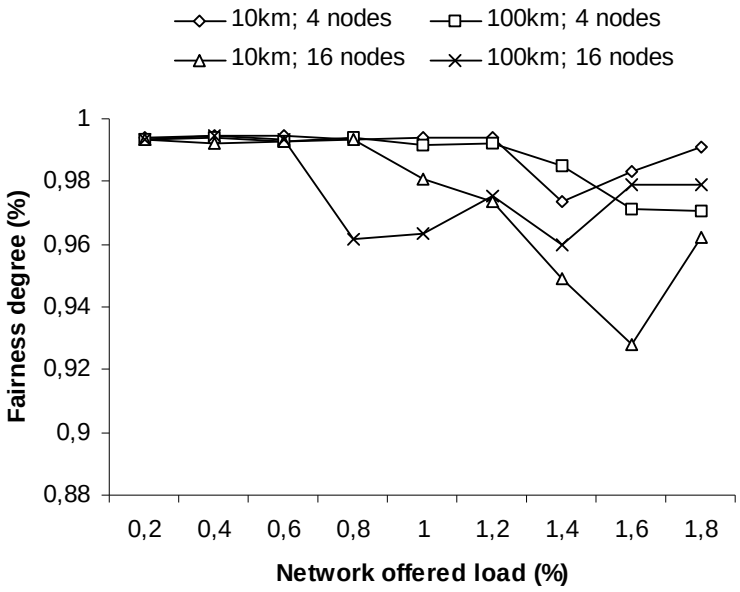


Figure 6.7.30 - Degree of throughput fairness in SPT+R under LCR and the symmetric traffic condition

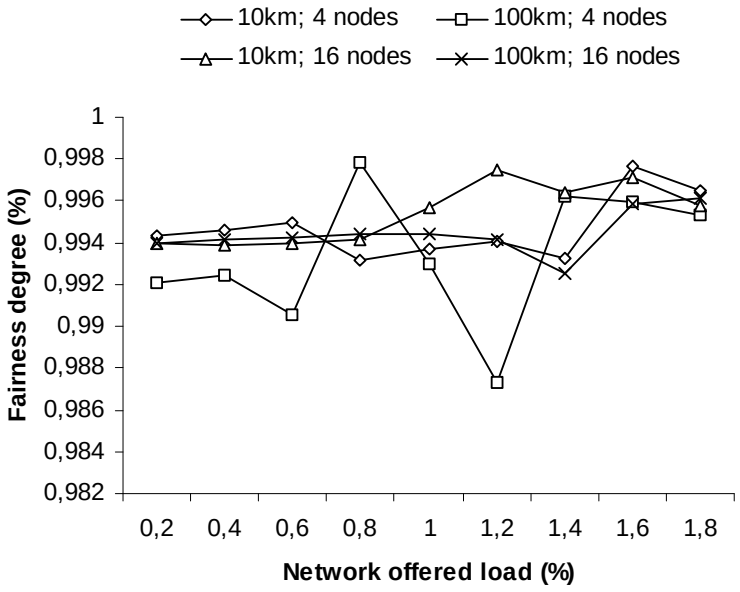


Figure 6.7.31 - Degree of throughput fairness in SPT+SC under LCR and the symmetric traffic condition

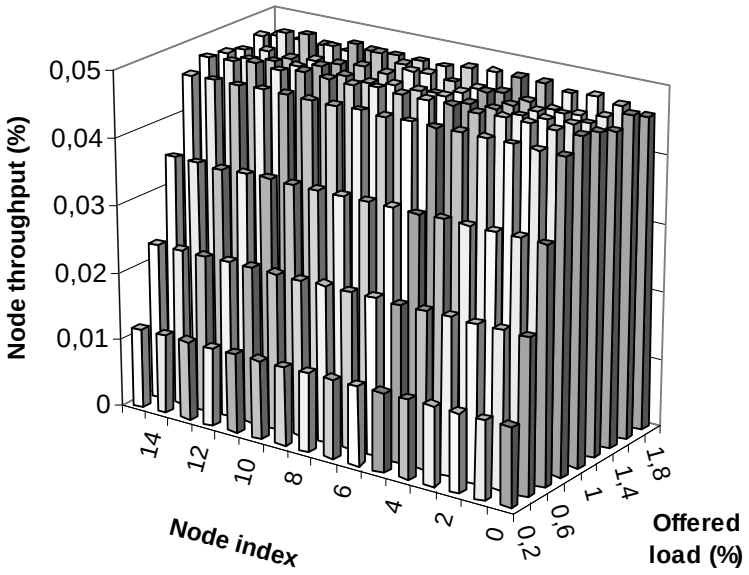


Figure 6.7.32 - Per node throughput in SPT+R under LCR and the symmetric traffic condition in a 16-node, 10km-ring

As Figure 6.7.32 shows, above the saturation point, which occurs approximately at the nominal capacity of the ring, the network becomes unfair with several nodes suffering from it.

Average network packet waiting time under the symmetric traffic distribution

Figure 6.7.33 shows the average network packet waiting times under each access control protocol as a function of the network offered load for a 100km-ring with 4 nodes. Figure 6.7.34 shows the average network packet waiting times as a function of the network offered load for a 100km-ring with 16 nodes.

The curves show that the effects of LCR fairness enforcement on the protocols to a certain extent superpose the characteristics of these protocols, hence providing all the protocols with similar packet waiting times.

Such a phenomenon is more evident when the number of nodes is small. As the number of nodes increases, the sensitiveness of SPT+SC and, in particular, of SPT+R, to an increasing number of nodes becomes evident.

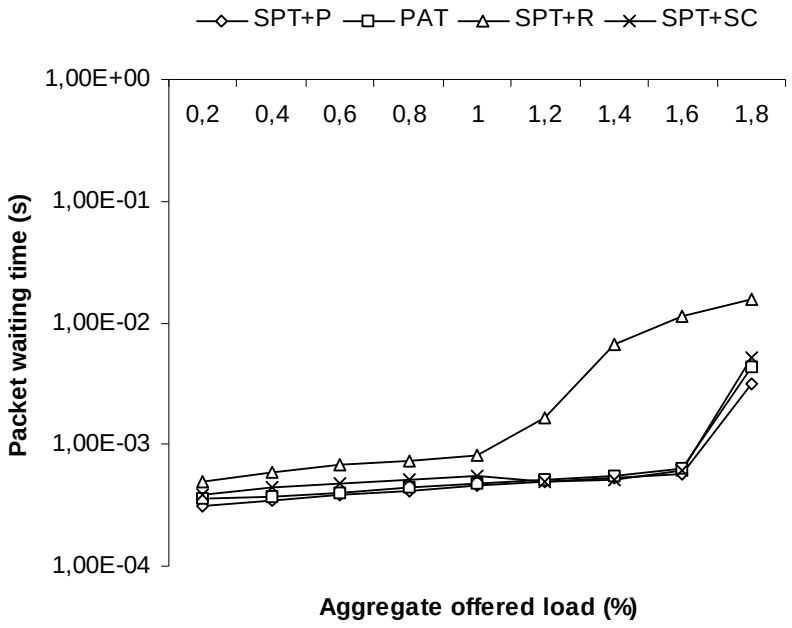


Figure 6.7.33 - Average network packet waiting time (in log. scale) in a 100km-ring with 4 nodes under the symmetric traffic distribution.

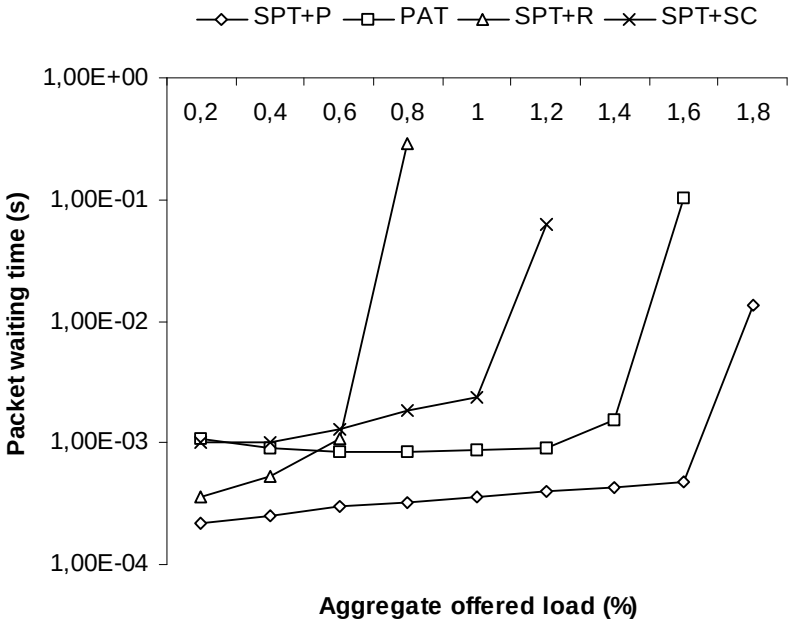


Figure 6.7.34 - Average network packet-waiting time (in log. scale) in a 100km-ring with 16 nodes under the symmetric traffic distribution.

Average waiting time in SPT+SC with multiple channels under the symmetric traffic distribution

Since SPT+SC is designed specifically for MOPS rings it is worthwhile to assess the effects of multiple channels on the packet waiting times. The results shown next assume a MOPS ring in a TTx^{1}-FRx^{C} configuration and a channel selection strategy that gives priority to the channel whose corresponding token is held. If no token is held then random selection applies.

Figure 6.7.35 shows the average network packet waiting times as a function of the aggregate offered load in a 10km-ring with 16 nodes and multiple channels.

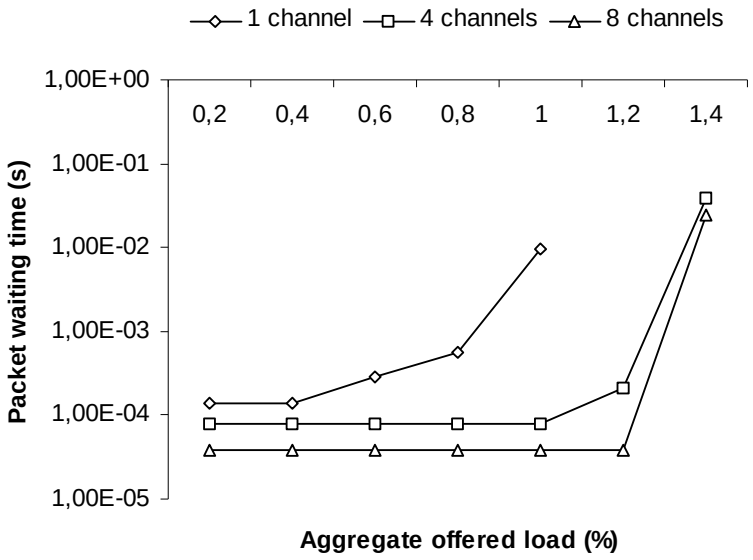


Figure 6.7.35 - Average network packet-waiting time (in log. scale) in a 10km-ring with 16 nodes and multiple channels and under SPT+SC and the symmetric traffic distribution.

The packet waiting times improve considerably with an increasing number of channels. As a matter of fact, so does the aggregate throughput. Although not shown, it can be seen from the curves that the aggregate throughput improves from 1 with a single channel to approximately 1.3 with eight channels, a gain of 30%.

Aggregate throughput under the asymmetric traffic distribution

Figure 6.7.36 to Figure 6.7.39 show the aggregate throughputs achieved by the four protocols combined with LCR under the asymmetric traffic distribution.

As it can be seen, only SPT+P achieves high aggregate throughputs independent of the ring length and the number of nodes. Again, that is because only SPT+P can exploit the fragmentation of the network capacity.

The aggregate throughputs achieved by the other protocols depend on the ring length and the number of nodes, and the reasons are those already explained.

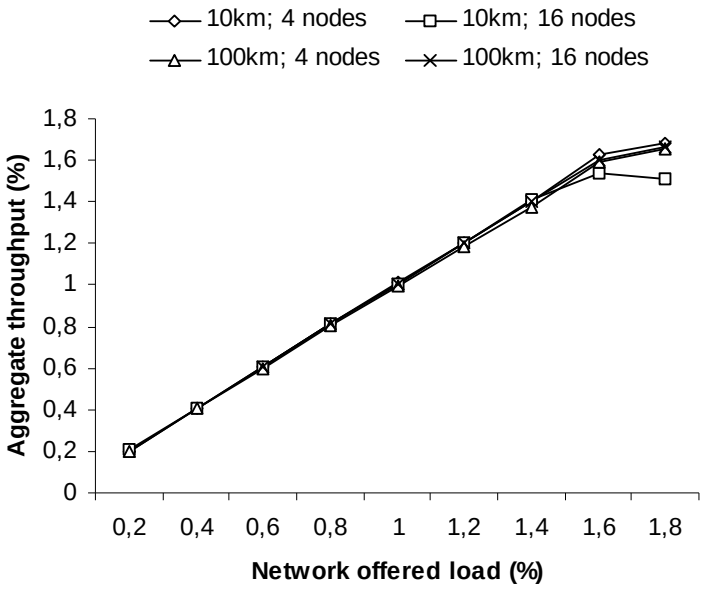


Figure 6.7.36 - Aggregate throughput in SPT+P under LCR and the asymmetric traffic condition.

The aggregate throughput of both PAT and SPT+R increases up to saturation, but above that it decreases sharply to a point at which it becomes steady, even though the load continues to increase.

The phenomenon occurs because of the mapping between the calculated fair rates and the allocated transmission quotas. As explained previously, a fair rate can be fractional, and the multiplication of a fraction by the fairness cycle capacity results in a transmission quota that is not an integer multiple of the slot size. Since it is not possible to allocate a fraction of a slot, waste of capacity occurs.

As the load increases nodes request more capacity from the network. Nevertheless as the load continues to increase the client nodes reach a point at which they have to request maximum capacity to cope with the backlogged traffic. Consequently, LCR yields fractional fair rates and the result is the reduction in the throughput achieved by the nodes because of waste. Such behaviour explains the aggregate throughput curves of PAT and SPT+R.

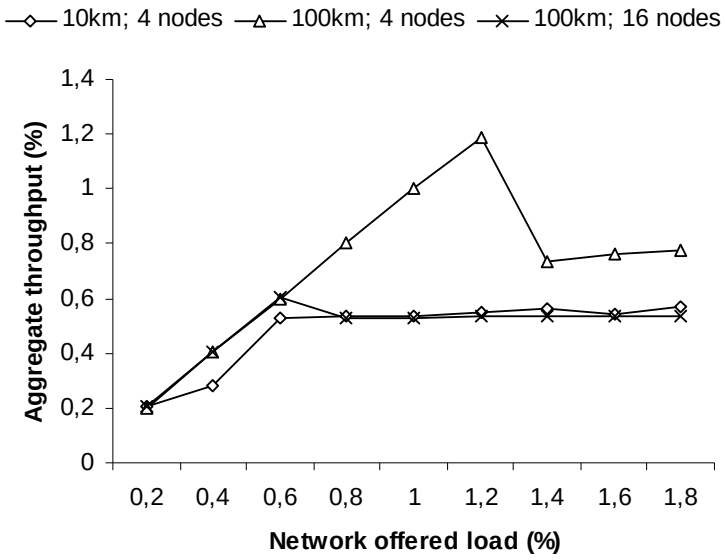


Figure 6.7.37 - Aggregate throughput in PAT under LCR and the asymmetric traffic condition.

It should be noted that the client nodes suffer the same throughput reduction, and the throughput of the server node suffers a reduction that is proportional to the sum of the reduction suffered by the other nodes. In other words, the throughput of all the nodes suffers, but the degree of fairness remains almost unaltered, as Figure 6.7.40 shows.

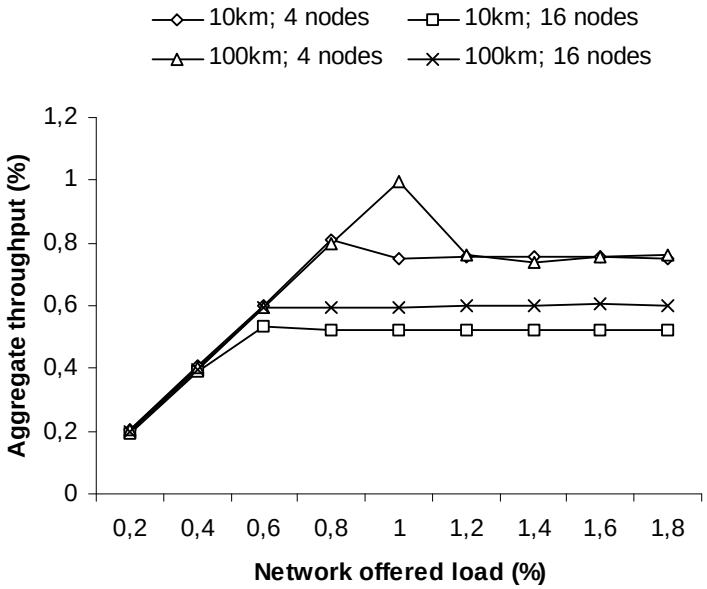


Figure 6.7.38 - Aggregate throughput in SPT+R under LCR and the asymmetric traffic condition.

Note that such a phenomenon occurs in PAT with the longer ring and in SPT+R with the smaller number of nodes. That is, the phenomenon occurs at the condition that suits the performance of each protocol the most. With the short ring the dominant factor behind the aggregate throughput achieved by PAT is the small number of slots. With the greater number of nodes the dominant factor behind the performance achieved by SPT+R is contention.

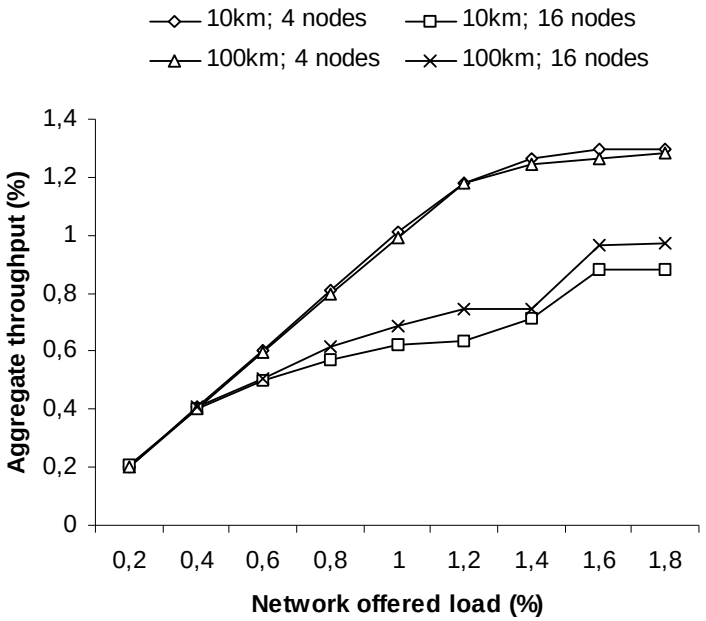


Figure 6.7.39 - Aggregate throughput in SPT+SC under LCR and the asymmetric traffic condition.

Figure 6.7.40 shows the per node throughput as a function of the network offered load in a 100km-ring with four nodes running PAT and LCR and under the asymmetric traffic condition.

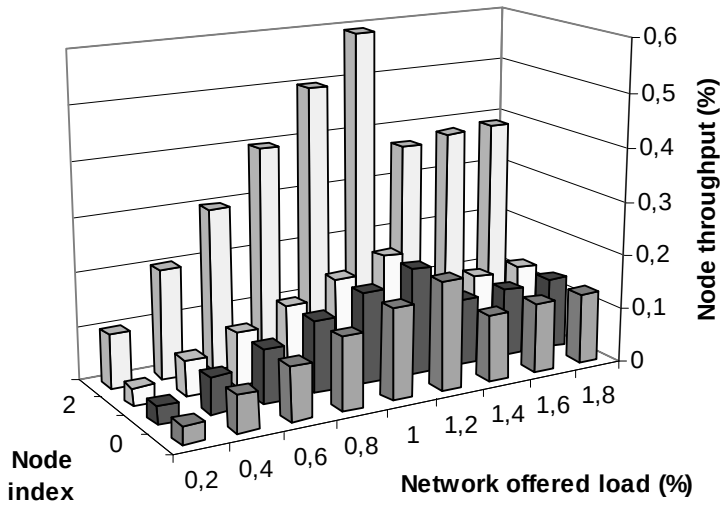


Figure 6.7.40 – Per node throughput in a 100km-ring with four nodes using PAT and LCR and under the asymmetric traffic condition.

Throughput fairness under the asymmetric traffic distribution

Figure 6.7.41 to Figure 6.7.44 show the degree of throughput fairness provided by the four protocols combined with LCR under the symmetric traffic distribution.

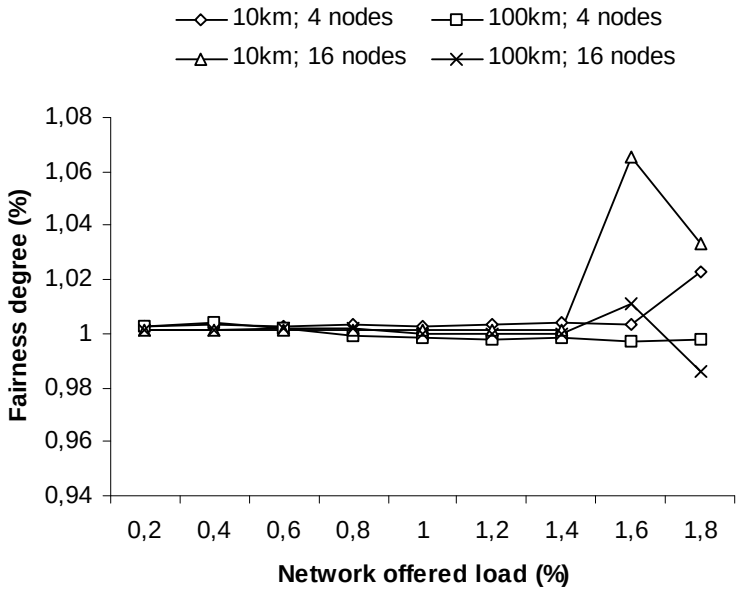


Figure 6.7.41 - Degree of throughput fairness in SPT+P under LCR and the asymmetric traffic condition

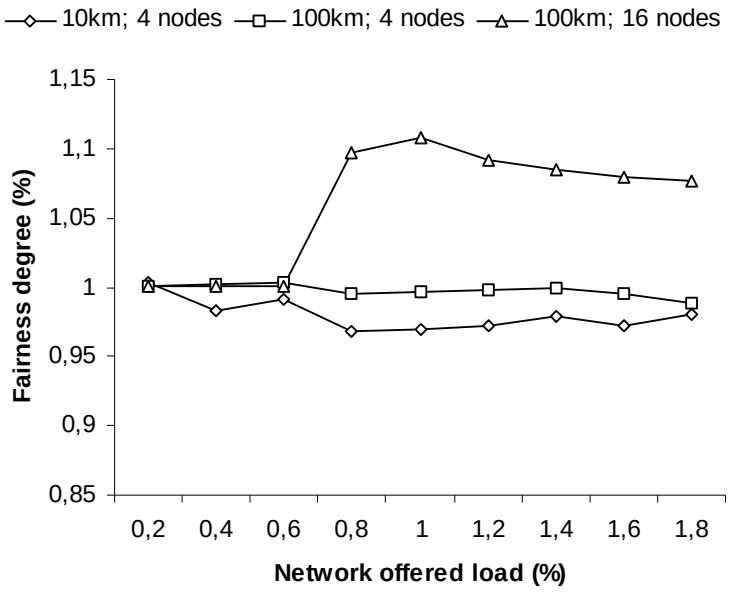


Figure 6.7.42 - Degree of throughput fairness in PAT under LCR and the asymmetric traffic condition.

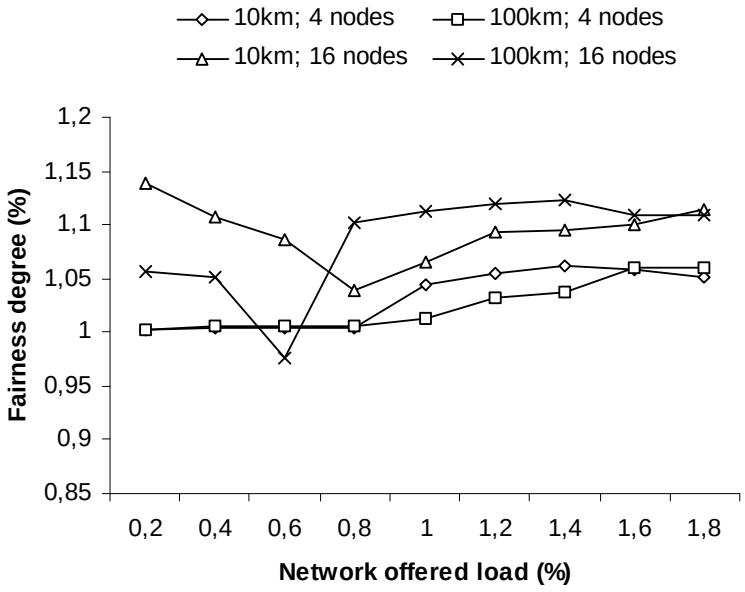


Figure 6.7.43 - Degree of throughput fairness in SPT+R under LCR and the asymmetric traffic condition.

As it can be seen, with the exception of SPT+SC, below the saturation point all the protocols achieve high degrees of throughput fairness, and even above that

point variations are small. This shows the efficiency of LCR in enforcing fairness at all network offered loads, including at saturation.

Contrary to SAT, LCR benefits the server node slightly in detriment to the client nodes above saturation.

The low degree of fairness seen in SPT+SC when there are 16 nodes stems from both the difficulty to forward the fairness control packet and the access control separation between SPT+SC and LCR.

A node can transmit whenever it holds a token, but LCR does not account for transmissions made by a node that holds a token. Thus, SPT+SC alone ensures every node the same access opportunity regardless of traffic distribution patterns.

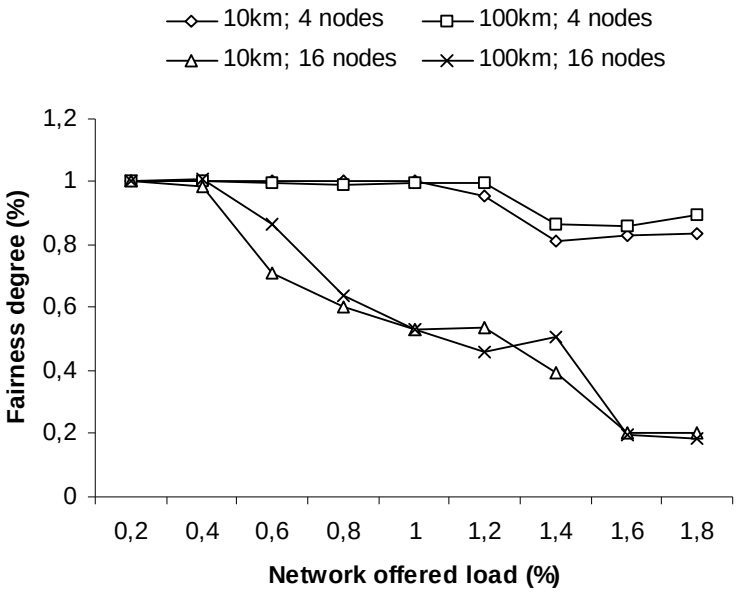


Figure 6.7.44 - Degree of throughput fairness in SPT+SC under LCR and the asymmetric traffic condition.

As explained previously, as the number of nodes increases it becomes more difficult to forward the control packet –also because SPT+SC permits every node to transmit regardless of what LCR determines, resulting in an increasing number of busy slot trains. Consequently, arbitrary nodes may receive the control packet long after they exhausted their transmission quotas. The result is unfairness.

Figure 6.7.45 shows the per node throughput achieved by SPT+SC under LCR in a 10km-ring with 16 nodes. The figure shows that the network becomes considerably unfair, in particular as the offered load increases.

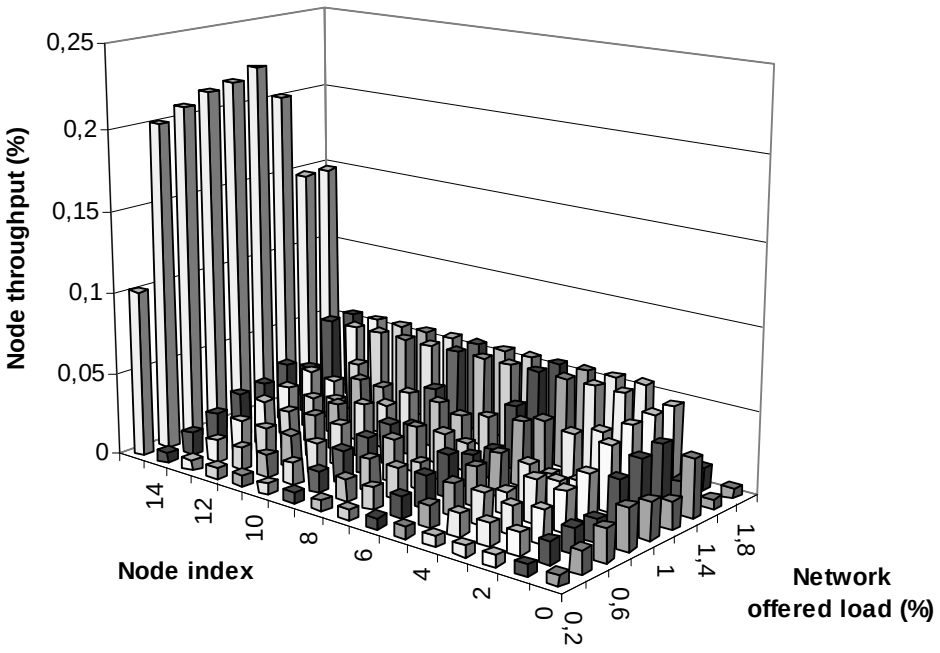


Figure 6.7.45 - Per node throughput in SPT+SC under LCR and the asymmetric traffic condition in a 16-node, 10km-ring.

Average network waiting time under the asymmetric traffic distribution

Since SPT+SC is designed specifically for MOPS rings it is worthwhile to assess the effects of multiple channels on the packet waiting times.

Figure 6.7.46 shows the average network packets waiting times experienced by the client nodes under each access control protocol as a function of the network offered load for a 100km-ring with 4 nodes. Figure 6.7.47 shows the average network packet-waiting times experienced by the server node as a function of the network offered load for a 100km-ring with 4 nodes.

Note that LCR provides the client nodes and the server node with the same packet waiting times, which shows that the throughputs are fair according to the definition of throughput fairness for this traffic condition.

Again because of the superposition of the LCR rules over the rules of the access control protocols, the packet waiting times experienced by the protocols are determined by LCR and are nearly flat below saturation.

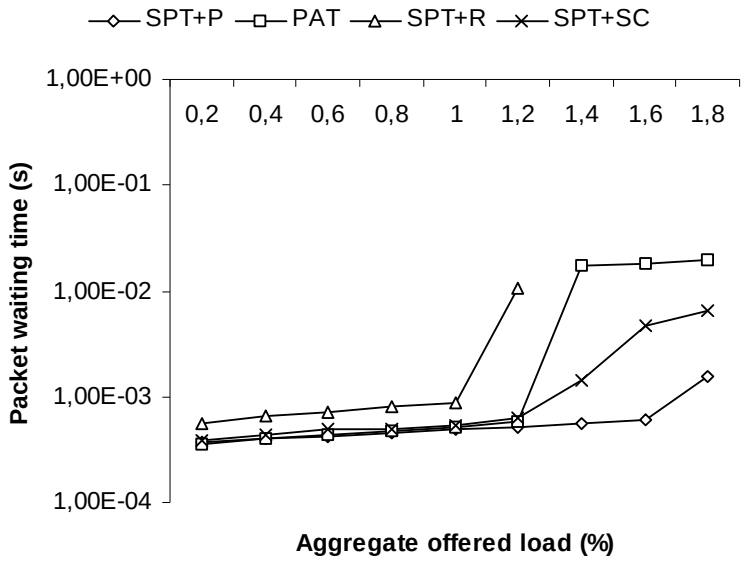


Figure 6.7.46 - Average packet-waiting time (in log. scale) experienced by the client nodes in a 100km-ring with 4 nodes under the asymmetric traffic distribution.

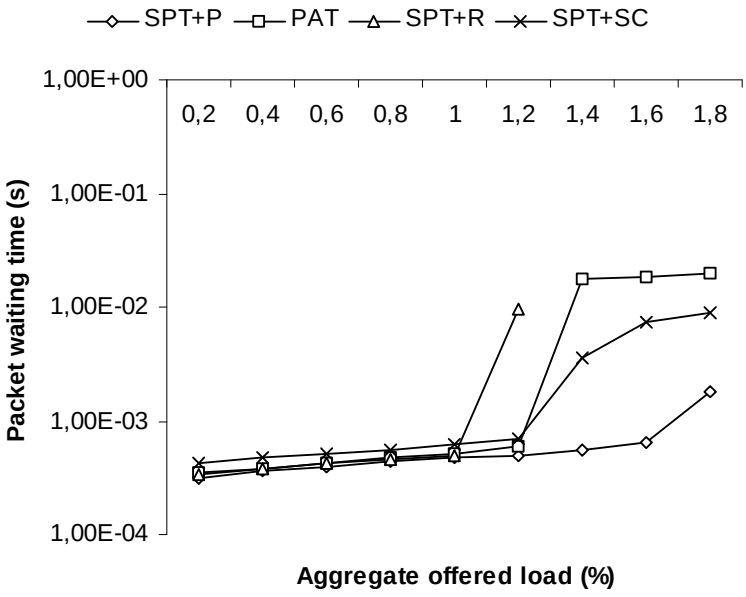


Figure 6.7.47 - Average packet-waiting time (in log. scale) experienced by the server node in a 100km-ring with 4 nodes under the asymmetric traffic distribution.

Figure 6.7.48 shows the average network packet-waiting times experienced by client nodes under each access control protocol as a function of the network offered load for a 100km-ring with 16 nodes. Figure 6.7.49 shows the average

network packet-waiting times experienced by the server node as a function of the network offered load for a 100km-ring with 16 nodes.

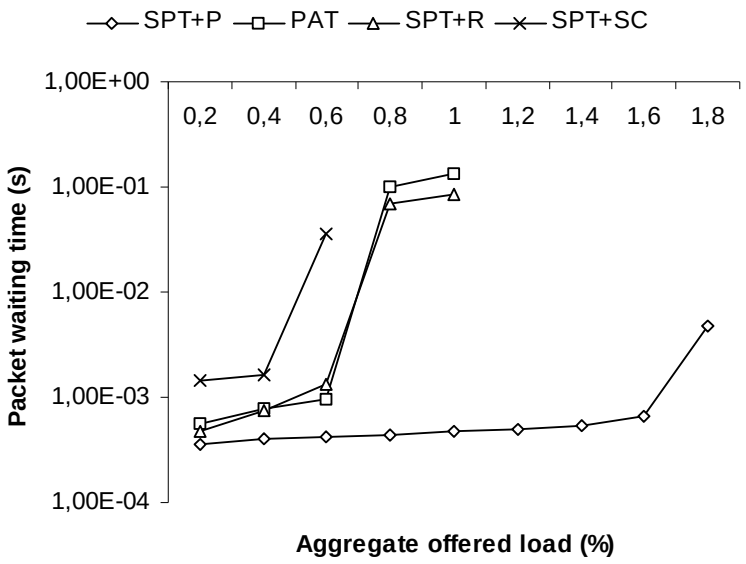


Figure 6.7.48 - Average packet-waiting time (in log. scale) experienced by the client nodes in a 100km-ring with 16 nodes under the asymmetric traffic distribution.

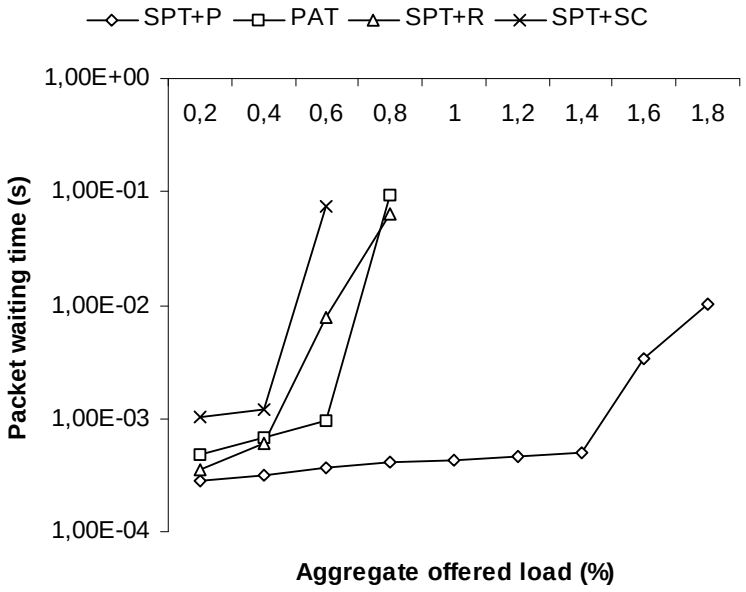


Figure 6.7.49 - Average packet-waiting time (in log. scale) experienced by the server node in a 100km-ring with 16 nodes under the asymmetric traffic distribution.

The packet waiting times in SPT+P suffer little influence from an increasing number of nodes. On the other hand, the packet waiting times experienced by the other protocols are too high. This shows the difficulty of SPT+R and SPT+SC to forward the control packet, and the limited number of slots in PAT for the transmission of data packets as well as the control packet.

Average waiting time in SPT+SC with multiple channels under the asymmetric traffic distribution

The results shown next assume a MOPS ring in a TTx^{1}-FRx^{C} configuration, and a channel selection strategy that gives priority to the channel whose corresponding token is held. If no token is held then random selection applies.

Figure 6.7.50 shows the average network packet-waiting times experienced by the client nodes as a function of the network offered load for a 10km-ring with 16 nodes. Figure 6.7.51 shows the average network packet-waiting times experienced by the server node as a function of the network offered load for a 10km-ring with 16 nodes.

The curves show that the packet waiting times experienced by the client nodes improve as the number of channels increases. That is because nodes get unconstrained transmission opportunities more frequently.

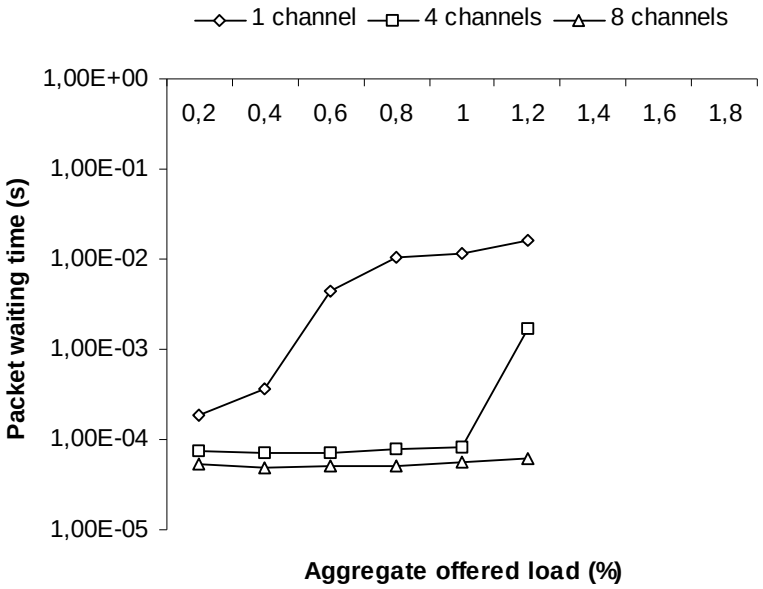


Figure 6.7.50 - Average packet-waiting time (in log. scale) experienced by the client nodes in a 10km-ring with 16 nodes and multiple channels under the asymmetric traffic distribution.

Nevertheless, the curves also show that, although an increase in the number of channels may improve the packet waiting times experienced by the server node, as the number of channels approximates the number of nodes LCR becomes less effective. That is because a node can transmit at a single channel at a time, and

whenever there are two empty slots available, or more, a node always attempts to select the slot on the channel whose token that node holds. Consequently, reuse decreases as the number of channels increases, and LCR looses its utility. That explains why the server node experiences high packet waiting times.

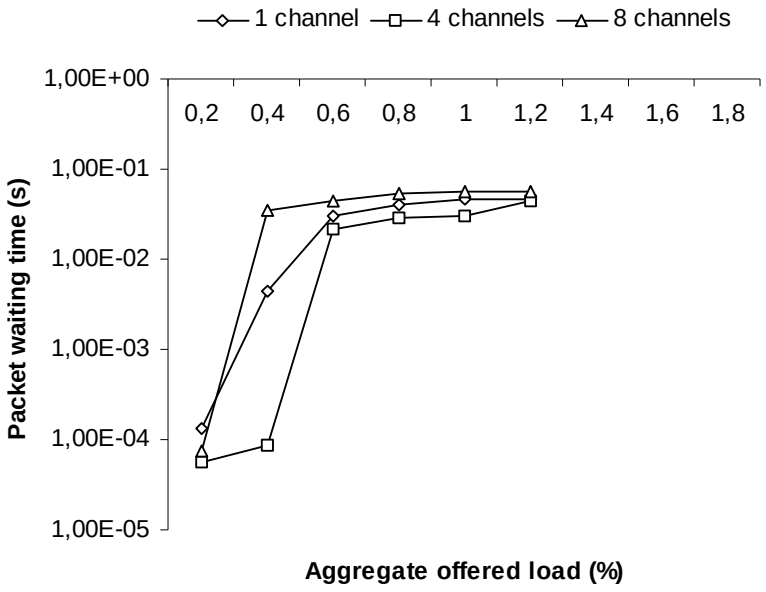


Figure 6.7.51 - Average packet-waiting time (in log. scale) experienced by the server node in a 10km-ring with 16 nodes and multiple channels under the asymmetric traffic distribution.

6.8 Discussion

The simulation results discussed in this chapter show that LCR is the only fairness protocol that can cope with traffic asymmetries typical of the Internet; both SAT and GCR fail to ensure fairness in asymmetric traffic conditions.

Nevertheless, LCR scales bad because the size of its control packet is given by the square of the number of nodes. As shown by the simulation results, as the control packet increases it becomes more difficult to forward the control packet and, consequently, fairness, performance, or both can deteriorate.

With the exception of SPT+P, the aggregate throughput of all the access control protocols under LCR degrades as the number of nodes increases. In addition, because it becomes difficult to forward the control packet the degree of throughput fairness can deteriorate too.

It should be emphasized that with 16 nodes the control packet is 1024B-long (that is, $16 \times 16 \times 4B = 1024B$), which is smaller than the typical 1500B-MTU of network links. This means that these protocols may have problems to transmit large packets independent of the number of nodes and the used fairness protocol. What is more, with bit rates increasing constantly there is a trend to increase

MTUs too, and with the implementation of path MTU discovery [Mogul1990] TCP hosts are likely to generate messages with the MTU size rather than the default 552B-message segment size as they currently do.

The only protocol that can exploit the fragmentation of the network capacity, SPT+P achieves high aggregate throughputs and high degrees of throughput fairness under LCR independent of the traffic distribution, the ring length, and the number of nodes.

Although SPT+P relies on SAR, because of its conception it demands less re-assembly buffer memory than the traditional SAR. This is important since in MOPS rings the bandwidth x latency product is high, and under this condition traditional SAR re-assembly buffers may have to hold information for long time, and having many buffers allocated simultaneously leads to high memory demands. Furthermore, it may also generate high bursts of re-assembled packets that the upper layers cannot cope with.

As the performance results of SPT+SC under LCR and the asymmetric traffic condition show, the integration between the access control protocols and LCR plays a strong part in the resulting performance and fairness. Therefore, different integration strategies may result in completely different performance figures.

An alternative integration strategy between SPT+SC and LCR is to let LCR control access regardless of token possession. Tests at initial design stages show that doing so leads to lower performances because nodes use their quota to transmit small packets when they do not possess the token and end up forbidden to transmit large packets when they do possess the token because they have no quota left.

It should be noted though that as the number of channels approximate the number of nodes such a phenomenon is reduced since the frequency at which nodes hold a token increases. Of course, that depends on the transceiver tunability of the destination nodes.

Chapter 7
Conclusions

This chapter emphasises the focus of this work and how it relates to existing works on the subject, highlights the main contributions of this dissertation and evaluates the obtained results, and points to further investigations.

7.1 General considerations

With the convergence of applications, services, media, and specialised networks to the Internet, the support of Internet services, the transport of Internet traffic, or both is pre-requisite to the acceptance of any new network technology or network design. Nevertheless, almost all the existing MAC protocols for MOPS rings focus solely on achieving OPS over the ring topology, neglecting the characteristics of the Internet traffic -the Internet traffic consists of variable-size packets that arrive in bursts, is often asymmetric, and it can change its distribution pattern dynamically.

This dissertation focuses on MAC protocols for MOPS rings that can cope with the characteristics of the Internet traffic. It addresses the problems involving the transport of Internet traffic over MOPS rings, the possible solutions to achieve that, and the effectiveness of such solutions.

As the related works, the work contained in this dissertation considers the conceptual node architecture upon which almost all MOPS ring architectures rely. Unlike those works though, this work aims at no specific network architecture. Rather, it intentionally attempts to be as open and general as possible.

7.2 Main contributions and evaluation of the results

The contributions of this dissertation to the state-of-the-art of MAC protocols for MOPS rings are listed below:

- Definition of an open system model that makes it possible for MAC protocols in general to be used with various networks;
- Design of access control protocols that attempt to transport variable size packets as single units;
- Design of access fairness protocols that can not only ensure fairness under various traffic patterns, but also have the potential to adapt to dynamic changes in traffic patterns;
- Evaluation of both the effectiveness and the performance of the protocols in the transport of traffic with the characteristics of the Internet's.

Simulation data show that among SAT, GCR, and LCR, only the latter is able to enforce throughput fairness under both symmetric and asymmetric traffic distribution patterns. Furthermore, under asymmetric traffic distribution patterns LCR achieves much higher throughputs.

Since Internet traffic often has asymmetric distribution patterns, effectiveness and high performance under such patterns become more important than complexity. Therefore, although LCR is more complex then SAT and GCR, it is more suitable for the Internet.

Simulation results of each access control protocol integrated with LCR show that only SPT+P can achieve high throughputs and low packet waiting times under various network conditions, and can guarantee high degrees of throughput fairness under those same conditions. The results highlight the low sensitiveness of SPT+P to traffic workloads, traffic distribution patterns, packet sizes, ring lengths, and number of nodes, and they show how such parameters can affect the performance and the degree of throughput fairness achieved by the integrated protocols. Therefore, among the four access control protocols integrated with LCR, SPT+P is the most suitable for the Internet, even though the use of re-assembly buffers makes this protocol the most expensive and not necessarily the simplest of all.

According to queuing size measurements (not shown in this dissertation) and the packet waiting times discussed previously, packet queues utilisation rates in SPT+P are very low and remain nearly steady up to sustained loads close to saturation. Also, because it uses cyclic reservations that can work at ring length time scales or longer, LCR has the potential to adapt to traffic burstiness and volatile traffic distribution patterns quickly. Therefore, although bursty traffic and volatile traffic distribution patterns were not considered in the simulations, both SPT+P and LCR could cope with such Internet traffic characteristics.

Although QoS is out of the scope of this work, the low queue utilisation rates and low packet waiting times experienced under SPT+P integrated with LCR indicate that some levels of QoS could be supported under these protocols. As a matter of fact, coefficients of variance (COVs) of packet waiting times collected in the simulations under symmetric traffic distribution patterns (not shown in this dissertation) ranging from 1.0 to 0.5, approximately, support that statement. Note that since packets experience no queuing at intermediate nodes the COVs express end-to-end packet delay variation correctly.

Despite its superiority over the other protocols, SPT+P may not necessarily be the best choice under any scenario. As pointed out in [Odlyz2000], data networks utilisation rates are low in the average and will continue that way. Therefore, even though SPT+R is significantly sensitive to traffic workloads, packet sizes, and number of nodes, it may still be more suitable to Internet networks with average low utilisation rates than SPT+P since it does the job and does not require re-assembly buffers like SPT+P. Furthermore, the use of a single control packet in LCR prejudices the performance achieved by SPT+R. Different integration strategies will likely lead to different performance numbers.

It should be noted, however, that the MTU of network links should increase with the transmission bit rates, and so should the size of TCP packet segments as the use of path MTU discovery increases. Since TCP packets constitute the majority of the Internet traffic, packet sizes should increase too. Therefore, SPT+R may become less efficient as the Internet evolves.

On the other hand, the dominant traffic in the future might no longer be TCP, and the packet sizes may change considerably.

The difficulty to predict how the Internet services and traffic will evolve makes it hard to select a protocol among others when they all are too sensitive to traffic characteristics. Therefore, the more insensitive to such parameters the protocol is, the better.

The complexity required to the transport of variable size packets under contention in the vertical domain and the collected performance results raise the question whether a protocol that uses indiscriminate SAR operations would not do better than SPT+P, even though such a protocol should require more re-assembly buffers than SPT+P.

7.3 Topics for future research

This dissertation leaves several topics for further investigation. The first one concerns the integration of LCR with the proposed access control protocols. As shown in Chapter 6, the size of the control packet used by LCR leads to poor scalability.

Two solutions to reduce the negative effects of the control packet that can be envisaged are:

- Each node holds the control packet for a certain time before releasing it, whereas the actual holding time depends on how timely or late the packet is. Doing so increases the fairness cycle length and decreases the effects of forwarding delays. For instance, if each node is supposed to hold the control packet for 50 slots before attempting its transmission, and a certain node receives the control packet with a delay of 25 slots, then that node holds the control packet for only 25 slots;
- Periodically each node broadcast its requests and calculates its fair rates, whereas the broadcast and fair rate calculation periods need not to be equal. Doing so increases the number of control packets to the number of nodes, but is also reduces the quadratic size factor of control packets to a linear factor that equals the number of nodes, thus lowering forwarding delays.

Another research topic for further investigation is the effects of self-similar traffic on the performance of the access control protocols. Self-similar traffic is bursty, and it may change dynamic and quickly not only the network workload, but also the fragmentation patterns of the network capacity and, consequently, the contention patterns. How such characteristics can affect the performance of the access control protocols should, therefore, be evaluated.

The ability of the fairness protocols to cope with traffic burstiness and volatile traffic distribution patterns deserves investigation too. As the simulation results show, the traffic distribution patterns have heavy influence on the performance of the protocols. Therefore, the transition from one pattern to another may affect the performance of the protocols too.

The comparison between the proposed protocols and a protocol that uses SAR operations arbitrarily (e.g., ATM) is of interest too. Comparing the performance numbers of each protocol, and the characteristics of each, will help determine which protocol suits MOPS rings the most.

The implications of these protocols on other system components also deserve attention at some later stages of investigation. An example is the impact on queuing management. For instance, a typical characteristic of SPT+P is that queues remain almost empty even at loads close to saturation, from which point the queues overflow abruptly. Nevertheless, random early discard (RED) [Floyd1993] relies on queue threshold indicators to detect queue build-up and starts discarding packets before the queue overflows. Thus, the ability of RED to react satisfactorily to such abrupt queue overflows should be studied.

Another issue is the effects of the protocols studied in this dissertation on TCP. TCP is sensitive to round trip delays and packet loss, and slight changes in the underlying transport mechanisms can result in unexpected TCP behaviours. Since the protocols have different performance characteristics, their effects on TCP should be investigated carefully too. In this sense, the larger delays experienced by big packets and its effects on TCP deserve special attention, in particular because as bit rates increase there is a trend to increase the MTU of the links, thus leading to bigger packets.

References

[Adams2002] A. Adams, J. Nicholas, and W. Siadak: Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). IETF Internet Draft <draft-ietf-pim-dm-new-v2-02.txt>, October 2002.

[Anast1997] G. Anastasi, L. Lenzini, and B. Meini: Performance Evaluation of a Worst Case Model of the MetaRing MAC Protocols with Global Fairness. In Performance Evaluation, Vol. 29, No. 2, pp. 127-151, March 1997.

[Anast2001] G. Anastasi, L. Lenzini, and Y. Ofek: Tradeoff between the Cycle Complexity and the Fairness of Ring Networks. In Elsevier Microprocessors and Microsystems, Vol. 25, No. 1, pp. 41-59, March 2001.

[ANSI1988] (ISO 9314-2) Fiber Distributed Data Interface (FDDI) - Media Access Control. 1988.

[ANSI2001] Synchronous Optical Network (Sonet) – Basic Description including Multiplex Structure, Rates, and Formats. ANSI T1.105-2001.

[Armit1996] G. Armitage: Support for Multicast over UNI 3.0/3.1 based ATM Networks. IETF RFC 2022, November 1996.

[Atkin1995] R. Atkinson: Security Architecture for the Internet Protocol. IETF RFC 1825, 1995.

[Balak2001] H. Balakrishnan, and V. N. Padmanabhan: How Network Asymmetry Affects TCP. In IEEE Communications Magazine, Vol. 39, No. 4, pp. 60-67, April 2001.

[Bengi2002] K. Bengi: Optical Packet Access Protocols for WDM for WDM Networks. Kluwer Academic Publishers, 2002.

[Berts1987] D. Bertsekas, and R. Gallager: Data Networks. Prentice Hall, 1987.

[Blake1998] S. Blake et al.: An architecture for Differentiated Services. IETF RFC 2475.

[Blume2001] M. S. Blumenthal, and D. D. Clark. Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World. In ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 70-109, August 2001.

[Bux1981] W. Bux. Local area subnetworks: a performance comparison. In IEEE Transactions on Communications, Vol. 29, No. 10, pp. 1465-1473, October 1981.

[Cai2000] J. Cai, A. Fumagalli, and I. Chlamtac: The Multitoken Interarrival Time (MTIT) Access Protocol for Supporting Variable Size Packets Over WDM Ring Network. In IEEE Journal on Selected Areas of Communications, Vol. 18, No. 10, pp. 2094-2103, October 2000.

[Cao2001] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun: On the nonstationarity of Internet Traffic. In Procs. of the ACM SIGMETRICS Joint International Conference on Measurements and Modelling of Computer Systems, pp. 102-112, Cambridge, MA, US, June 16-20, 2001.

[Cao2002a] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun: Internet Traffic: Statistical Multiplexing Gains. In DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling, February 12-15, 2002.

[Cao2002b] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun: Internet Traffic Tends Toward Poisson and Independent as the Load Increases. In Nonlinear Estimation and Classification, pp. 83-110, Springer, 2002.

[Caren2002] Carena et al.: RINGO: A demonstrator of WDM optical packet network on a ring topology. In Procs. of the 6th IFIP Working Conference on Optical Network Design and Modelling (ONDM), Torino, Italy, February 2002.

[Carpe1996] B. Carpenter: Architectural Principles of the Internet. IETF RFC 1958, June 1996.

[Chen1993] J. S.-C. Chen, I. Cidon, and Y. Ofek: A Local Fairness Algorithm for Gigabit LAN's/MAN's with Spatial Reuse. In IEEE Journal on Selected Areas in Communications, Vol. 11, No. 8, pp. 1183-1192, October 1993.

[Chlam1995] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and P. T. Poggiolini: A Contention/Collision Free WDM Ring Network for Multi Gigabit Packet Switched Communication. In Journal of High Speed Networks, Vol. 4, No. 2, 201-219, 1995.

[Chlam1999] I. Chlamtac, V. Elek, A. Fumagalli, and C. Szabo: Scalable WDM Access Network Architecture Based on Photonic Slot Routing. In IEEE/ACM Transactions on Networking, Vol. 7, No. 1, pp. 1-9, February 1999.

[Cidon1993] I. Cidon, and Y. Ofek: MetaRing – a full-dulex ring with fairness and spatial reuse. In IEEE Transactions on Communications, Vol. 41, No. 1, pp. 110-120, January 1993.

[Cidon1997] I. Cidon, L. Georgiadis, R. Guerin, and Y. Shavitt: Improved Fairness Algorithms for Rings with Spatial Reuse. In IEEE/ACM Transactions on Networking, Vol. 5, No. 2, pp. 190-204, April 1997.

[Claff1998] K. Claffy, G. Miller, and K. Thompson: The nature of the beast: recent traffic measurements from an Internet backbone. In Internet Society (ISOC) Internet Summit (Inet), Geneva, Switzerland, 21-24 July, 1998.

[Clark1988] D. D. Clark: The Design Philosophy of the Darpa Internet Protocols. In Procs. of ACM SIGCOMM'98 Symposium, pp. 106-114, Vancouver, Canada, August 31-September 4, 1998.

[Deeri1995] S. Deering, and R. Hinden: Internet Protocol, version 6 (IPv6) Specification. IETF RFC 1883, 1995.

[Dey2000] D. Dey, Ton Koonen, M. R. Salvador: Network Architecture of a Packet-Switched WDM LAN/MAN. In Procs. of the Fifth Annual Symposium of the IEEE/LEOS Benelux Chapter, pp. 256-259, Delft, The Netherlands, October 30, 2000.

[Dey2001a] D. Dey, A.M.J. Koonen, D. Geuzebroek and M. R. Salvador: FLAMINGO: A Packet-switched IP over WDM Metro Optical Network. In Procs. of 6th European Conference on Networks & Optical Communications (NOC'2001), pp. 4000-4007, Ipswich, UK, June 26-29, 2001.

[Dey2001b] D. Dey, A. van Bochove, A.M.J. Koonen, D. Geuzebroek and M. R. Salvador: FLAMINGO: A Packet-switched IP-over-WDM All-optical MAN. In Procs. of 27th European Conference on Optical Communications (ECOC'2001), pp. 480-4812, Amsterdam, NL, 30 September - 4 October, 2001.

[Dobos1992] W. Dobosiewicz, P. Gburzynski, and V. Maciejewski. A Classification of Fairness Measures for Local and Metropolitan Area Networks. In Elsevier Computer Communications, Vol.15, No. 5, pp. 295-304, June 1992.

[Dobos1993] W. Dobosiewicz, and P. Gburzynski: On token protocols for high-speed multiple-ring networks. In Procs. of IEEE International Conference on Network Protocols (ICNP'93), pp. 300-307, San Francisco, CA, US, October 19-22, 1993.

[Dobos1995] W. Dobosiewicz, and P. Gburzynski: On a MAC protocol based on distributed cycles. In Journal of High Speed Networks, Vol. 4, No. 3, pp. 275-286, 1995.

[Dutto1995] H. J. R. Dutton, and P. Lenhard: Asynchronous Transfer Mode (ATM): Technical Overview. Prentice Hall PTR, 1995.

[Estri1998] D. Estrin et al.: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. IETF RFC 2362, June 1998.

[Fairh2002] G. Fairhurst, and L. Wood: Advice to link designers on link Automatic Repeat reQuest (ARQ). IETF RFC 3366, August 2002.

[Falco1985] R.M. Falconer, J. L. Adams, and G. M. Walley: Orwell: a protocol for an integrated services local network. British Telecom Technologies Jornal, Vol. 3, No. 4, pp. 27-35, October 1985.

[Floyd1993] S. Floyd, and V. Jacobson: Random Early Detection gateways for Congestion Avoidance. In IEEE/ACM Transactions on Networking, Vol.1, No. 4, pp. 397-413, August 1993.

[Foudr1991] E. C. Foudriat, K. Maly, C. M. Overstreet, S. Khanna, and F. Paterra: A carrier sensed multiple access protocol for high data rate ring networks. In ACM Computer Communications Review, Vol 21, No. 2, pp. 59-70, 1991.

[Fowle1999] M. Fowler, and K. Scott: UML Distilled. Addison-Wesley, second edition, 1999.

[Frans1998] J. Fransson, M. Johansson, M. Roughan, L. Andrew, and M. A. Summerfield: Design of a Medium Access Control Protocol for a WDMA/TDMA Photonic Ring Network. In Procs. of IEEE Global Telecommunications Conference (GLOBECOM'98), pp. 307-312, Sydney, Australia, November 8-12, 1998.

[Hill2001] M. Hill et al: 1x2 Optical Packet Switch Using All-Optical Header Processing. In Electonic Letters, Vol. 37, No. 12, pp. 774-775, 2001.

[Hopp1980] A. Hopper: The Cambridge ring-a local network. Advanced Techniques for Microprocessor Systems, F. K. Hanna (ed.), Peter Pergrinus Ltd., pp. 67-71, Stevenage, U.K., 1980.

[IEEE1998] (ISO/IEC 8802-5:1998) IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 5: Token Ring Access Method and Physical Layer Specification.

[IEEE2001] (IEEE Std 802-2001) Local and Metropolitan Area Networks: Overview and Architecture. 2001.

[IEEE2002] (IEEE 802.3-2002) IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific

requirements--Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

[Imai1994] K. Imai, T. Ito, and N. Morita: ATMR: Asynchronous transfer mode ring protocol. In Computer Networks and ISDN Systems, Vol. 26, No. 6-8, pp. 785-798, March 1994.

[ISO1994] ISO/IEC 7498-1:1994: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.

[ITUT2000] Architecture of transport networks based on the synchronous digital hierarchy (SDH). ITU-T G.803 (03/00), 2000.

[Jacob1990] V. Jacobson: Compressing TCP/IP Headers for Low-Speed Serial Links. IETF RFC 1144, February 1990.

[Jain1986] R. Jain, and S. A. Routhier: Packet trains -- measurement and a new model for computer network traffic. In IEEE Journal on Selected Areas in Communications, Vol. 4, No. 6, pp. 986-995, September 1986.

[Jain1991] R. Jain: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, Inc., 1991.

[Kang1995] C.-S. Chen, B.-S. Park, J.-D. Shin, and J.-M. Jeong: A broadband ring network: multichannel optical slotted ring. In Elsevier Computer Networks and ISDN Systems, Vol. 27, No. 9, pp. 1387-1398, 1995.

[Karn2002] P. Karn: Advice for Internet Subnetwork Designers. IETF Internet Draft <draft-ietf-pilc-link-design-12.txt>, January 2002.

[Klein1973] L. Kleinrock, and S. Lam: Packet-Switching in a Slotted Satellite Channel. In Procs. of the AFIPS National Computer Conference (NCC), Vol. 42, pp. 703-710, New York, NY, US, June 1973.

[Law2000] A. M. Law, and W. D. Kelton: Simulation modeling and analysis. Mc-Graw Hill, 2000.

[Lelan1994] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson: On the Self-Similar Nature of Ethernet Traffic (Extended Version). In IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1-15, February 1994.

[Marsa1993] M. A. Marsan, C. Casetti, S. M. Grasso, and F. Néri: Slot Reuse in MAC Protocols for MAN's. In IEEE Journal on Selected Areas of Communications, Vol. 11, No. 8, pp. 1290–1301, October 1993.

[Marsa1996] M. A. Marsan, A. Bianco, E. Leonardi, M. Meo, and F. Neri: MAC Protocols and Fairness Control in WDM Multi-Rings with Tunable Transmitters and Fixed Receivers. In IEEE Journal of Lightwave Technology special issue on Multiwavelength Optical Technology and Networks, Vol.14, No.6, pp.1230-1244, June 1996.

[Marsa1997] M. A. Marsan, A. Bianco, E. Leonardi, F. Neri, and S. Toniolo: An Almost Optimal MAC Protocol for All-Optical WDM Multi-Rings with Tunable Transmitters and Fixed Receivers. In Procs. of IEEE International Conference on Computers and Communications (ICC), pp. 437-442, Montreal, Canada, June 1997.

[Matsu1998] M. Matsumoto, and T. Nishimura: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. In ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, pp.3-30, January 1998.

[Mayer1996] A. Mayer, Y. Ofek, and M. Yung: Approximating Max-Min Fair Rates via Distributed Local Scheduling with Partial Information. In Proc. of IEEE INFOCOM, pp. 928-936, San Francisco, CA, USA, March 24-28,1996.

[Mitra1986] I. Mitrani, J. L. Adams, R. M. Falconer: A modelling study of the Orwell ring protocol. In Procs. of the International seminar on Teletrafic Analysis and Computer Performance Evaluation. Elsevier Science Publishers, pp. 429-438, Amsterdam, NL, December 1986.

[Mogul1990] J. Mogul, and S. Deering: Path MTU Discovery. IETF RFC 1191, November 1990.

[Moors1997] T. Moors: Asynchronous Transfer Mode Implementation Issues and Solutions. Ph.D. dissertation, Australian Telecommunications Research Institute, Curtin University of Technology, 1997.

[Morri1984] R. J. I. Morris, Y. I. Wang: Some results for multi-queue systems with multiple cyclic servers. In Performance of Computer Communication Systems. Elsevier Science Publishers, pp. 4.2A-5-1 - 4.2A-5-7, 1984.

[Moy1994] J. Moy: Multicast Extensions to OSPF. IETF RFC 1584, March 1994.

[Mukhe1992] B. Mukherjee: WDM-based local lightwave networks part I: Single-hop systems. In IEEE Network magazine, Vol. 6, No. 3, pp. 12-27, May 1992.

[Murat2001] M. Murata, and K. Kitayama: A perspective on photonic multiprotocol label switching. IEEE Network Magazine, Vol. 15, No. 4, pp. 56-63, July/August 2001.

[Odlyz2000] A. Odlyzko: The Internet and other networks: utilization rates and their implications. In Elsevier Information Economics and Policy Journal, Vol. 12, No. 4, pp. 341-365, December 2000.

[Ofek1994] Y. Ofek: Overview of the MetaRing Architecture. In Computer Networks and ISDN Systems, Vol. 26, No. 6-8, pp. 817-830, March 1994.

[Pasch1991] H. L. Pasch, and I. G. Niemegeers: A high-speed slotted access ring access mechanism with dynamically adaptive slor sizes. In Procs. of EFOC/LAN'91, pp. 457-461, London, England, June 1991.

[Paxso1995] V. Paxson and S. Floyd: Wide-Area Traffic: The Failure of Poisson Modeling. In IEEE/ACM Transactions on Networking, Vol. 3, No. 3, pp. 226-244, 1995.

[Pawli2002] K. Pawlikowski, H.-D. Joshua Jeong, and J.-S. Ruth Lee: On Credibility of Simulation Studies of Telecommunication Networks. In IEEE Communications Magazine, Vol. 40, No. 1, pp. 132-139, January 2002.

[Peter2000] L. L. Peterson, and B. S. Davie: Computer Networks: A Systems Approach. Morgan Kaufmann Publishers, 2000.

[Poste1981] J. Postel: Transmission Control Protocol. IETF RFC 793, 1981.

[Qiao1999] C. Qiao, M. Jeong, A. Guha, X. Zhang and J. Wei: WDM Multicasting in IP over WDM Networks. In Procs. of the IEEE International Conference on Network Protocols (ICNP), pp. 89-96, Toronto, Canada, October 1999.

[Ramas1998] R. Ramaswami, and K. N. Sivarajan: Optical Networks: A Practical Perpesctive. Morgan Kaufmann Publishers, 1998.

[Reame1977] C. C Reames, and M. T. Liu: A Loop Network for Simultaneous Transmission of Variable-Length Messages. In Procs. of the fourth annual International Symposium on Computer Architecture (ISCA), pp. 193-200, January 1977.

[Rodel1999] D. Rodellar, and C. Bungarzeanu: Simulation of Multi-Channel MAC Protocols with Poisson and Fractal Traffic Sources. In Procs. of World Multiconference on Systemics, Cibernetics and Informatics (SCI), Vol. 4, August 1-3, 1999.

[Ryu1996] B. K. Ryu, and S. B. Lowen: Point Process Approaches to the Modeling and Analysis of Self-Similar Traffic – Part I: Model Construction. In Procs. of INFOCOM, pp. 1468-1475, San Francisco, CA, US, March 24-28, 1996.

[Saltz1984] J. Saltzer, D. Reed, D.D. Clark: End-to-end arguments in system design. In ACM Transaction on Computer Systems, Vol. 2, No. 4, pp. 277-288, November 1984.

[Salva2000] M. R. Salvador, S. H. de Groot and D. Dey: Supporting IP Dense Mode Multicast Routing Protocols in WDM All-Optical Networks. In Procs. of the First SPIE/IEEE/ACM International Conference on Optical Networking and Communications (OPTICOMM), Vol. 4233, pp. 167-178, Richardson, Texas, USA, October 22-26, 2000.

[Salva2001a] M. R. Salvador, S. Heemstra de Groot, and D. Dey: Supporting PIM-SM in All-Optical Lambda-Switched Networks. In (digital) Procs. of the 19th Brazilian Symposium on Computer Networks (SBRC), Florianopolis, SC, Brazil, 16pp., May 2001.

[Salva2001b] M. R. Salvador, S. H. de Groot and D. Dey: An All-Optical WDM Packet-Switched Network Architecture with Support for Group Communication. In Procs. of IEEE International Conference on Networking (ICN), Springer Lecture Notes in Computer Science (LNCS) 2093, pp. 326-335, Colmar, France, July 2001.

[Salva2001c] M. R. Salvador, S. H. de Groot and D. Dey: Supporting IP Dense Mode Multicast Routing Protocols in All-Optical Lambda-Switched Networks. In SPIE Optical Networks Magazine special issue on Routing Architectures and Technologies for Next-Generation WDM-Based Internet Networks, Vol. 2, No. 6, pp. 35-45, November/December 2001.

[Salva2002a] M. R. Salvador, S. Heemstra de Groot, and D. Dey: A preemption-enabled time slotting MAC protocol for all-optical ring LANs/MANs. In Procs. of IFIP 6th Working Conference on Optical Networks Design and Modelling (ONDM 2002), Torino, Italy, February 4-6, 2002.

[Salva2002b] M. R. Salvador, S. Heemstra de Groot, and D. Dey: MAC Protocol of a Next-Generation MAN Architecture Based on WDM and All-Optical Packet Switching. In Kluwer Academic Publishers Journal of Telecommunications Systems, Vol. 19, No. 3-4, pp. 377-401, March 2002.

[Salva2003a] M. R. Salvador, S. Heemstra de Groot, and D. Dey: A slot concatenation access protocol to transport variable size packets in all-optical WDM rings. Selected for fast track publication in SPIE Optical Networks Magazine, 2003.

[Salva2003b] M. R. Salvador, S. Heemstra de Groot, and D. Dey: A contention access protocol for WDM optical packet-switched rings. CTIT Technical Report, 2003.

[Salva2003c] M. R. Salvador, S. Heemstra de Groot, and D. Dey: A local proactive fairness protocol for packet switching rings. CTIT Technical Report, 2003.

[Shrik2000a] K. V. Shrikhande et al.: HORNET: A Packet-Over-WDM Multiple Access Metropolitan Area Ring Network. In IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, pp. 2004-2016, October 2000.

[Shrik2000b] K. V. Shrikhande et al.: CSMA/CA MAC Protocols for IP-HORNET: An IP over WDM Metropolitan Area Ring Network. In Procs. of IEEE Global Telecommunications Conference (GLOBECOM'2000), Vol. 2, pp. 1303-1307, San Francisco, CA, USA, Nov. 27 - Dec. 1, 2000.

[Shrik2001] K. V. Shrikhande et al.: Performance Demonstration of a Fast-Tunable Transmitter and Burst Mode Packet Receiver for HORNET. In Optical Fiber Communication (OFC) Conference and Exhibit Technical Digest, Vol. 4, paper ThG2-T1-3, Anaheim, California, US, March 17-22, 2001.

[Summe1997] M.A. Summerfield: MAWSON: A Metropolitan Area Wavelength Switched Optical Network. In Procs. of 3rd Asia Pacific Conference on Communications (APCC '97), Vol. 1, pp. 327-331, Sydney, Australia, December 7-10, 1997.

[Thomp1997] K. Thompson, G. J. Miller, and R. Wilder: Wide-Area Internet Traffic Patterns and Characteristics. In IEEE Network Magazine, Vol. 11, No. 6, pp. 10-23, November/December 1997.

[vanAs1994a] H. R. van As: Media access techniques: The evolution towards terabit/s LANs and MANs. In Computer Networks and ISDN Systems, Vol. 26, No. 6-8, pp. 603-656, March 1994.

[vanAs1994b] H. R. van As, W. W. Lemppenau, H. R. Schindler, and P. Zafiropulo: 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, March 1994.

[vanAs2001] H. R. van As, K. Bengi, and A. Lila: Cyclic-Reservation RPR MAC Protocol with Link-Fairness. Presentation at IEEE 802.17 RPR meeting, September 2001.

[Waitz1988] D. Waitzman, C. Partridge, and S. Deering: Distance Vector Multicast Routing Protocol. IETF RFC 1075, November 1988.

[Wang2000] G. Wang: Extensions to OSPF/IS-IS for Optical Routing. IETF Internet Draft <draft-wang-ospf-isis-lambda-te-routing-00.txt>, March 2000.

[Zafir1987] M. Zafirovic-Zukovic, and I. G. Niemegeers: Analytical models of the slotted ring protocols in HSLANs. In Procs. of IFIP WG 6.4 Workshop on High Speed Local Area Networks, pp. 115-134, Aachen, North Holland, 1987.

[Zafir1988] M. Zafirovic-Vukotic: Performance Modelling and Evaluation of High Speed Serial Interconnection Structures. Ph.D. Thesis, University of Twente, Enschede, the Netherlands, 1998.

[Zafir1999] M. Zafirovic-Zukovic, and I. G. Niemegeers: Waiting Time Estimates in Symmetric ATM-Oriented Rings with the Destination Release of Used Slots. In IEEE/ACM Transactions on Networking, Vol. 7, No. 2, pp. 251-261, April 1999.

[Zhang1999] X. Zhang, J. Wei and C. Qiao: On Fundamental Issues in IP over WDM Multicast. In Procs. of the 8th IEEE International Conference on Computer Communications and Networks (IC3N), pp. 84-90, Boston, USA, October 1999.

Appendix A
List of acronyms

This appendix contains the list of acronyms used throughout the dissertation. Note that neither state variables nor message names are considered as acronyms. Therefore, they are not included in this appendix.

ACF - Access control field
ACM - Association for computing machinery
ADDM - Add-drop decision making
AOTF - Acousto-optic tuneable filter
AP - Access point
ARP - Address resolution protocol
ATM - Asynchronous transfer mode
ATMR - ATM ring
AWG - Arrayed waveguide
B - Bytes
BER - Bit error rate
BOF - Begin of frame
BORN - Broadband optical ring network
CDO - Capability description object
CERN - European organization for nuclear research
CH - Channel
CLL - Capability list length
CLU - Control logic unit
COV - Coefficient of variance
CR - Cambridge ring
CRC - Cyclic redundancy check
CRMA - Cyclic reservation multiple access
CSMA/CA - Carrier sense multiple access with collision avoidance
CSMA/RN - Carrier sensed multiple access ring network
DA - Destination address
DBR - Distributed Bragg reflector
DC - Direct current
DCP - Distributed cycle protocol
DLCN - Distributed loop computer network
DMS - Data minislot
DVMRP - Distance vector multicast routing protocol
EDFA - Erbium-doped fibre amplifier
EOF - End of frame

ESCA - Empty slot contention/collision avoidance
FBG - Fibre Bragg grating
FCFS - First-come-first-serve
FCS - Frame check sequence
FDDI - Fibre distributed data interface
FDL - Fibre delay line
FPF - Fabry-Perot filter
FRx - Fixed Rx
FSM - Finite state machine
FTx - Fixed Tx
Gb/s - Gigabit per second
GCR - Global cyclic reservation
GCSR - Grating coupler sampled reflector
GMIB - Group membership information base
HD - Header detector
HOL - Head-of-line
HORNET - Hybrid opto-electronic ring network
HP - Header processor
HPU - HP unit
HT - Header type
IEEE - Institute of electrical and electronics engineers
IETF - Internet engineering task force
IP - Internet protocol
IPv6 - IP version 6
ISO - International standardization organisation
ITD - Intertoken distance
KB - Kilobyte
LAN - Local area network
LCR - Local cyclic reservation
LRD - Long-range dependence
MAC - Medium access control
MAN - Metropolitan area network
Max - Maximum
MAWSON - Metropolitan area wavelength-switched optical network
Min - Minimum
MOPS - Multiple-wavelength optical packet-switched
MOSPF - Multicast extensions to open shortest path first
MSS - Multiple subcarrier signalling
MTIT - Multitoken interarrival time
MTU - Maximum transmission unit
NIU - Network interface unit
OADM - Optical add-drop multiplexer
OPS - Optical packet switching
OSI - Open systems interconnection
PAT - Packet aggregate transmission

PCI - Protocol control information
PE - Protocol entity
PHY - Physical layer
PIM-DM - Protocol independent multicast - dense mode
PIM-SM - PIM - sparse mode
PLI - Payload length information
PType - Protocol type
PVL - Payload vector length
QoS - Quality of service
R/A - Request/allocation
RAP - Request/allocation protocol
RB - Receive buffer
RED - Random early discard
RFC - Request for comment
RINGO - Ring optical network
RM - Reference model
RTDM - Reception and transmission decision making
Rx - Receiver
SA - Source address
SAR - Segmentation and reassembly
SDH - Synchronous digital hierarchy
SOA - Semiconductor optical amplifier
Sonet - Synchronous optical network
SPIE - Society for optical engineering
SPT+P - Slotted packet transmission with preemption
SPT+R - Slotted packet transmission with retransmission
SPT+SC - Slotted packet transmission with slot concatenation
SRR - Synchronous round-robin
Sup-FRP - Superposition of fractional renewal processes
TB - Transmit buffer
Tb/s - Terabit per second
TCP - Transport control protocol
TDMA - Time division multiplexing access
TE - Time elapsed
THT - Token holding time
TIAT - Token interarrival time
TIB - Transceiver information base
TLI - Train length information
TRTr - Token rotation timer
TRx - Tuneable Rx
TS - Time stamp
TTIT - Target token interarrival time
TTL - Time-to-live
TTx - Tuneable Tx
Tx - Transmitter

UCN - Universal channel network
UML - Unified modelling language
VOQ - Virtual output queuing
VPN - Virtual private network
WAN - Wide area network
WADM - Wavelength add-drop multiplexer
WDM - Wavelength division multiplexing
WWW - World wide-web

Appendix B
Synchronisation

This appendix is concerned with synchronisation in MOPS rings. It defines a synchronisation algorithm for the initialisation of MOPS rings and discusses mechanisms to maintain the network synchronised.

MOPS rings rely on the network synchronisation to work properly, where network synchronisation means the synchronisation between every two adjacent time slots as well as the synchronisation between the slot headers and their corresponding payload slots. The protocols introduced in Chapter 4 assume that networks are always synchronised. Therefore, before becoming operational a MOPS ring must achieve synchronisation.

The first form of synchronisation (on the same channel) usually requires the election of a leader or master node that will become responsible for inserting, removing, or delaying slots during normal operation to maintain the network synchronised.

The synchronisation algorithm described in this work combines leader election with synchronisation in one step. The leader node divides the network into time slots and inserts the slot headers accordingly to make that explicit.

Because of the transparent nature of MOPS rings and the various possible transceiver configurations, synchronisation can only be achieved through signalling. Therefore, the algorithm defines a synchronisation header. The synchronisation header carries no payload, and it contains the following fields:

- SA: the address of the node that transmitted the header;
- Header type (HT): the type of the header;
- Time stamp (TS): the exact time the header was issued.

The algorithm works as follows. Upon initialisation, amongst other things, every node enters the synchronisation (SYNCHRO) state and starts a timer called time elapsed (TE). A node in the SYNCHRO state is forbidden to transmit or receive payload data.

If the node does not detect a header within MAX_WAIT_TIME, then the node generates a TS, saves it locally for future use, and issues a synchronisation (SYNC) header with SA updated with the node's address and TS updated with the generated time stamp.

If the node detects the arrival of a header, it reads HT to determine the type of the header. If HT is different than SYNC, meaning either that the node is joining a network that is already operational or that another node has already been elected leader, then the node switches to the OPERATIONAL state, thus becoming

allowed to transmit or receive payload slots. If HT equals SYNC, then the node reads SA to determine who issued the header.

If SA does not match the address of the node then the node reads TS to determine what to do with the header. If either the node has not issued a single SYNC header, or the received TS is smaller than the TS saved locally, then the node forwards the header and switches to the WAITING state. A node in this state waits for the arrival of a normal header for a pre-determined time, after which it switches back to the SYNCHRO state unless it switches to the OPERATIONAL state first.

If the received TS is greater than the TS saved locally, then the node discards the header, generates a new TS, and issues a SYNC header with SA and TS updated accordingly.

If SA matches the node's own address then the node switches to the OPERATIONAL state.

Figure B.1 depicts the FSM of the algorithm.

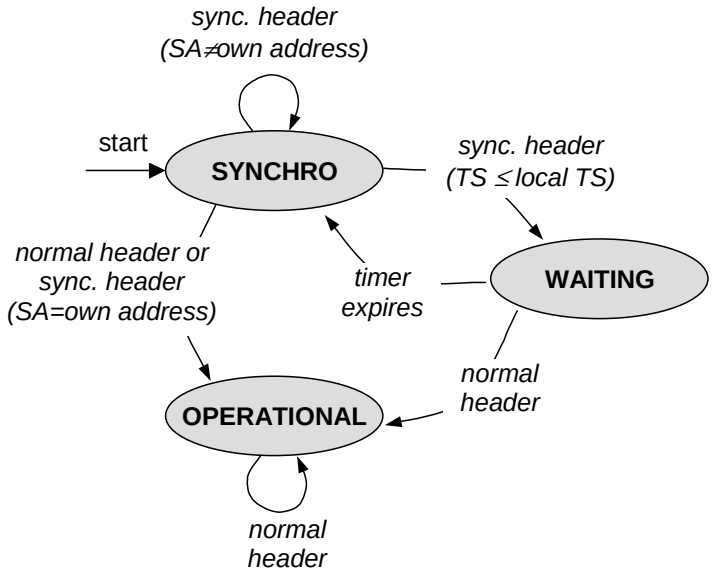


Figure B.1 - FSM of the synchronisation algorithm

Figure B.2 depicts the synchronisation algorithm. The terms SYNC and PAYLOAD denote a synchronisation header and a normal header.

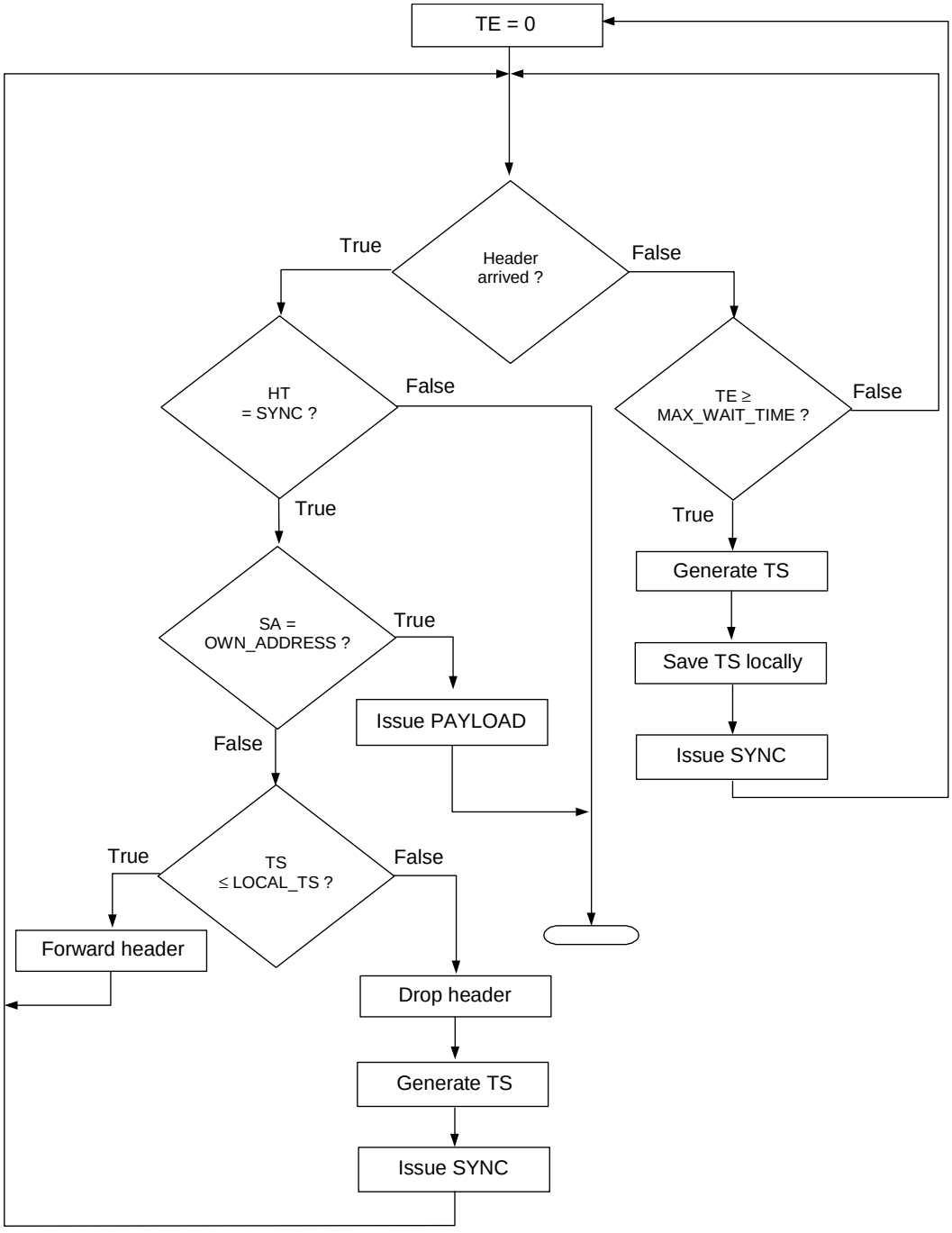


Figure B.2 - Synchronisation algorithm for MOPS rings

The second form of synchronisation involves dropping, adding, removing, and transmitting at the right time as well as controlling dispersion. Dropping of a slot takes place immediately after the processing of the corresponding header, and the removal of the corresponding payload occurs upon synchronisation with the first detected BOF sequence on the dropped channel. The addition of a slot occurs immediately after the selection of a packet for transmission.

In protocols that rely on the transmission of slot headers ahead of their corresponding payload slot, it might happen that a node detects an empty slot and gets ready to transmit before the current payload slot has past completely, consequently causing a collision.

Therefore, to avoid that a system parameter can be introduced to delay transmissions. In protocols such as SPT+P that parameter may have a positive value, but in protocols such as PAT that parameter contains the value zero. This way the solution becomes applicable to various protocols.

Dispersion becomes an issue in networks longer than 50km when transmissions traverse the entire network in the optical domain. Therefore, dispersion on the control channels is usually not an issue because headers are usually retransmitted by every node along the path, and the distance between any pair of adjacent nodes is rarely greater than 50km in LANs and MANs.

One way to cope with dispersion is to define a time gap between every two adjacent time slots based on the worst-case dispersion. This solution is simple, but it may lead to poor performance if the worst-case dispersion is too high. Other solutions are possible, but they are out of the scope of this work.

During normal operation each node is responsible for maintaining the header slots synchronised. To do that, each node keeps a timer per control channel, and it resets the appropriate timer each time it receives a header on the corresponding channel. Whenever a node wishes to insert a header it should do so at the time indicated by the timer.

Whenever a node processes an incoming header it delays that header by some amount of time. To make sure that a header is forwarded in time, each node maintains another timer whose expiration value equals the average processing delay. If the processing of a header ends before the expiration time then the node holds the header until the timer expires, but if the processing of that header ends when the timer expires, or after that, then the node forwards the header immediately.

To make sure that adjacent slots do not overlap, some time gap between adjacent slots should exist.

Appendix C
Group communication

This chapter is concerned with group communication in MOPS rings, and it defines a mechanism to support group communication services of the upper layers.

The protocols as described previously include no mechanisms to achieve group communication. Therefore, they can support point-to-multipoint communications transparently only when the network architecture offers a broadcast medium [Salva2001b], such as in Flamingo.

As discussed in Chapter 3, under certain architectures the only way to support multicast is by using unicast. Nevertheless, because of the transceiver configuration independence and heterogeneity, to deliver multicast packets of a given session to all the receivers, a source node needs to know not only the identity of the receivers and their location, but also on which channels those nodes can receive. What is more, the source node also needs to know whether a receiver can either drop or drop and forward at the same time. Based on such information the source node can construct routing trees and use multicast or unicast transmissions to deliver information to the receivers over each tree.

Each node should maintain a (local) group membership information base (GMIB) containing all the multicast sessions originated at that node associated with all the receivers that subscribed for those sessions.

Two types of mechanisms can be used to obtain such information: distributed and centralised. Distributed mechanisms proposed in the literature include [Qiao1999, Salva2000, Salva2001a, Salva2001c, Zhang1999]. The idea behind these mechanisms is to let either the source request for group membership information explicitly, or the receivers send join request messages towards the source. Nevertheless, these mechanisms are primarily designed for circuit-switched WDM networks, in which there is no need for protocol implementation in hardware. Therefore, the control channel can transport various types of messages.

A suitable solution for IP multicast over MOPS rings is that proposed for IP multicast over ATM in [Armit1996]. In this centralised solution there is a server that acts as a registry and maintains a global GMIB. A router that has a multicast receiver attached to it registers itself with the server informing its IP address and the group address of interest. A source node that has to forward multicast traffic of a certain session and needs to set-up a corresponding multicast tree consults with the server for the IP addresses of the receivers of that session.

Figure A.1 illustrates the mechanism. In Figure A.1a a router with an attached receiver registers with the multicast server for session #5. In Figure A.1b a router

receiving multicast traffic of session #5 interacts with the multicast server to obtain the addresses of the receivers of that session.

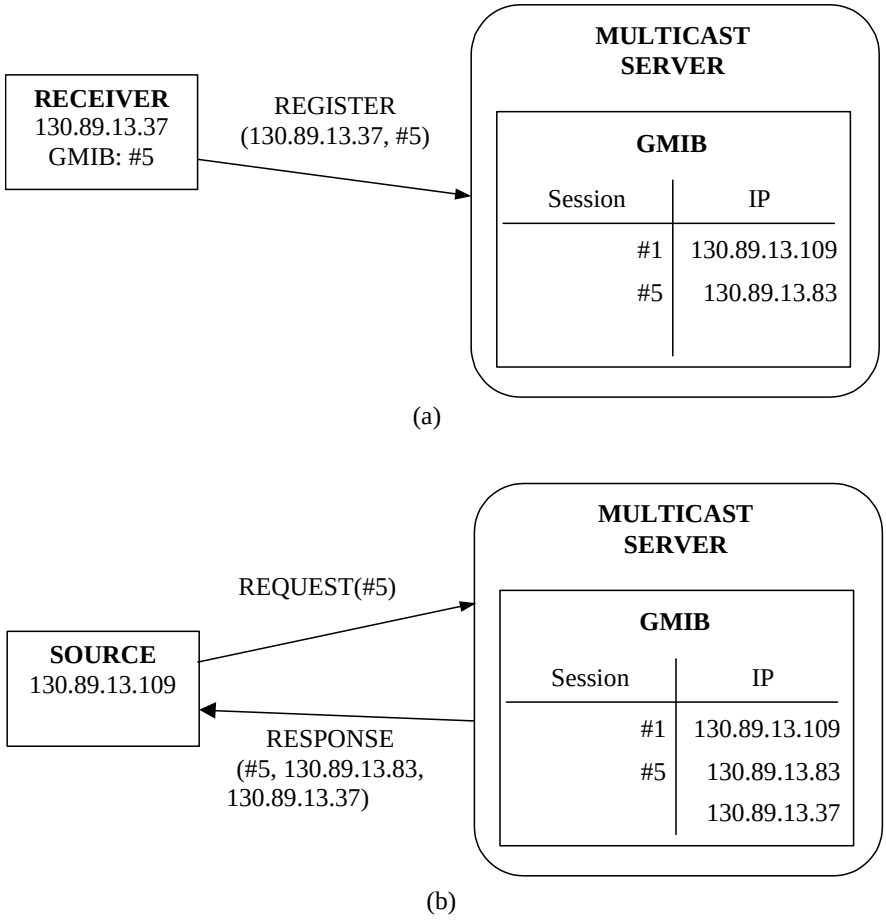


Figure A.1 - Example of interactions with multicast server

PAT and SPT+SC demand some special care with respect to multicast:

- PAT: in this protocol a single slot header determines the destination of one packet or more. Therefore, one-to-many traffic can only be multiplexed together with unicast traffic if transmitted as unicast;
- SPT+SC: because of the distance transmission constraint of SPT+SC, a source node can broadcast a packet only if the node immediately upstream, or the source node itself, holds the token.

Appendix D
Addressing

This chapter is concerned with addressing in MOPS rings, and it defines both MAC address and an address resolution mechanism for such networks.

The traditional network-layering model separates network protocols into layers, each with its own addressing mechanism. Thus, to allow for the communication between two peer protocol entities (PEs) above the MAC layer, each residing on a distinct node, some form of address mapping is necessary.

The protocol that provides such a mapping is known as address resolution protocol (ARP). There are two basic types of ARPs: distributed and centralised. Typically used in IP over broadcast medium networks (e.g., IEEE 802 family), in distributed ARP a source node that wishes to transmit and does not know the MAC address of the destination node broadcast an ARP REQUEST message informing its IP and MAC addresses and the IP address of the destination. Every node processes the message, but only the node whose IP address matches the IP destination address included in the message responds. The ARP RESPONSE message includes the data contained in the ARP REQUEST message plus the MAC address of the destination node.

Distributed ARP is simple and efficient in broadcast medium networks. Nevertheless, only the FLAMINGO network can provide a broadcast medium. Therefore, to work with any MOPS ring another solution is required.

The solution is centralised ARP. Typically used in IP over non-broadcast medium networks (e.g., ATM), in centralised ARP there is a server node that maintains the mappings between IP and MAC addresses. Nodes periodically register with the server informing their IP and MAC addresses. A source node that wishes to transmit sends a REQUEST message to the server informing its IP and MAC addresses and the MAC address of the destination node. The server stores the IP and MAC addresses of the source node for efficiency and sends a RESPONSE message back to the destination node informing the MAC address of the destination.

The only change necessary to the ARP used in IP over ATM is the MAC address. Instead of an ATM address, the ARP should use a specific MAC address for MOPS rings. Typically, networks use the standard IEEE 48-bit addresses. Nevertheless, MOPS rings add another addressing level: channel. To transmit to a given destination node a source node has to know both the destination's address and at least one of the channels that the destination can drop; note that ring networks always comprise two counter-rotating optical fibres, or more, for resilience. Therefore, optical fibre ring may add another address level into the problem.

The reception capability of the network nodes may vary or even be heterogeneous. For instance, a node A may be capable of receiving only on channel 1, a node B may be capable of receiving on any channel, but only one packet at a time, and a node C may be capable of receiving on all the channels simultaneously. It can be even possible for nodes to receive on certain ranges of channels.

Therefore, any scheme for the codification of the reception capability of a node should capture these particularities. This work codifies the reception capability of a node by means of capability description objects (CDOs). A CDO allows for the representation of a single Rx unit, and it contains the following fields:

- Coverage: 1-octet field that tells how many channels the Rx unit can drop at a time, hence also indicating whether the Rx unit is tuneable or not;
- Channel (CH) code [i]: 1-octet field that contains the channel code; up to 256 channels can be supported in this representation.

Figure D.1 depicts the structure of a CDO.

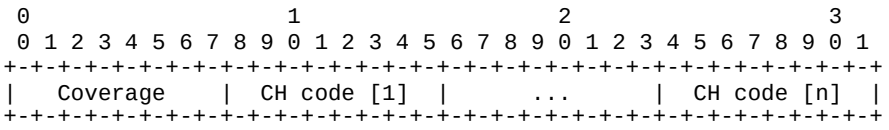


Figure D.1 – CDO layout

CDOs of the same node are multiplexed together for efficiency. The CDO multiplexer frame layout contains the following fields:

- Capability list length (CLL): 8-bit field that describes the number of capability description objects (CDOs) included;
- CDO [i]: variable size field that contains the *i*-th CDO.

Figure D.2 depicts the CDO multiplexer frame layout.

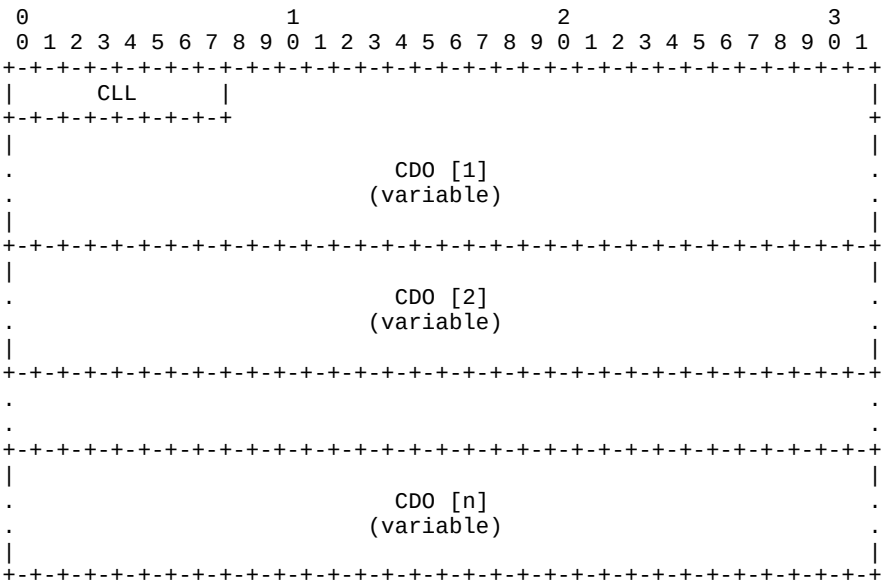


Figure D.2 - CDO multiplexer frame layout

Figure D.3 illustrates a node registering with the ARP server. The server maintains two databases, one containing the mapping between IP addresses and IEEE 802 addresses, called ARP table, and one containing the mapping between CDOs and IEEE 802 addresses, called transceiver information base (TIB).

Note that the REGISTER message includes the CDO multiplexer object in addition to the fields usually contained in that message.

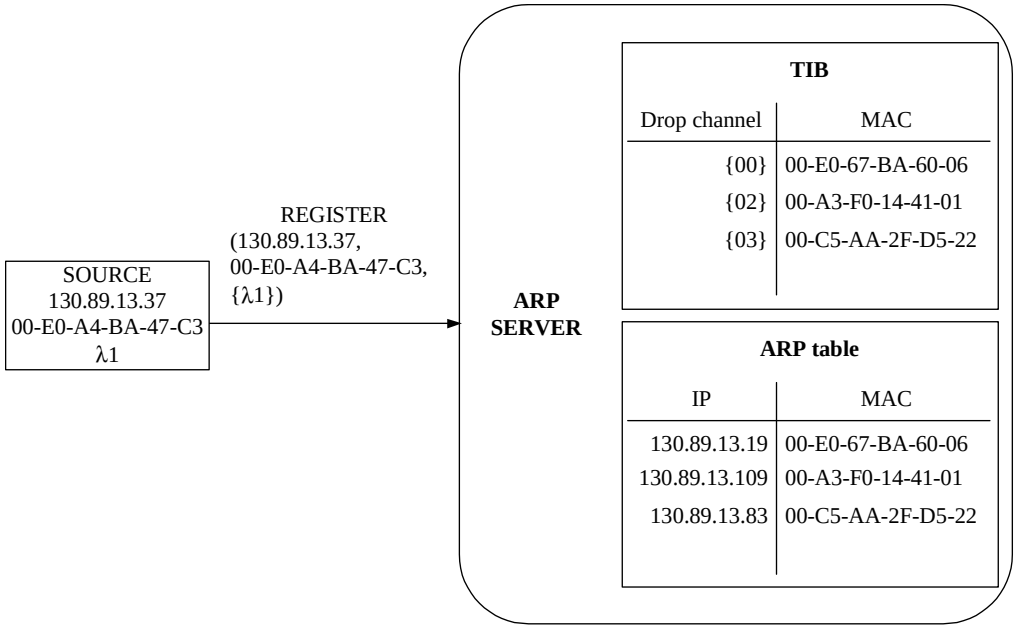


Figure D.3 – Example of register operation

Figure D.4 illustrates a node requesting a MAC address from the server. In addition to the fields usually contained in REQUEST and RESPONSE messages, these messages include the CDO multiplexer object; the inclusion of the CDO multiplexer object in the REQUEST message aims at efficiency. In the REQUEST message the CDO multiplexer object describes the reception capability of the source. In the RESPONSE message the CDO multiplexer object describes the reception capability of the destination.

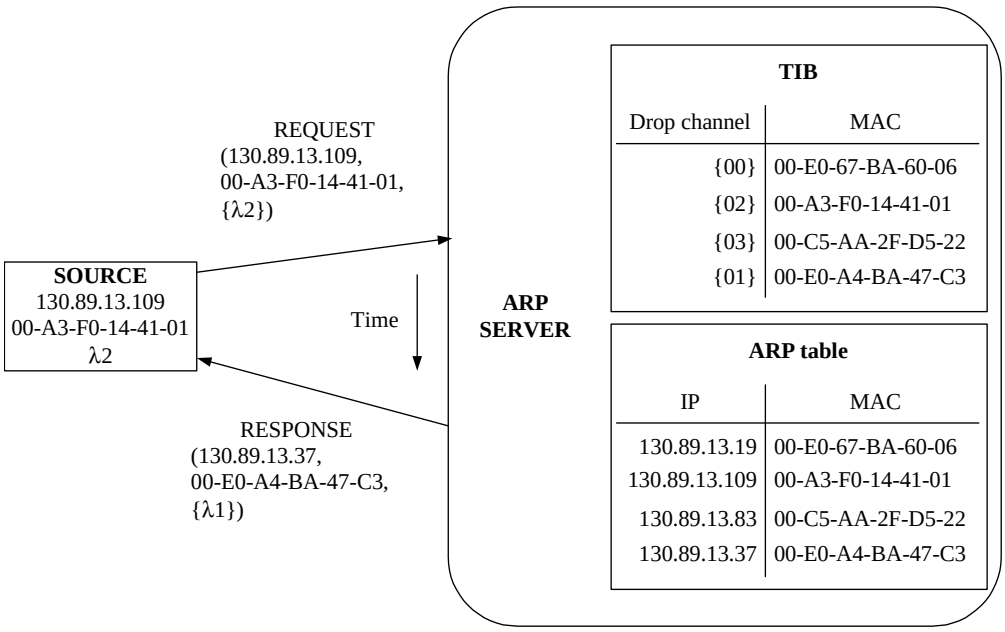


Figure D.4 – Example of request and response operations

Career resumé

Marcos Rogério Salvador was born in Americana, S.P., Brazil, on August 28, 1972. He obtained the bachelor degree in computer science at the Piracicaba Municipal School of Engineering (EEP), Brazil, in 1994, and the Master of Science degree, also in computer science, at Federal University of São Carlos (UFSCar), Brazil, in 1997.

During his studies, from 1989 to 1997, he also worked at Brazilian universities and companies as computer operator, network users support, and, eventually, as telecommunication systems senior analyst. In 1998 he joined the Centre for Telematics and Information Technology (CTIT) of the University of Twente, in The Netherlands, as research associate, to study medium access control protocols for optical packet-switched WDM ring networks. The results of his work in CTIT have been published in journals and conference proceedings on the subject, and they are presented in this Ph.D. dissertation.