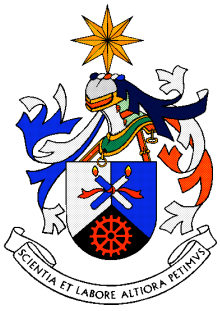
UNIVERSITY OF BEIRA INTERIOR
DEPARTMENT OF COMPUTER SCIENCE



Architectures and Algorithms for IPv4/IPv6-Compliant Optical Burst Switching Networks

Nuno Manuel Garcia dos Santos
(5-year Bachelor of Science)

Thesis submitted to the University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Computer Science and Engineering

Tese submetida à Universidade da Beira Interior para obtenção do Grau de Doutor em Engenharia Informática

Covilhã, Portugal
2008

UNIVERSITY OF BEIRA INTERIOR
DEPARTMENT OF COMPUTER SCIENCE

Architectures and Algorithms for IPv4/IPv6-Compliant Optical Burst Switching Networks

Nuno Manuel Garcia dos Santos
(5-year Bachelor of Science)

Thesis submitted to the University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Computer Science and Engineering

Tese submetida à Universidade da Beira Interior para obtenção do Grau de Doutor em Engenharia Informática

Covilhã, Portugal
2008

Thesis written under the supervision of Dr. Mário Marques Freire,
Associate Professor of Computer Science of the University of Beira Interior, Portugal, and
performed in enterprise environment under the
co-supervision of Dr. Paulo Miguel Nepomucemo Pereira Monteiro,
Head of Research, Nokia Siemens Networks Portugal SA.

Tese realizada sob a orientação do Doutor Mário Marques Freire,
Professor Associado do Departamento de Informática da Universidade da Beira Interior, com co-orientação em ambiente industrial do
Doutor Paulo Miguel Nepomucemo Pereira Monteiro,
Responsável pelo Grupo de Investigação, Nokia Siemens Networks Portugal SA.

FACIT MIHI MAGNA QVID POTENS EST.

Para o Nuno e para a Patrícia.

Acknowledgements

Gratitude is first and foremost due to my supervisors Prof. Mário M. Freire and Prof. Paulo P. Monteiro for their constant trust, guidance, advice and wisdom. Their vision for the future on so many aspects of the networking technology and their ability to discover relevant and interesting research areas are of immense value and paramount.

I am grateful to the University of Beira Interior for the opportunity to research in an enterprise environment while earning my PhD, and to Siemens S.A., Siemens Networks S.A., Nokia Siemens Networks Portugal S.A. and the Portuguese Fundação para a Ciência e Tecnologia (FCT) for the scholarship grants awarded during this period.

Gathering the concepts for the new architecture and implementing them on a simulator was a joint work with my friend and former colleague Przemyslaw Lenkiewicz, for which I am very grateful.

I am grateful to my colleagues at Nokia Siemens Networks Portugal S.A. for the frequent fruitful discussions and the joint inventions, in particular to Marek Hajduczenia and to Carlos Santiago. I am also extremely grateful for their friendship through a very enduring and challenging period of my life.

I am most grateful to all my family and friends for their permanent help, support, wise advice and friendship, in particular to Filomena and Carlos Marçal, my godparents who made possible for me to stay in Lisbon during this period, also to my father and mother-in-law João and Maria da Ascensão, and to my very good friends Maria José Silva (*in memoriam*), Jacinto Rebordão and Rita Rogado, and João Marcelo.

Utterly grateful, I can not thank enough my wife, Anabela, my mother, Alice, my son Nuno and my daughter Patrícia, and my brother Rui. Their love and permanent encouragement were the true reason that made me embrace and nourish this endeavour.

Preface

This thesis presents the research and its conclusions on Optical Burst Switched Networks with bursts based on IPv6 and IPv4 packets. The work described here started in 2004 and was developed over little more than three years of full time intense research activity within the Information and Communications organization, Research and Development Group 1 division, Research department (IC RD1 R) at Siemens SA (in 2006 the RD1 group became part of Siemens Networks S.A. and in 2007 this company merged with Nokia Networks, resulting in the new company Nokia Siemens Networks Portugal S.A.) in Alfragide, Amadora, Portugal, under the joint supervision of Professor Mário Marques Freire, Associate Professor at the Department of Computer Science of the University of Beira Interior, Covilhã, Portugal, and Professor Paulo Miguel Nepomucemo Pereira Monteiro, Director of the Research department of the Information and Communication Division, Research and Development 1 group, (IC RD1 R) in Siemens at Alfragide.

The research group at IC RD1 R in Siemens was formed by a group of around fifteen researchers and two doctorates. Interaction of the Research department with other departments within Siemens S.A. in Portugal and with Siemens AG (Germany) was frequent. The focus of its research activities ranged from the transport to the protocol layers of network science. The research group included Physicists, Electrotechnical Engineers, Informatic Engineers and Mathematicians. Siemens IC in Portugal had recently been distinguished as a worldwide international competence centre for Optical Networks within the Siemens group and its Optical Networks Labs rank amongst the first in both equipment and research activities within the company.

The Intellectual Property department at Siemens SA in Alfragide is responsible for the initial evaluation of new and innovative ideas reported in Invention Disclosures produced by the IC RD1 R group and by other Siemens employees. In Portugal, the evaluation of these internal reports was done with the goal of assessing its patent potential, firstly by a team of three intellectual property specialists and secondly, if the

Invention Disclosure showed potential, by a conjoint team of German and Portuguese specialists who ultimately evaluated the quality of the Invention Disclosures over several parameters and then decided on whether the idea was to be patented (and where) or not. The classification of the Invention Disclosures ranked from 1 to 6, from "not interesting" to "very disruptive invention". Just as a statistical note, on the 2004/2005 fiscal year, there were only two level 5 patents awarded to Siemens Portugal and since 2002 that no patent received a level 6 evaluation in Siemens worldwide. Additionally Siemens only considers patenting inventions that are susceptible to create financial revenue to the company within an acceptable time frame. After an Invention Disclosure is submitted to the Intellectual Property department and accepted to file as patent, there is a delay of about six to eighteen months before the Siemens AG Intellectual Property department allows for the publication of results in conferences or scientific journals. This restriction reflects directly on the amount of publications submitted by the author, particularly in scientific journals which usually have long acceptance cycles.

Throughout the research programme, the author submitted ten Invention Disclosures, some of which are described here as main contributions. In result of one of the patents, the author, conjointly with Marek Hajduczenia and Prof. Paulo P. Monteiro received the 2005 Siemens Research and Innovation Award. During its stay at Siemens IC RD1 R in Alfragide, the author worked cooperatively with the Systems Engineering department and conjointly proposed the production of one prototype machine, as defined in the Invention Disclosure on the Internet Protocol Packet Aggregator and Converter (IP-PAC) machine, currently submitted as an Innovation Project to the Nokia Siemens Networks AG, having successfully passed the initial evaluation by the global Nokia Siemens Networks Innovators Round Table.

The author was recipient of a joint scholarship awarded by **Siemens S.A. / Siemens Networks S.A. / Nokia Siemens Networks Portugal S.A.** and by the **Fundação para a Ciência e a Tecnologia** (FCT), through its grant contract number SFRH/BDE/15527/2004; the author was also a researcher of the FCT project **CONDENSA – IPv4/IPv6-Compliant Optical Burst Switching Network Design with Enhanced Signalling Architectures**, contract POSI/EEA-CPS/60247/2004.

Abstract

This thesis is devoted to Optical Burst Switched (OBS) networks, being focused on presenting new solutions to OBS network performance as a whole, from the ingress to the core nodes including the burst assembly, switching and routing tasks.

We present several new solutions to problems of the OBS networks, namely, a new burst assembly machine concept named Internet Protocol Packet Aggregator and Converter (IP-PAC), with a new burst assembly algorithm, dynamic and adaptive to network traffic fluctuations. Furthermore we propose the use of this machine not only in OBS networks but also as an aggregation device for other networks. We identify several benefits from the use of the IP-PAC concept, in particular its contribution in eliminating bottleneck problems.

We also present two new routing algorithms, Extended Dijkstra and Next Available Neighbour. The first is a balance and symmetry concerned algorithm that keeps the features of the Dijkstra Algorithm, *i.e.* it remains a shortest path simple algorithm. It is suitable to use in simulation as its behaviour tends to be closer to the behaviour of real networks since it does not overload unnecessarily some links more than others. It may also be used in machines where the computation of paths is Dijkstra based and situations of equal-cost routes may exist. Its performance is evaluated for OBS networks. The second algorithm is a dynamic non-deterministic routing algorithm that is applicable to OBS and to other networks. Its main feature is that in a situation of imminent burst loss, the burst is routed to another node, being this neighbour node the first available according to a series of metrics. As these metrics are used in a particular undetermined moment of time, the result is non-deterministic routing.

The study of the characteristics of burst traffic is of capital importance to understand the behaviour of OBS networks. As a pre-requisite to this aspect of the research, we study OBS networks tributary IP traffic, using a series of recorded real IPv4 packet traces. In this thesis, we conclude that the main OBS network performance

metric, burst loss ratio, is equivalent to other metrics like packet loss ratio and byte loss ratio when burst assembly is performed with efficiency concerned algorithms and using real traffic.

We assess the efficiency of main burst assembly algorithms and propose a new dynamic burst assembly algorithm, with thresholds that adapt to traffic conditions to allow an optimized burst assembly process. While assessing the assembly of bursts, we found that in real traffic conditions most bursts will be at around 9 KB of size. We also study the effect of the implementation of larger IP packets size for IPv6 and conclude that there are routing and switching benefits to reap from the usage of larger IP packets. Furthermore, we conclude that for burst assembly tasks, IPv4 and IPv6 behave similarly and thus the conclusions drawn on IPv4 datagrams can be extended to IPv6 packets, including Jumbograms.

The main contribution of this thesis is the proposal of a new OBS architecture, named Common Control Channel OBS, or C^3-OBS for short. In this architecture we propose the passive broadcast of the control packets in a special tree-like control channel, as a way to disseminate the information throughout the network. We then propose the use of a Local Network Model (LNM) database structure at each node to allow concise network management and behaviour prediction. In the C^3-OBS nodes we propose and test a new routing and scheduling algorithm we named Travel Agency Algorithm. We analyse some problems that rise in this new approach and propose solutions, a new approach using network domains for OBS as a way to minimize the flooding of the network with control packets, and some special features on the Travel Agency Algorithm as a way to identify and solve concurrent reservation situations. We assess and compare the performance of the C^3-OBS architecture with regular OBS architecture for several topologies.

Finally we present the research conclusions and propose directions for future work.

Resumo

Esta tese estuda as redes ópticas com comutação de agregados de pacotes (*Optical Burst Switched* - OBS), incidindo particulamente na apresentação de novas soluções para as redes OBS como um todo, desde os nós de ingresso onde são realizadas as tarefas de montagem dos agregados de pacotes, até aos nós nucleares, incluindo a comutação e o reencaminhamento.

São apresentadas várias soluções para problemas conhecidos em redes OBS, em particular, um novo conceito de máquina chamada IP-PAC (para *Internet Protocol Packet Aggregator and Converter*) com um novo algoritmo de montagem de agregados de pacotes, dinâmico e adaptativo às condições flutuantes do tráfego. Propõe-se ainda que o uso desta máquina não se limite às redes OBS, mas seja alargado a outro tipo de redes. São identificadas várias vantagens do uso do IP-PAC, em particular a sua contribuição na eliminação de problemas de engarrafamento de tráfego.

São ainda apresentados dois novos algoritmos de encaminhamento, o *Extended Dijkstra* e o *Next Available Neighbour*. O primeiro é um algoritmo melhor balanceado e simétrico do que o algoritmo de Dijkstra, mas que ainda assim se mantém como um algoritmo simples de caminho mais curto. A sua utilização é adequada em simuladores ou em máquinas onde se aplique a computação de caminhos usando algoritmos de caminho mais curto. O desempenho deste algoritmo é avaliado para redes OBS e apresentado nesta tese. O segundo algoritmo é dinâmico e não-determinístico, aplicável a redes OBS e a outros tipos de redes. A sua principal característica é que numa situação de perda de um agregado de pacotes por escassez de recursos no caminho que lhe foi atribuído, o algoritmo promove o reencaminhamento deste agregado para outro nó vizinho, sendo também este nó definido circunstancialmente de acordo com um conjunto de métricas que dependem do estado actual da rede. O resultado é um mecanismo de encaminhamento não determinístico.

O estudo das características do tráfego de agregados de pacotes foi determinante

na compreensão das redes OBS, tendo-se, para tal, usado um conjunto de dados relativos a tráfego real IPv4. Nesta tese é apresentada uma comparação entre os rácios de perda de agregados de pacotes, perda de pacotes e perda de bytes em redes OBS, concluíndo-se que quando estas redes usam algoritmos que são eficientes na produção de agregados de pacotes, estas métricas são equivalentes.

Também é apresentada a classificação de eficiência de algoritmos de agregação de pacotes e é proposto um novo algoritmo dinâmico, com limiares de agregação que variam de acordo com as condições do tráfego. Na avaliação da eficiência do processo de agregação dos pacotes, foi descoberto que em condições reais de tráfego, a maioria dos agregados de pacotes terão um tamanho ao redor dos 9 KB. Seguindo esta linha de investigação, foi avaliado o impacto do uso de maiores pacotes IP nas redes tributárias, tendo-se concluído que existem benefícios na utilização de maiores pacotes IP, em termos do esforço de encaminhamento e comutação. Dado que o comportamento dos pacotes IPv6 será semelhante ao dos datagramas IPv4, conclui-se que a agregação de pacotes pode ser feita de modo agnóstico quanto à versão do protocolo.

A principal contribuição desta tese é a proposta de uma nova arquitectura OBS, designada *Common Control Channel* OBS, ou abreviadamente C^3-OBS , a qual propõe a propagação automática dos pacotes de controlo por toda a rede como forma de disseminar a informação por toda a rede. É ainda proposto o uso de uma base de dados contendo um modelo local da rede (*Local Model Network* – LNM) em cada nó, como forma de permitir a gestão da rede e dos pedidos de reserva de recursos. Sobre este LNM é proposta a implementação de um novo algoritmo de encaminhamento e de gestão de reserva de recursos chamado *Travel Agency Algorithm* (TAA). São identificados e analisados os problemas que surgem da utilização da arquitectura C^3-OBS e são propostas soluções: o uso de domínios de rede em OBS como forma de minimizar a inundação da rede por pacotes de controlo e novas técnicas implementadas no TAA como forma de identificar e resolver situações de reserva concorrente. É feita a avaliação e comparação das redes C^3-OBS com as redes OBS.

Finalmente, são apresentadas as conclusões desta investigação e propostas algumas sugestões para trabalho futuro.

Keywords

Optical Burst Switching Networks, Optical Burst Switching Network Architectures, Optical Internet, Burst Assembly Algorithms, IPv4 and IPv6 Burst Assembly, Routing Algorithms

Palavras Chave

Redes Ópticas com Comutação de Agregados de Pacotes, Arquitecturas de Redes Ópticas com Comutação de Agregados de Pacotes, Internet Óptica, Algoritmos de Montagem de Agregados de Pacotes, Montagem de Agregados de Pacotes IPv4 e IPv6, Algoritmos de Encaminhamento.

Contents

Acknowledgements ..... **ix**

Preface ..... **xi**

Abstract ..... **xiii**

Resumo ..... **xv**

Keywords ..... **xvii**

Contents ..... **xix**

List of Figures ..... **xxiv**

List of Tables ..... **xxxii**

Abbreviations and Acronyms ..... **xxxiii**

Extended Abstract in Portuguese ..... **xxxix**

Chapter 1. Introduction **1**

1.1. Thesis focus and scope 1

1.2. Problem statement and goals of the research 3

1.3. Organization of the thesis 4

1.4. Main contributions for the advance of the scientific knowledge 6

Chapter 2. Optical Burst Switching Networks **9**

2.1. Introduction 9

2.2. Burst switching 10

2.3. The optical burst switching network concept 11

2.4. Optical burst switching network architectures 18

2.4.1. Basic architecture of an optical burst switching network 18

2.4.2. Dynamic wavelength-routed optical burst switching network architecture 20
2.4.3. Other architectures 20
2.5. Burst assembly of IP traffic 21
2.6. Resource reservation protocols 24
2.6.1. Types of resource reservation protocols in OBS networks 24
2.6.2. Immediate versus delayed reservation 34
2.6.3. Void filling versus no void filling 36
2.6.4. Just Enough Time (JET) resource reservation protocol 37
2.6.5. Horizon resource reservation protocol 38
2.6.6. Just in Time (JIT) resource reservation protocol 38
2.6.7. JumpStart resource reservation protocol 39
2.6.8. JIT^{+} resource reservation protocol 41
2.6.9. E-JIT resource reservation protocol 41
2.7. Approaches for contention resolution in OBS 43
2.7.1. Prioritization 43
2.7.2. Optical buffering 44
2.7.3. Burst segmentation 45
2.7.4. Burst and traffic grooming 45
2.7.5. Routing strategies and algorithms 46
2.7.6. Wavelength assignment algorithms 48
2.8. IP over WDM 50
2.9. IP over OBS 51
2.10. TCP over OBS 53
2.11. Summary 54

Chapter 3. Routing Algorithms for Optical Burst Switched Networks 55
3.1. Introduction 55
3.2. Main routing algorithms and strategies 55
3.3. Static and dynamic routing 56
3.3.1. Static routing 58
3.3.2. Extended Dijkstra routing algorithm 61
3.3.3. The Travel Agency algorithm 78

3.4. Dynamic routing strategies 79
3.4.1. Deflection Routing 79
3.4.2. Next Available Neighbour (NAN) routing algorithm 80
3.5. Summary 89

Chapter 4. Burst Assembly Algorithms and its Performance Evaluation for IPv4 and IPv6 91
4.1. Introduction 91
4.2. Main burst assembly algorithms 92
4.2.1. Maximum Burst Size assembly algorithm 92
4.2.2. Maximum Time Delay assembly algorithm 93
4.2.3. Hybrid Assembly algorithm 94
4.3. Real IP traffic 95
4.3.1. Real IPv4 traffic 96
4.3.2. Examined data set 97
4.3.3. Generated IPv6 traffic 98
4.3.4. Results 102
4.3.5. Header replacement algorithm 104
4.3.6. Payload reconstruction algorithm 105
4.4. Burst assembly simulation and evaluation metrics 108
4.4.1. Evaluation metrics 109
4.4.2. Burst assembly variables 110
4.4.3. Results 110
4.5. Relevance of OBS network performance metrics 121
4.5.1. Basic assumptions and burst assembly algorithms 122
4.5.2. Burst assembly simulation 123
4.5.3. Burst loss versus packet loss 124
4.5.4. Conclusions 127
4.6. Summary 128

Chapter 5. IP Packet Aggregator and Converter Machine Concept 129
5.1. Introduction 129
5.2. IPv6 burst transmission motivation 129

5.3. Machine concept 131
5.3.1. Placement of an IP-PAC in an IP network 131
5.3.2. Communication between an IP-PAC machine and a non-IP-PAC machine 133
5.3.3. Communication between IP-PAC machines 134
5.4. IP-PAC burst assembly algorithm 135
5.5. Simulation and results 137
5.6. Discussion and Conclusion 145
5.7. Summary 148

Chapter 6. Architecture and Performance Evaluation of Common Control Channel Optical Burst Switched Networks 149
6.1. Introduction 149
6.2. Architecture of a C^3-OBS network 150
6.2.1. Architecture of core nodes 150
6.2.2. Common control channel 154
6.2.3. The Local Network Model 159
6.2.4. The Travel Agency Algorithm 161
6.2.5. Well informed nodes and degree of network awareness of nodes 162
6.2.6. Scope, limitations and proposed solutions 167
6.3. IP-PAC and burst assembly in C^3-OBS 169
6.4. C^3-OBS domains 172
6.5. Modelling and simulation of OBS and C^3-OBS networks 177
6.5.1. Performance assessment through simulation 177
6.5.2. Burst traffic model 179
6.5.3. Simulator architecture 180
6.5.4. Simulator validation 182
6.5.5. Simulation results 184
6.6. Performance assessment of OBS networks 186
6.6.1. Performance assessment of EON and NSFnet with OBS and C^3-OBS 191
6.6.2. Burst delay in OBS and C^3-OBS real topologies 197

6.6.3. NAN routing for OBS and C^3-OBS 200
6.7. Summary 202

Chapter 7. Final Conclusions and Future Work..... 203
7.1. Final conclusions 204
7.2. Future work 208

Annex A OBS simulator architecture 203

References 223

List of Figures

Figure 1 – Schematic representation of an OBS network. 15
Figure 2 – A schematic representation of an edge node. 16
Figure 3 – A schematic representation of a core node. 16
Figure 4 – An integrated IP over WDM architecture as proposed in LOBS [65]. 21
Figure 5 – Lightnet Architecture sample with three homogeneous lightpaths. 25
Figure 6 - Control packet structure as proposed by Oh and Kang in [86]. 27
Figure 7 – Classification of resource reservation protocols. 28
Figure 8 – Schematic representation of messages and burst transmission for a Tell and Wait OBS network with five nodes. 29
Figure 9 – Schematic representation of messages and burst transmission in an Intermediate Node Initiated (INI) Reservation OBS network with five nodes (configuration of OXC is not shown for all the nodes). 32
Figure 10 – Schematic representation of the signalling messages in an immediate reservation protocol (*e.g.* JIT, JIT$^+$ and JumpStart) OBS network with five nodes. 35
Figure 11 – Schematic representation of the signalling messages in a delayed reservation protocol (*e.g.* JET, Horizon and JumpStart) OBS network with five nodes. 35
Figure 12 – Scheduling two bursts using no-void-filling signalling protocols (*e.g.* Horizon, JIT or JIT$^+$). 36
Figure 13 – Scheduling two bursts using void-filling protocol JET. 36
Figure 14 – Schematic representation of messages and burst transmission depicting implicit or estimated release in an immediate reservation scenario for an OBS network with five nodes. 40
Figure 15 – Schematic representation of messages and burst transmission depicting explicit release in an immediate reservation scenario for an OBS network with five nodes. 40
Figure 16 – Scheduling three burst in JIT$^+$. 41

Figure 17 – Operation of E-JIT resource reservation protocol (rejecting a burst). 43
Figure 18 – Wavelength Path scheme showing five paths using different wavelengths. 47
Figure 19 – Virtual Wavelength Path scheme showing five paths using different wavelengths in the same path. 48
Figure 20 – Approaches for IP over WDM. 51
Figure 21 - IP-over-OBS proposed hierarchical model [115]. 52
Figure 22 - IP-over-OBS proposed data burst structure [115]. 53
Figure 23 - IP-over-OBS proposed control packet generic structure [115]. 53
Figure 24 - IP-over-OBS proposed Burst Header Packet structure [115]. 53
Figure 25 – Schematic representation of an OBS network with 6 nodes and 9 links. 57
Figure 26 – The 11 nodes COST 239 / EON network. 61
Figure 27 – The 14 nodes NSFnet network topology. 62
Figure 28 – Four nodes ring network. 66
Figure 29 – Dijkstra algorithm routes (version A and version B) in a four nodes ring network. 66
Figure 30 – Seven nodes network with nine bidirectional links. 68
Figure 31 – Routes created by the Extended Dijkstra algorithm in a four node ring topology. 71
Figure 32 – Number of created routes per link and direction in a seven nodes topology with 3 four-node ring networks sharing links obtained with the Dijkstra algorithm. 73
Figure 33 – Number of created routes per link and direction in a seven nodes topology with 3 four-node ring networks sharing links obtained with the Extended Dijkstra algorithm. 73
Figure 34 – Number of created routes per link and direction in a nine-node topology with 4 four-nodes ring networks sharing links obtained with the Dijkstra algorithm. 74
Figure 35 – Number of created routes per link and direction in a nine-node topology with 4 four-nodes ring networks sharing links obtained with the Extended Dijkstra algorithm. 74

Figure 36 – Burst loss probability versus routing algorithm for three burst loss scenarios (around 50%, 10% and 1%) for four network ring and ring based topologies. 75
Figure 37 – Comparison of burst loss probability versus number of data channels for the eleven nodes COST 239 topology OBS network, using the Dijkstra and the Extended Dijkstra routing algorithms. 75
Figure 38 – Standard deviation on the number of routes created over links for the Dijkstra and the Extended Dijkstra routing algorithms, for several topologies. 76
Figure 39 – Symmetry test for the Dijkstra (D) and Extended Dijkstra routing algorithms (ED), showing mean number of routes in each direction and overall mean and maximum and minimum number of routes per link for several topologies. 77
Figure 40 – Schematic representation of an OBS network with Deflection Routing (CPs are not depicted). 80
Figure 41 – Sample MPLS network topology with three sample virtual routes. 84
Figure 42 – Next Available Neighbour routing example (CPs are not depicted). 85
Figure 43 – Performance comparison for an OBS network with Next Available Neighbour routing and with shortest path routing only for the NSFnet topology with 14 nodes and 1 km links, showing burst loss probability versus number of data channels. 88
Figure 44 – Flowchart for the Maximum Burst Size algorithm. 93
Figure 45 – Flowchart for the Maximum Time Delay algorithm. 94
Figure 46 – Flowchart for the Hybrid Assembly algorithm. 95
Figure 47 – Internal format of the *tsh* data packet format from NLANR [148]. 97
Figure 48 – IPv4 packet size (partial graph for the TXS-1123774395 trace). 98
Figure 49 – Sample IPv4 to IPv6 packet conversion under the Header Replacement algorithm. 101
Figure 50 – Sample IPv4 to IPv6 conversion under the Original Payload Retrieval algorithm. 102
Figure 51 – Maximum, minimum and average number of packets in the selected file traces per network collection point. 103

Figure 52 – Maximum, minimum and average percentage of packets bigger than 1500 bytes in the selected traces. 103
Figure 53 – Increase on the number and size of packets in the selected traces, converted from IPv4 to IPv6 by the application of the header replacement algorithm. 104
Figure 54 – Percentage of increase on the number of packets for the conversion of IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 μs, for MTU=1500 bytes. 106
Figure 55 – Percentage of increase for the conversion of IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 μs, for MTU=9K bytes. 107
Figure 56 – Weighted percentage of increase on the number of packets for all traces for the conversion from IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 μs, for MTU=9K bytes and MTU=64K bytes. 107
Figure 57 – Complementary Cumulative Distribution function for burst inter-arrival time for MBS threshold = 9 KB. 112
Figure 58 – Weighted average burst inter-arrival time versus network load for MBS. 112
Figure 59 – Average packet delay per burst for MBS. 113
Figure 60 – Burst inter-arrival time versus burst assembly threshold time for MTD. . 114
Figure 61 – Number of packets in bursts versus burst assembly (aggregation) thresholds for MTD. 114
Figure 62 – Burst Size Histogram for MDT when time = 10^4 s..... 115
Figure 63 – Number of packets in bursts relative distribution for MTD = 100 s..... 116
Figure 64 – Average delay of packets in bursts versus burst assembly threshold times. 116
Figure 65 – Burst inter-arrival time versus burst assembly scenarios for HA, considering three network collection points (AMP = AMPATH, Miami, Florida, USA, ANL = Argonne National Laboratory to STARTAP, MEM = University of Memphis)..... 118
Figure 66 – Burst size histogram for HA (T=10^3 s, S=9 KB) on AMP data. 118

Figure 67 – Performance comparison for the burst assembly algorithms versus burst assembly scenarios, depicting both average burst size (solid line plots) and average packet delay (dotted line plots). 121
Figure 68 – Standard deviation ratio of average burst size (measured in bytes) and average burst size (measured in number of packets). 124
Figure 69 – Burst, packet and byte loss ratios for different burst assembly scenarios in an OBS JIT four nodes ring network (time thresholds in x-axis are s). 126
Figure 70 – Burst, packet and byte loss ratios for different burst assembly scenarios in an OBS JET four nodes ring network, 64 KB burst size threshold, 40 s time threshold and variable number of data channels (in x-axis of the graph)..... 126
Figure 71 – Schematic representation of a utilization of IP-PAC machines at the edges of a core network, with non-IP-PAC machines. 132
Figure 72 – Scheme of the format of an IP-PAC generated packet..... 132
Figure 73 – IP-PAC as gateway depicting several burst assembly queues (Q1 to Qn). 134
Figure 74 – IP-PAC machines as ingress and egress nodes in a backbone..... 134
Figure 75 – Generic Hybrid Burst Assembly algorithm with dynamic threshold control. 136
Figure 76 - Packet or burst count per data trace versus various aggregation scenarios (times T are in s and sizes S are in bytes)..... 138
Figure 77 – IP-PAC packet count per data trace for various burst assembly time thresholds (MEM series of data traces). 139
Figure 78 – IP-PAC packet compression ratio for various burst assembly time thresholds (MEM series of data traces). 140
Figure 79 – IP-PAC number of bytes in burst transmissions for various burst assembly time thresholds (MEM series of data traces). 140
Figure 80 – IP-PAC byte compression ratio for various burst assembly time thresholds (MEM series of data traces). 141
Figure 81 – Simulator version 3.2 after running a simulation..... 142
Figure 82 – Schematic representation of an OBS architecture..... 150

Figure 83 – Example of a signalling and burst transmission in a C^3-OBS network..... 152
Figure 84 – Schematic illustration of the block architecture for the C^3-OBS core node..... 154
Figure 85 – Schematic illustration of a mesh network with 6 nodes and 8 links (upper scheme) and one possible control channel topology (lower scheme). 156
Figure 86 – Schematic illustration of C^3-OBS OXC in a topology showing open and closed switches for the control channel..... 156
Figure 87 – Sequence of propagation of a control packet in a sample network for C^3-OBS..... 158
Figure 88 – Sample Hamiltonian control channel topology for the sample network in Figure 85 for C^3-OBS. 159
Figure 89 – Sequence of propagation of a control packet issued by node 1 for the sample network in Figure 85 for C^3-OBS using the Hamiltonian control channel topology in Figure 88. 159
Figure 90 – Evolution of the degree of network-awareness for a core node in a C^3-OBS network. 163
Figure 91 – Sample concurrent control packet event. 166
Figure 92 – Bursts in a concurrent control packet scenario at Node 2. 166
Figure 93 –Number of messages per second in the control channel versus link length versus message size. 168
Figure 94 – Manageable aggregated bandwidth versus CP size for three control message sizes (100, 200 and 300 bytes). 169
Figure 95 – Sample C^3-OBS network with IP-PAC machines as edge nodes. 172
Figure 96 – Example of domains in C^3-OBS 174
Figure 97 – Schematic illustration of the block architecture for the frontier C^3-OBS node..... 177
Figure 98 – UML Class Diagram for the Network package. 181
Figure 99 – UML Class Diagram for the Event List package. 181
Figure 100 – Burst loss probabilities as a function of the number of available data channels, comparing the results published in [84] and obtained by

simulation when the mean burst size was defined equal to T_{OXC}, for current technology values. 183
Figure 101 – Burst loss probabilities as a function of the number of available data channels, comparing the results published in [84] and obtained by simulation when the mean burst size was defined equal to $5.T_{OXC}$, for current technology values. 183
Figure 102 - Burst loss probabilities as a function of the number of available data channels, obtained by simulation when the mean burst size was defined equal to $5.T_{OXC}$, for current technology values, showing confidence intervals. 186
Figure 103 – Burst loss probability versus number of data channels for three and five nodes bus topologies for OBS and C^3-OBS. 189
Figure 104 – Burst loss probability versus number of data channels for four, six and eight nodes ring topologies for OBS and C^3-OBS. 189
Figure 105 – Topology for the eight nodes ring network (left scheme) and the implemented control channel topology (right scheme). 190
Figure 106 – Topology for the eight nodes ring network with two chords (left scheme) and the implemented control channel topology (right scheme). . 190
Figure 107 – Burst loss probability versus number of data channels for eight nodes ring topology and eight nodes with cross-connections (second with sixth node, fourth with eighth node) for OBS and C^3-OBS. 191
Figure 108 – The 19-node European Optical Network (EON) topology. 192
Figure 109 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (full scaled links). 192
Figure 110 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (1 km links). 194
Figure 111 – Concurrent Reservation occurrence probability for 19-node EON with full-scale links and with short equal-length links of 1 km. 194
Figure 112 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (scaled reduced links and scaled full length links). 195

Figure 113 – Burst loss probability versus number of data channels for 14-node NSFnet topology for OBS and C^3-OBS (scaled full length links). 196
Figure 114 – Burst loss probability versus number of data channels for EON topology for C^3-OBS for different Departure Horizon delays (1 km length links, 4 searchable paths)..... 196
Figure 115 – OBS network mean transmission time for a burst versus number of available data channels in 4x4 Mesh Torus, 19-node EON and 14-node NSFnet topologies. 198
Figure 116 – C^3-OBS architecture mean burst transmission time versus number of available data channels for a burst in 4x4 Mesh Torus, 19-node EON and 14-node NSFnet topologies. 198
Figure 117 – Burst loss probability versus number of data channels showing C^3-OBS and C^3-OBS with NAN routing for the 14-node NSFnet topology (1 km length links)..... 201
Figure 118 – Burst loss probability versus number of data channels showing C^3-OBS and C^3-OBS with NAN routing for the 14-node NSFnet topology (full scaled link lengths)..... 201

List of Tables

Table 1 – Burst transmission time (in s) for different transmission ratios and different burst sizes. 44
Table 2 – Routing table for a ring network with four nodes. 67
Table 3 – Route Matrix for the seven node nine links topology for Dijkstra. 69
Table 4 – Route Matrix for the eight nodes ring topology for Dijkstra. 71
Table 5 – Route Matrix for the eight nodes ring topology for Extended Dijkstra. 72
Table 6 – Route Matrix for the eleven nodes COST 239 topology for Dijkstra. 72
Table 7 – Route Matrix for the eleven nodes COST 239 topology for Extended Dijkstra. 72
Table 8 – Characteristics of data collection sites. 98
Table 9 – Network Activity for the selected trace files. 111
Table 10 – Simulation results for an OBS network with Aggregated Packets Statistics (Transmission). 143
Table 11 – Simulation results for an OBS network with Aggregated Packets Statistics (Reception) 144
Table 12 – Values for simulation parameters for OXC configuration and node setup times. 184
Table 13 – Sample network configuration file for the 4 node ring network. 217
Table 14 – Sample simulation results for a short simulation run. 221

Abbreviations and Acronyms

10GE	:	10 Gigabit per second Ethernet
AAL5	:	ATM Adaptation Layer 5
ABT	:	ATM Block Transfer
ACK	:	Acknowledgement Message
ARDA	:	Advanced Research and Development Agency
ATM	:	Asynchronous Transfer Mode
BER	:	Bit Error Rate
BFC	:	Burst Framing Control
BHP	:	Burst Header Packet
C^3-OBS	:	Common Control Channel Optica Burst Switching
CAPEX	:	Capital Expenditure
CHDLC	:	Cisco High Level Data Link
CoS	:	Class of Service
COST	:	European Cooperation in the field of Scientific and Technical Research
CP	:	Control Page
CRC	:	Cyclic Redundancy Code
DA	:	Dijkstra's Algorithm
DCP	:	Double Control Packet
DiffServ	:	Differentiated Services
DND	:	Domain Network Diameter
DS-3	:	Digital Service Level 3 carrier, its data rate is 44.736 Mbit/s
DWDM	:	Dense Wavelength Division Multiplexing
DWR-OBS	:	Dynamic Wavelength-Routed Optical Burst Switching
E/O	:	Electronic to Optic conversion
ECMP	:	Equal Cost Multi Path
E-JIT	:	Enhanced Just in Time

EON	:	European Optical Network
FCCN	:	Portuguese Fundação para a Computação Científica Nacional
FCFS	:	First Come First Served
FDDI	:	Fibre Distributed Data Interface
FDL	:	Fibre Delay Lines
FIFO	:	Fist In First Out
Gb/s	:	Gigabit per second
GMPLS	:	Generalized Multi Protocol Label Switching
HA	:	Hybrid Algorithm
HBP	:	Header Burst Packet
HTTP	:	Hypertext Transfer Protocol
ICMPv6	:	Internet Control Message Protocol version 6
ID	:	Identification
IETF	:	Internet Engineering Task Force
IETF	:	Internet Engineering Task
IGP	:	Interior Gateway Protocol
INI	:	Intermediate Node Initiated
IntServ	:	Integrated Services
IP	:	Internet Protocol
IPoDWDM	:	Internet Protocol over Dense Wavelength Division Multiplexing
IPoWDM	:	Internet Protocol over Wavelength Division Multiplexing
IP-PAC	:	Internet Protocol – Packet Aggregator and Converter
IPPM	:	Internet Protocol Performance Metrics group (IETF)
IPv4	:	Internet Protocol version 4
IPv6	:	Internet Protocol version 6
IPvFuture	:	Internet Protocol version Future
ISO	:	International Organization for Standardization
ISP	:	Internet Service Providers
ITU-T	:	International Telecommunication Union – Telecommunication Standardization Sector
JET	:	Just Enough Time

JIT	:	Just In Time
JIT^{+}	:	Just In Time +
JITPAC	:	Just-in-Time Protocol Acceleration Circuit
KB	:	Kilo Byte
km	:	kilometre
LAN	:	Local Area Network
LAUC	:	Latest available unscheduled channel
LAUC-VF	:	Latest available unscheduled channel with void filling
LLC	:	Logical Link Control
LNM	:	Local Network Model
LOBS	:	Labeled Optical Burst Switching
MAC	:	Medium Access Control
MAN	:	Metropolitan Area Network
MB	:	Mega Byte
Mb/s	:	Megabit per second
MBS	:	Maximum Burst Size
MEMS	:	Micro-Electro-Mechanical Systems
MPλS	:	Multi Protocol Lambda Switching
MPLS	:	Multi Protocol Label Switching
MTD	:	Maximum Time Delay
MTU	:	Maximum Transmission Unit
NAK	:	Negative Acknowledgement
NAN	:	Next Available Neighbour
NCD	:	Network Control Diameter
NCSU	:	North Carolina State University
NGN	:	Next Generation Network
NGXC	:	Next Generation Cross Connects
NGXC/POS	:	Next Generation Cross Connects for Packet over SONET
NGXN/10GE	:	Next Generation Cross Connects for Packet over 10 Gigabit per second Ethernet
NLANR	:	National Laboratory for Applied Network Research
NNTP	:	Network News Transfer Protocol

ns-2	:	Network simulator version 2
NSF	:	USA National Science Foundation
NSFnet	:	National Science Foundation network
O/E	:	Optical to Electronic conversion
O/E/O	:	Optical to Electronic to Optical conversion
OBS	:	Optical Burst Switching
OC12c	:	Optical Carrier with 12 concatenated DS-3 circuits, approximately 622 Mb/s
OC3	:	SONET standard Optical Carrier, approximately 155 Mb/s
OC3c	:	SONET standard Optical Carrier, with 3 concatenated DS-3 circuits, approximately 155 Mb/s
OCS	:	Optical Circuit Switching
OPEX	:	Operational Expenditure
OPS	:	Optical Packet Switching
OSI	:	Open Systems Interconnection
OSPF	:	Open Shortest Path First
OXC	:	Optical Cross Connect
P2P	:	Peer to Peer
PAD	:	Packet Aggregation and De-aggregation
pJET	:	prioritized JET
POP3	:	Post Office Protocol 3
POS	:	Packet over SONET
QoS	:	Quality of Service
QoS	:	Quality of Service
RAM	:	Random Access Memory
RFCs	:	Request for Comments
RTP	:	Real Time Protocol
RWA	:	Route and Wavelength Assignment
RX	:	Reception
SAN	:	Storage Area Network
SCC	:	Signalling Connection Control
SDH	:	Synchronous Digital Hierarchy

SE	:	Signalling Engine
SFC	:	Signalling Framing Control
SLOB	:	Switch with large optical buffers
SMTP	:	Simple Mail Transfer Protocol
SNAP	:	Subnetwork Access Protocol
SONET	:	Synchronous Optical Network
STP	:	Spanning Tree Protocol
TAA	:	Travel Agency Algorithm
TAG	:	Tell And Go
TAW	:	Tell And Wait
TCP	:	Transport Control Protocol
TDM	:	Time Division Multiplexing
Terabit/s	:	Terabit per second
ToS	:	Type of Service
tsh	:	Time stamped header
TX	:	Transmission
TXP	:	Transponder Based Optical Layer
UDP	:	User Datagram Protocol
UML	:	Unified Modeling Language
VoIP	:	Voice over Internet Protocol
VPLS	:	Virtual Private LAN Service
WAN	:	Wide Area Network
WDM	:	Wavelength Division Multiplexing
WR-OBS	:	Wavelength-Routed Optical Burst Switching

Extended Abstract in Portuguese

Introdução

Esta secção apresenta, em português, o resumo alargado da tese de doutoramento intitulada "Arquitecturas e Algoritmos para Redes com Comutação Óptica de Agregados de Pacotes IPv4/IPv6" ("*Architectures and Algorithms for IPv4/IPv6-Compliant Optical Burst Switching Networks*").

Este resumo alargado tem a seguinte organização: em primeiro lugar, são apresentados os objectivos desta tese e as suas principais contribuições para o avanço da Ciência, de acordo com a opinião do autor. Seguidamente apresenta-se o enquadramento da tese e descrevem-se as conclusões relevantes das contribuições. A finalizar são descritas as conclusões deste trabalho e propostas linhas de investigação para trabalho futuro. Neste resumo as referências não estão numeradas sequencialmente para manter a coerência da numeração com o texto em inglês.

Descrição do problema e objectivos desta tese

Considerando que: as redes que transportam o protocolo Internet Protocol (IP) estão largamente disseminadas, que a transição da versão 4 do protocolo IP (IPv4 [9]) para a versão 6 (IPv6 [10]) está em curso há algum tempo, sendo esperado que a versão 6 do protocolo IP seja dominante num futuro próximo, e ainda que as redes ópticas de alta capacidade capazes de operar de modo flexível constituirão a principal tecnologia nas redes nucleares (em inglês, ditas *core networks*), e que a tecnologia de comutação óptica está limitada pela inexistência de memória óptica de acesso aleatório viável e tamanho e granularidade adequada, pela inexistência de lógica óptica capaz de, por exemplo, assegurar a interpretação e sincronização dos dados no domínio puramente óptico, e que o tempo que é necessário para configurar um elemento de comutação óptica tecnologicamente recente é ainda considerável (na ordem de alguns milisegundos

e proporcionalmente maior do que o tamanho médio de um pacote), e que este conjunto de limitações inviabilizam a desejável construção de uma rede de comutação de pacotes puramente no domínio óptico (*Optical Packet Switched networks* no original inglês, ou abreviadamente, OPS [11]), os assuntos abordados nesta investigação são os seguintes:

> em primeiro lugar, o estudo da transmissão de grupos muito grandes de pacotes IP numa rede óptica de comutação de agregados de pacotes (em inglês, *Optical Burst Switched networks* ou abreviadamente, OBS), de maneira eficiente, em particular, pela tentativa de formar agregados de pacotes conformes com pacotes IPv6 ou Jumbogramas IPv6 [12], mantendo a retro-compatibilidade destas redes com a versão IPv4;
> em segundo lugar, apresentar propostas tendentes à melhoria da tecnologia OBS por forma a conseguir melhores rácios de transmissão de agregados de pacotes e de optimização da utilização dos recursos de rede;
> em terceiro e último lugar, apresentar uma solução inovadora e integrada que possa ser adoptada e implementada pela indústria.

O objectivo desta investigação foi definir uma nova arquitectura OBS que permita a comutação óptica de agregados de pacotes usando agregados de pacotes IPv4 e IPv6, incluindo a definição de algoritmos de controlo da arquitectura e da montagem de agregados de pacotes IPv4 e IPv6.

Principais contribuições para o avanço do conhecimento científico

Na opinião do autor, as principais contribuições desta tese para o avanço da Ciência são as seguintes:

A primeira contribuição desta tese é a proposta e a avaliação do desempenho de um novo algoritmo de encaminhamento para redes OBS, chamado Extended Dijkstra. O estudo deste algoritmo inclui a definição de um novo conjunto de métricas de encaminhamento usadas para avaliar e comparar a eficiência de algoritmos de encaminhamento em redes OBS, e está apresentado no Capítulo 3. Este novo algoritmo

e a avaliação do seu desempenho foram apresentadas na The Second International Conference on Systems and Networks Communications (ICSNC 2007) [13].

A segunda contribuição desta tese é a proposta e a avaliação do desempenho de um algoritmo não-determinístico e dinâmico de encaminhamento chamado Next Available Neighbour. Este algoritmo é complementar a outros algoritmos existentes e está apresentado no Capítulo 3. Este algoritmo está em processo de patenteamento como patente europeia pela Nokia Siemens Networks AG [14], e uma versão alargada que inclui a avaliação do seu desempenho foi submetida para publicação numa revista[15].

A terceira contribuição é a avaliação do impacto do limite *de facto* de 1500 bytes no perfil de tráfego IPv6, descrito no Capítulo 4. Este estudo foi apresentado na *International Conference on Information Networking 2008 (ICOIN 2008)* [16] tendo recebido o *Best Paper Award*, e uma versão alargada que incluí resultados adicionais foi submetida para publicação numa revista [17].

A quarta contribuição é o novo algoritmo de montagem de agregados de pacotes, sensível ao tráfego, descrito no Capítulo 4. Este algoritmo está incluído na patente internacional WO 2007/118594 A1 da máquina IP-PAC, registada pela Nokia Siemens Networks AG [18]. Este algoritmo foi apresentado na *The 2nd International Conference on Distributed Frameworks for Multimedia Applications 2006 (DFMA 2006)*, que aceitou cerca de 60% das 67 submissões [19]. Uma versão alargada deste artigo foi apresentada na *6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2006)* [20], que aceitou 47 artigos de entre 132 submissões, uma taxa de aceitação de cerca de 36%.

A quinta contribuição é o estudo da relevância das métricas de avaliação de desempenho em redes OBS, *i.e.* a definição da relevância das métricas de Perda de Pacotes *versus* Perda de Agregados de Pacotes *versus* Perda de Bytes, apresentada no Capítulo 4. Este estudo foi apresentado na *5th International IFIP-TC6 Networking Conference (Networking 2006)* [21], que aceitou 20% das cerca de 440 submissões.

A sexta contribuição é o novo conceito de máquina IP-PAC, que está apresentado no Capítulo 5. Este conceito está descrito na patente internacional WO

2007/118594 A1 que descreve a máquina IP-PAC, registada pela Nokia Siemens Networks AG [18]. Uma versão alargada desta patente incluindo a avaliação do desempenho deste conceito, foi submetida para publicação numa revista [22].

A sétima contribuição desta tese consiste na proposta, definição e avaliação do desempenho de uma nova arquitectura para redes OBS, chamada *Common Control Channel Optical Burst Switching (C^3-OBS) Architecture* e descrita no Capítulo 6. Esta contribuição foi registada como patente internacional número WO 2006/072406-A1 [23] pela Nokia Siemens Networks AG. O conceito e a sua avaliação de desempenho foram apresentados na *Fifth International Conference on Networking (ICN 2006)*, que aceitou cerca de 40% dos 390 artigos submetidos [24]. Uma versão alargada deste artigo foi publicada no Elsevier Journal of Optical Switching and Networking [25] e resultados adicionais foram submetidos para publicação noutra revista [26].

A oitava contribuição é a proposta e definição de uma extensão à arquitectura do nó de rede C^3-OBS para uso de domínios em redes C^3-OBS, apresentada no Capítulo 6 e actualmente submetida para publicação numa revista [27].

Enquadramento da Tese

O paradigma de redes de comutação óptica de agregados de pacotes (Optical Burst Switching – OBS) foi inicialmente proposto por Yoo, Jeong e Qiao em 1997 [28-30]. A motivação para a criação deste paradigma é descrita por Xiong, Vandenhoute e Cankaya em [31] seguindo um raciocínio linear: a crescente necessidade de ligações com largura de banda elevada criada pelas Redes da Próxima Geração (*Next Generation Network* no original em inglês, NGN de forma abreviada), tais como a Internet do futuro, encontra solução na implementação nas redes de transporte e de infraestrutura de tecnologias de multiplexagem de canais de dados por divisão densa de comprimentos de onda (*Dense Wavelength Division Multiplexing*, ou abreviadamente DWDM). É expectável que os actuais comutadores electrónicos colapsem com um volume de tráfego tão elevado, apesar de os fabricantes estarem a envidar esforços no sentido de construírem comutadores electrónicos IP de alta velocidade. Actualmente existem comutadores electrónicos IP que gerem tráfego na ordem do Terabit por segundo, como

por exemplo o Cisco 12816 [3] que pode gerir um tráfego agregado de até 1280 Gb/s. No entanto a tecnoloia DWDM pode transportar vários Terabits por segundo e este valor aumenta com a capacidade de multiplexar mais canais de dados numa única fibra. Consequentemente, existe um desnível entre a capacidade de transporte da linha óptica e a capacidade de comutação do equipamento que constitui o nó de rede, resultando isto num engarrafamento de tráfego no nó. A construção de comutadores electrónicos na fasquia do Terabit por segundo é muito cara e não parece ser uma solução viável para as redes nucleares, dada a falta de escalabilidade deste tipo de soluções. Pelo contrário, os comutadores ópticos suportando IP sobre WDM (sendo WDM a abreviatura de *Wavelength Division Multiplexing*) evitam alguma redundância que existe nas tecnologias de transporte, como por exemplo IP sobre ATM sobre SONET/SDH. A proposta de transporte IP sobre WDM é apresentada na secção 2.8.

As redes ópticas com comutação de agregados de pacotes (OBS) são perspectivadas como uma tecnologia promissora para a implementação de redes rápidas e versáteis para transporte de dados usando as infraestruturas de redes ópticas. Este tese apresenta o estado da arte das redes OBS, e foca-se em alguns dos problemas conhecidos deste paradigma, apresentando soluções. A evolução do actual estado da arte das arquitecturas OBS é conseguido pela apresentação de uma nova arquitectura, chamada *Common Control Channel Architecure*, tal como é descrita no Capítulo 6. O seu âmbito abrange a avaliação das actuais propostas para redes OBS, e a proposta de novas soluções. Os resultados destas soluções são demonstrados por simulação, e para este fim foi construído um simulador.

O algoritmo de encaminhamento Extended Dijkstra

Os algoritmos de caminho mais curto, como por exemplo o algoritmo de Dijkstra apresentam um problema de sobrecarga na definição de caminhos sobre as ligações, quando usados para definir caminhos em topologias que incluem anéis em redes que implementem encaminhamento estático definido na origem. A definição do problema fica mais clara através do exame de um caso concreto: quando se aplica o

algoritmo de Dijkstra para definir os caminhos mais curtos entre todos os pares de nós de um grafo com quatro nós em anel, verifica-se que sobre um dos eixos, são definidos três vezes mais caminhos num sentido do que no outro. Este comportamento do algoritmo de Dijkstra, e possivelmente de outros algoritmos de definição de caminho mais curto, iterativos, quando aplicado a redes OBS que implementam definição estática de caminho no nó de origem, resulta numa sobrecarga de tráfego em algumas ligações. Para avaliar o desempenho dos algoritmos de encaminhamento, foram definidas duas métricas: homogeneidade e simetria da distribuição dos caminhos num grafo, sendo que se define o melhor algoritmo como o que, para um dado grafo, define os caminhos de forma mais homogénea e simétrica possível num cenário de probabilidade homogénea de geração e destino de tráfego.

Depois de definido o problema é apresentado o novo algoritmo de encaminhamento denominado *Extended Dijkstra*. Este algoritmo funciona do seguinte modo: quando há apenas um candidato a caminho mais curto, apenas o algoritmo de Dijkstra é aplicado; se há mais do que um candidato a caminho mais curto, é feita a eleição do caminho mais curto usando uma função de desempate que considera os identificadores dos nós de origem e destino.

Este algoritmo foi implementado e testado para redes OBS, sobre um conjunto de topologias simétricas, como por exemplo topologia bidireccionais em anel com oito nós, e para uma topologia homográfica da rede *European Optical Network*. Os resultados obtidos por simulação mostram que o *Extended Dijkstra* produz uma distribuição mais homogénea e simétrica de caminhos sobre as ligações disponíveis. Em consequência disso, o tráfego é melhor distribuido e há menos perda de agregados de pacotes quando este algoritmo é usado na definição de caminhos de uma rede OBS.

O algoritmo de *Extended Dijkstra* pode ser usado em simuladores em substituição do algoritmo de Dijkstra, uma vez que produz caminhos mais homogénea e simetricamente distribuídos, ou ainda pode ser implementado em equipamentos que implementem algoritmos iterativos de definição de caminho mais curto.

O algoritmo de encaminhamento Next Available Neighbour

Sabe-se que nas redes OBS, o rácio de perda de agregados de pacotes é mais elevado quanto maior for o número de nós que o agregado de pacotes tem que atravessar. Este problema é uma consequência da necessidade de ter que encontrar, de forma cumulativa, todos os recursos disponíveis num percurso longo entre o nó de ingresso e o nó de egresso. Numa rede sobrecarregada isto significa que um agregado de pacotes pode não chegar ao seu destino final ao ser bloqueado no nó imediatamente anterior, sendo que entretanto ocupou recursos valiosos na rede, de forma inútil. A perda de agregados de pacotes pode ainda ser causada por avaria de um segmento de rede, em cujo caso os agregados perder-se-ão naquela secção até que o erro seja detectado e definido um novo caminho sobre recursos alternativos. Para além disto, há ainda a considerar a existência de nós oclusos, sendo estes definidos como os nós onde, numa rede, ocorre uma taxa de bloqueio de agregados mais alta do que a média dos outros nós.

Tendo em vista estes cenários, criou-se o algoritmo de encaminhamento *Next Available Neighbour (NAN)*, o qual, numa situação de bloqueio de um agregado de pacotes, promove o seu encaminhamento para um outro nó vizinho usando a ligação disponível nesse instante. Este algoritmo é um algoritmo de encaminhamento complementar e que pode ser usado com outros algoritmos de encaminhamento, uma vez que só é aplicável em situações de eminente bloqueio do agregado de pacotes, e está em processo de submissão para patente europeia pela Nokia Siemens Networks AG.

O algoritmo de encaminhamento *Next Available Neighbour* é um algoritmo dinâmico uma vez que a definição do nó que vai hospedar o agregado de pacotes é feita circunstancialmente, quer seguindo um conjunto de regras bem definidas, como por exemplo, o nó mais próximo do destino final original, ou o nó que está ligado pela ligação menos sobrecarregada, ou outra regra, incluindo por exemplo, uma selecção aleatória do nó vizinho. Neste caso, o NAN será um algoritmo de encaminhamento dinâmico e não determinístico.

Dado que o agregado de pacotes é precedido por um pacote de controlo (CP), o nó vizinho saberá com antecipação que o agregado que vai receber não lhe é destinado e portanto não o submeterá à tarefa de desmontagem. O agregado será armazenado numa memória electrónica para ser reintroduzido na rede OBS assim que seja possível, tendo em vista o nó de destino inicialmente definido. Esta estratégia permite aumentar, por exemplo, a ocorrência de eventos de agrupamento de tráfego uma vez que um agregado hospedado pode ser complementado com os pacotes que, no nó vizinho, se destinem ao nó de egresso original.

Por outro lado, foi estudado o atraso que se introduz pela aplicação do encaminhamento NAN e verifica-se que o atraso de fazer a hospedagem temporária de um agregado num nó vizinho é inferior ao tempo de atraso do agregado se tiver que ser reintroduzido na rede pelo nó de ingresso original.

Foi feita a avaliação do desempenho do algoritmo de encaminhamento NAN quer para redes OBS quer para redes C^3-OBS, tendo-se concluído o seguinte: quando a rede tem escassez de recursos para transportar agregados de pacotes, o encaminhamento NAN é detrimental para o desempenho da rede, já que se está a forçar o nó vizinho com mais tráfego do que aquele que ele já consegue processar e transmitir. No entanto, quando os recursos de rede deixam de ficar sobrecarregados, o encaminhamento NAN apresenta uma melhoria significativa comparativamente a outras estratégias.

O impacto do limite de facto de 1500 bytes no perfil de tráfego IPv6

As características do tráfego IPv4 estão intimamente ligadas ao limite de tamanho de 1500 bytes visível na esmagadora maioria dos pacotes transmitidos. Este facto é muito provavelmente consequência da presença da tecnologia Ethernet nas redes locais, uma vez que o tamanho máximo utilizável da carga de uma trama Ethernet é precisamente de 1500 bytes. Uma vez que não é previsível o desaparecimento das tecnologias Ethernet nas redes locais, onde o tráfego é gerado, assume-se que o tráfego IPv6 estará ainda sujeito a este limite, sendo portanto de esperar que o perfil de tráfego IPv6 seja semelhante aos actuais perfis de tráfego IPv4.

Este estudo teve origem na necessidade de obter dados reais de tráfego IPv6, à data inexistentes. Para colmatar esta lacuna, foi decidido refabricar os dados reais de tráfego IPv4, existentes em grande número e capturados em vários cenários e locais de rede. Decidiu-se aplicar dois algoritmos nesta conversão dos ficheiros de dados reais IPv4: a primeira conversão consistia em fazer a simples substituição do cabeçalho IPv4 pelo cabeçalho IPv6 equivalente. No entanto, verificou-se que em muitos casos esta substituição resultava num pacote cujo tamanho excedia o limite de 1500 bytes do pacote original. Quando tal sucedia, gerava-se um segundo pacote IPv6, pelo que neste cenário de conversão, muitos pacotes IPv4 que tinham tamanhos próximos de 1500 bytes deram origem a dois pacotes IPv6. Verificou-se que neste cenário de conversão de tráfego IPv4 para tráfego IPv6, o aumento do número de bytes transmitidos não era significativo, sendo no máximo inferior a 2.2%. No entanto, e para alguns ficheiros de dados, o aumento do número de pacotes transmitidos chegou a ser de cerca de 47%.

Dado que uma aplicação, no processo de gerar o conteúdo que mais tarde vai ser encapsulado num pacote IP, não tem preocupações com o tamanho dos dados que está a gerar, submeteram-se os dados reais IPv4 a um segundo algoritmo de conversão de ficheiros. Neste, se se verifica a ocorrência de dois ou mais pacotes, com os mesmos pares de endereço e protocolo para a origem e o destino, e se estes pacotes tiverem registos de tempo suficientemente próximos, então é assumido que estes pacotes podem ter resultado da distribuição do conteúdo inicialmente gerado por uma aplicação por dois pacotes IPv4. Foram investigados os limiares de tempo aplicáveis ao conceito de "suficientemente próximo", e usados valores entre os zero e um segundos. Verificou-se que para zero segundos o aumento no número de pacotes era consistente com a aplicação do algoritmo de substituição de cabeçalho anteriormente descrito, e que para tempos maiores, se verificava uma diminuição do número de pacotes gerados. O valor de limiar para o qual o algoritmo de conversão de ficheiros apresenta um valor de número de pacotes IPv4 igual ao número de pacotes IPv6 gerados é de cerca de 447 s. Para valores de tempo superiores, verifica-se já um fenómeno de multiplexagem estatística, resultado da agregação de vários pacotes num único pacote.

Aplicou-se seguidamente o algoritmo de conversão usando um limiar de 9 KB para o tamanho de pacote, correspondente a uma trama Ethernet JumboFrame, e um

limiar de tamanho de 64 KB, correspondente ao tamanho máximo de um pacote IP. Verificou-se que, para o limiar de tamanho de 9 KB, quando observado o tempo de 447 s, existe uma diminuição de cerca de até 50% no número de pacotes gerados. Para o limiar de 64 KB os valores na diminuição do número de pacotes gerados são semelhantes aos obtidos para 9 KB. Desta maneira concluíu-se que a utilização de Ethernet com JumboFrames poderia resultar num decréscimo de até 50% do número de pacotes transmitidos nas redes estudadas.

O novo algoritmo de montagem de agregados de pacotes sensível ao tráfego

No estudo de tráfego real IPv4 e tráfego IPv6 convertido, foram aplicados os principais algoritmos de montagem de agregados de pacotes e feita a sua avaliação de desempenho. Foram testados os algoritmos de montagem de agregados de pacotes com vários tipos de limites: limitados pelo tamanho do agregado, limitados pelo tempo de montagem e simultaneamente limitados pelo tamanho do agregado e pelo tempo de montagem. Foi ainda feita a avaliação da eficiência destes algoritmos, tendo em consideração que um algoritmo eficiente é aquele que consegue agregar o maior número de pacotes minimizando o atraso médio desses pacotes, causado pela tarefa de agregação.

Foi observado que a eficiência dos algoritmos de montagem de agregados de pacotes variava com os ficheiros de tráfego IPv4 usados, isto é, a eficiência dos algoritmos de montagem de agregados de pacotes é função não apenas dos limiares definidos mas também das características do tráfego tributário do algoritmo, uma vez que ligações com maior intensidade de tráfego podem gerar agregados maiores. Sabendo que as condições do tráfego variam não só com a localização geográfica mas também no tempo, propõe-se um novo algoritmo de agregação cujos limiares são inicialmente definidos, mas alteráveis durante a operação do algoritmo, por forma a possibilitarem a adaptabilidade do algoritmo às condições do tráfego tributário.

O algoritmo adaptativo funciona como segue: se o limiar que está correntemente a definir a conclusão da montagem do agregado é o limiar de tamanho e este valor é

ainda menor do que a unidade máxima de transmissão (em inglês *Maximum Transmission Unit* ou MTU), então este limiar será aumentado num dado rácio; se o limiar activo na conclusão da montagem do agregado é o limiar de tempo e o tamanho do agregado está dentro do intervalo de tamanho aceitável (sendo o intervalo definível por um rácio do tamanho máximo), então este limiar será diminuído num dado rácio; finalmente, se o limiar que está a definir a conclusão da montagem do agregado de pacotes for o limiar de tempo e o agregado de pacotes é menor do que o intervalo de tamanho aceitável (sendo o intervalo definível por um rácio do tamanho máximo), então este limiar será aumentado num dado rácio, não excedendo o valor inicialmente definido como máximo para o limiar de tempo. Este algoritmo adaptativo faz parte da máquina IP-PAC apresentada no Capítulo 5.

A definição da relevância das métricas de avaliação de desempenho de redes OBS

A métrica usada para a avaliação de desempenho para redes OBS tem sido o rácio de perda de agregados de pacotes, calculado como o quociente entre o número de agregados de pacotes bloqueados nos nós da rede e os agregados de pacotes gerados nos nós de ingresso para um dado intervalo de tempo. Recentemente, alguns autores sugeriram que este rácio não era significativo e que o rácio de perda de pacotes dentro dos agregados era mais diferente.

Este aspecto da investigação propôs-se estudar o significado das três métricas possíveis para avaliação do desempenho de redes OBS: rácio de perda de agregados de pacotes, rácio de perda de pacotes dentro dos agregados e ainda rácio de perda de bytes, calculados, respectivamente, como sendo os quocientes entre a quantidade de agregados de pacotes perdidos e a quantidade de agregados de pacotes gerados, o número de pacotes existentes nos agregados de pacotes perdidos sobre o número de pacotes apresentados à rede para agregação, e o tamanho em bytes dos agregados perdidos e o volume de tráfego em bytes apresentado à rede para agregação.

Para tal usou-se um conjunto de ficheiros contendo tráfego real IPv4, os quais serviram para alimentar as filas de agregação existentes nos nós de ingresso de uma rede

OBS. O algoritmo de agregação usado foi o algoritmo híbrido, que considera conjuntamente os limiares de tempo de agregação e de tamanho do agregado de pacotes gerados. A rede simulada foi uma rede em anel bidireccional com quatro nós, definida para operar a vários níveis de carga.

Foram também simulados protocolos sensíveis ao tamanho do agregado e que implementam preenchimento de vazios, como por exemplo o protocolo JET, e o protocolo JIT. O algoritmo de agregação foi definido por forma a garantir a eficiência do processo de montagem dos agregados, isto é, minimizando o tempo de atraso causado aos pacotes e maximizando o número de pacotes contidos no agregado. Os agregados gerados pelos cenários de simulação foram muito heterogéneos, quer em termos de máximo e mínimo de número de pacotes quer em termos do tamanho em bytes dos agregados.

Os resultados obtidos por simulação mostram que em todos os cenários de carga, quer para o protocolo JIT quer para o protocolo JET, os rácios de perda de agregados, de perda de pacotes e de perda de bytes não são iguais, mas têm valores muito próximos, o que está de acordo com a Lei dos Grandes Números, confirmando que o rácio de perda de agregados de pacotes é uma métrica significativa para a avaliação de desempenho de redes OBS.

O conceito de máquina IP Packet Aggregator and Converter

A agregação de pacotes de dados foi inicialmente proposta no princípio da década de 80 para a tecnologia ATM. No entanto, o princípio de agregação de tráfego é válido para outras tecnologias. Nesta perspectiva, foi apresentado o conceito de máquina de agregação *IP Packet Aggregator and Converter*, publicado como patente internacional pela Nokia Siemens Networks AG.

Actualmente está a assistir-se à transição do protocolo Internet da versão 4 para a versão 6, como resultado de um conjunto de necessidades e medidas políticas. Isto resulta numa situação em que algumas redes operam apenas com IPv4, outras operam apenas com IPv6 e outras ainda operam simultaneamente com IPv4 e IPv6.

Naturalmente a comunicação de redes IPv4 com redes IPv6 implica à utilização de um dispositivo de transformação do formato nativo no formato de destino numa maneira tal que os pacotes possam ser transmitidos e interpretados correctamente.

O conceito de máquina IP-PAC destina-se a aumentar a eficiência da transmissão de pacotes IPv4 e IPv6 em redes nucleares, encapsulados num único pacote IPv6 ou numa versão futura do protocolo Internet (dito *IPvFuture*).

O IP-PAC realiza a agregação de pacotes IPv4 e IPv6 de acordo com um algoritmo adaptativo de montagem de agregados de pacotes e a re-encapsulação do agregado de pacotes num único pacote IPv6/Future.

Como o IP-PAC é uma máquina de agregação, pode ser usada em redes onde a transmissão de pacotes de dados beneficie da multiplexagem estatística resultante do processo de agregação. Neste sentido o IP-PAC pode ser utilizado numa variedade de cenários de rede, realizando transmissões de agregados de pacotes quer para outras máquinas IP-PAC quer para máquinas que não possuem capacidade de desmontagem de agregados de pacotes. Neste último caso, o IP-PAC envia de forma sequencial os pacotes entretanto armazenados na fila de espera, sendo estes comutados individualmente na rede, possivelmente aproveitando ainda assim os benefícios do efeito de acomodação de caminho.

O IP-PAC pode ser utilizado como nó de fronteira de uma rede OBS, sendo que neste caso são da sua responsabilidade as tarefas de agregação e desagregação de pacotes, sendo as tarefas de comutação e encaminhamento dos agregados realizadas pelos nós nucleares da rede OBS.

A arquitectura de redes OBS Common Control Channel Optical Burst Switching

As redes OBS são características pela forma como as decisões de definição tráfego são tomadas nos nós de ingresso e de fronteira, como por exemplo as decisões relativas às tarefas de agregação e definição de caminho. Esta abordagem é penalizadora

da qualidade das decisões tomadas, uma vez que as experiências e a informação disponíveis nos nós nucleares ou nos outros nós de fronteira não contribuem para a tomada de decisões bem informadas por um dado nó de ingresso.

Para resolver este problema, a arquitectura de redes OBS denominada Canal de Controlo Comum (ou *Common Control Channel* em inglês, dita C^3-OBS) propõe um esquema de controlo que é distribuído por toda a rede, ao invés do esquema de controlo individual implementado pelos nós OBS. Esta arquitectura está submetida como patente internacional pela Nokia Siemens Networks AG.

O controlo distribuído é conseguido pela aplicação de um canal de controlo que promove a difusão dos pacotes de controlo por todos os nós da rede, de forma imediata e completamente óptica. Desta maneira, quando um dado nó emite um pacote de controlo, todos os nós da rede recebem pelo menos uma cópia desse pacote.

O canal de controlo comum é implementado pela junção à entrada e divisão à saída de todos os canais de controlo provenientes de cada uma das fibras ópticas. Para evitar a reinserção ilimitada em ciclo de um dado pacote de controlo, cada nó possui um conjunto de interruptores ópticos que bloqueiam a propagação do sinal, por forma a que o canal de controlo comum seja topologicamente configurado como uma árvore.

Ao receber todos os pacotes de controlo apenas com o atraso inerente à propagação dos dados na fibra óptica, cada nó consegue manter uma base de dados, denominada *Local Network Model* (LNM). O LNM é uma extensão das bases de dados propostas pelos protocolos de sinalização e reserva de recursos, feita por forma a reflectir o estado dos pedidos de reserva de recursos não só do próprio nó, como também de todos os outros nós da rede. O LNM é operado por um conjunto de algoritmos denominado *Travel Agency Routing Algorithm* (TAA) responsável pela manutenção das bases de dados.

Quando o primeiro pacote de dados entra na fila de montagem do agregado, o nó de ingresso pede ao nó nuclear a que está ligado para autorizar a entrada do agregado, fornecendo-lhe a informação de que dispõe nesse instante: o endereço de destino, o possível tamanho do agregado e o seu provável instante de partida. O nó C^3-OBS

nuclear que recebe este pedido vai, por consulta ao seu LNM, definir quando e como o agregado vai ser encaminhado na rede. Por *quando* quer dizer-se que o agregado pode ter que ficar algum tempo mais na fila de agregação, até os recursos estarem todos disponíveis. Chama-se horizonte de partida (ou *Departure Horizon* em inglês) ao intervalo de tempo admissível para a entrada do agregado na rede, e na tese foram apresentados resultados obtidos por simulação para diferentes valores deste horizonte de partida. Por *como* quer dizer-se que, aquando desta consulta, o TAA examina a disponibilidade de recursos para os *k* caminhos mais curtos entre o nó de origem e o nó de destino, tendo em conta a informação que está reflectida na base de dados que é o LNM, fornecida pelos vários pacotes de controlo que foram sendo disseminados de forma automática pela rede.

A arquitectura C^3-OBS foi testada para um conjunto de topologias, regulares e reais, como por exemplo, redes em anel e topologia homográficas das redes EON e NSFnet, estas últimas quer com ligações de comprimento real, quer com ligações escaladas ou de comprimento fixo. Verifica-se que as redes C^3-OBS desempenham várias ordens de grandeza melhor do que os seus equivalentes OBS, em particular quando os recursos de rede não estão saturados, em cujo caso o ganho em desempenho desta arquitectura é marginal.

A arquitectura de um nó C^3-OBS para implementação de domínios de rede

Um dos principais problemas trazidos pela disseminação automática não controlada da arquitectura C^3-OBS é a redundância de informação dos pacotes de controlo, uma vez que um nó pode receber várias cópias do mesmo pacote de controlo, cada uma trazida por um ramo diferente da árvore que conecta esse nó. Outro dos problemas decorre da utilização da arquitectura C^3-OBS em redes com um elevado número de nós ou redes com ligações muito longas, cenários que podem implicar tempos de propagação da informação dos pacotes de controlo demasiado longos, com o consequente resultado de as decisões de ingresso estarem a ser tomadas sobre uma base de dados que não está actualizada.

Como forma de limitar o âmbito de disseminação automática dos pacotes de controlo, e ainda como forma de limitar zonas da rede C^3-OBS a apenas alguns nós, foi proposto o conceito de domínio de rede C^3-OBS.

Um conjunto de nós C^3-OBS formam um domínio C^3-OBS quando cada um desses nós mantém um modelo local de rede (LNM) que reflecte apenas o estado da rede para esse conjunto de nós. Consequentemente, o tamanho do LNM e o tempo de disseminação da informação dos pacotes de controlo é reduzido comparativamente ao tamanho e tempo necessários para a manutenção de um LNM relativo à totalidade da rede.

Um conjunto de domínios C^3-OBS comunicam entre si por meio de nós de fronteira. Estes nós têm a particularidade de servir de interface entre dois ou mais domínios, isto é, mantêm um LNM que reflecte o estado das reservas para todos os nós de todos os domínios com que fazem interface. Estes nós são ainda responsáveis pelo encaminhamento dos agregados entre domínios, uma vez que podem decidir sobre o caminho que o agregado vai tomar quando entrar no novo domínio.

O diagrama esquemático da arquitectura de um nó de fronteira C^3-OBS é diferente do diagrama de um nó C^3-OBS, sendo ambos apresentados nesta tese. É ainda apresentado um estudo dos tempos de separação quando o agregado tem que viajar através de vários domínios.

Conclusões e propostas de trabalho futuro

As redes OBS são uma tecnologia viável (em oposição ao OPS), interessante quer do ponto de vista de investimento em capital quer do ponto de vista de investimento operacional (em oposição às redes SONET e SDH). As redes OBS prometem ser rápidas e versáteis. No entanto, os desenvolvimentos tecnológicos na indústria das telecomunicações estão sujeitos a um grande número de restrições e tensões, quer por parte dos operadores, quer ainda por parte dos utilizadores, dos regulamentadores e finalmente como consequência da concorrência que dinamiza este

sector. Como primeira conclusão, é preciso enfatizar que não é possível prever o papel que as tecnologias OBS terão nas redes ópticas do futuro.

Esta tese focou-se na apresentação de soluções para alguns problemas das redes OBS, soluções essas organizadas de uma forma coerente numa abordagem que cobre desde o ínicio até ao fim de uma comunicação em OBS.

Há outros assuntos que reclamam atenção nas redes OBS, como por exemplo, o uso de protocolos de sinalização e reserva de recursos. Foi mostrado que o JIT^+ e o E-JIT têm melhor desempenho do que outros protocolos mais complexos como resultado das suas bases de dados simplificadas. Mas tal como o JIT^+ é uma extensão ao conceito do JIT pela possibilidade de admitir uma segunda reserva, qual seria o resultado se se seguir esta linha de raciocínio? Isto é, quais seriam as implicações de um algoritmo do tipo E-JIT se fossem permitidas 3, 4, 5 ou 10 reservas? Onde está o limite de simplicidade para estas bases de dados?

Mais, foi mostrado que o E-JIT é mais eficiente quando a rede opera com agregados mais pequenos, porque neste caso os tempos associados à operação do canal permitem uma maior eficiência. Qual seria o desempenho do E-JIT para tráfego real usando algoritmos eficientes para a montagem de agregados de pacotes? E mais ainda, existe possibilidade de melhorar os outros protocolos seguindo a mesma linha de raciocínio do E-JIT?

Outra questão surge quando se aborda o desempenho de protocolos de sinalização e de reserva de recursos. É assumido que o JET e outros protocolos que admitem reservas por preenchimento de vazios têm tempos de configuração maiores resultantes da sua maior complexidade. Mas exactamente quantas reservas simultâneas um protocolo destes terá que gerir? Ou dito de outra forma, é importante avaliar o tamanho de uma base de dados operacional para estes tipos de protocolos como forma de aferir o limite de complexidade da manutenção das suas bases de dados.

Apesar de vários autores terem já abordado este assunto, principalmente por via do estudo dos algoritmos de montagem de agregados de pacotes, seria interessante estudar o comportamento das redes OBS com agregados de pacotes realmente grandes.

Ou dito de outra maneira, existem benefícios em definir uma unidade máxima de transmissão para OBS?

Outro assunto que carece de mais investigação é o estudo de redes híbridas de nós OBS e nós C^3-OBS. Também promissor é o uso de caminhos e circuitos hamiltonianos para a definição da topologia do canal de controlo comum como forma de limitar a replicação excessiva dos pacotes de controlo nas redes C^3-OBS, e a avaliação destas redes com canais de controlo hamiltonianos de um ponto de vista de rácio de bloqueio de agregados e de atraso médio dos pacotes nos agregados.

De um outro ponto de vista, o número dos *k* caminhos seleccionáveis para o TAA poderá estar relacionado com o grau nodal médio da rede, ou pelo menos, com o grau nodal médio do nó em questão.

A utilização conjunta do encaminhamento NAN com diferentes limiares para o horizonte de partida tem também que ser investigada nas redes C^3-OBS. Em redes OBS seria interessante avaliar o uso cumulativo de *Fixed Alternate Routing* ou de *Deflection Routing* com encaminhamento NAN.

Finalmente, podemos aplicar um novo mecanismo de qualidade de serviço atribuíndo a cada classe de agregados de pacotes um intervalo de tempo diferente para o horizonte de partida, fazendo desta área um novo campo de investigação.

Chapter 1.

Introduction

1.1. Thesis focus and scope

Optical Burst Switched (OBS) networks are envisaged as a promising solution, capable of fast and versatile transport of data packets using optical network infrastructures. These infrastructures consist of a series of nodes that are interconnected with optical fibres in which packets are transported being these nodes responsible for the switching of the packets.

Optical networks are often classified according to its generational features. The first generation of optical networks was deployed to solve the bottleneck problem at the link level due to high speed transmission ratios, acting the fibres as a very high speed, low loss data transport medium. In these networks, nodes interpret and switch packets in the electronic plane, much like in electronic networks. First generation optical networks constitute the backbone of large area networks such as the GEANT2 network [1]. Optical networks can also be found at smaller network scales, connecting network elements in metropolitan areas or where the use of copper wires is unadvisable, *e.g.*, in electronically sensible or very demanding industrial environments, usually transporting Ethernet at 10, 100, 1000 or 10000 Mb/s.

The second generation of optical networks aimed to solve the traffic bottleneck problem at the electronic routers, a consequence of the increase of the transport capacity in the optical links. The second generation consists of almost static virtual circuits set up on multiplexed channels in a fibre and has been able to cope with more or less success with current traffic demands.

The continuous demand posed by users who want to access data everywhere, every time, from all sorts of communication devices, powered by the enormous success of Internet, poses a traffic transport problem to network owners and operators. In fact, fibre capacity, actually in the Terabyte range and growing [2], mismatches the traffic handling capacity of such nodes, actually in the Gigabyte range (*e.g.* [3]) and not growing as fast as traffic demand. The increase in the data transport capacity for a single fibre, estimated to be of around 50% per year [4], is only surpassed by the network demand to transmit data, estimated to grow between 75% and 100% each six months [5-8], thus allowing for the prediction of new overloads on these networks in a not too distant future.

Pursuing this increasing need to transport data, a third generation of optical networks, also known as Next Generation Optical Internet, will further address the bottleneck issues on the nodes, allowing infrastructure owners to deploy new services with increased available bandwidth, *e.g.* triple play services. This new generation will implement the Internet Protocol over Wavelength Division Multiplexing (IP over WDM) as a scheme to decrease both operational and capital expenditures (OPEX and CAPEX, respectively) for network owners and operators.

OBS networks fall in this last category. Using WDM technology, an OBS network uses *n-1* channels of the *n* available channels, to transport data in a transparent manner in an optical network, while it uses one reserved channel as control channel to transport the control packets that contain routing and switching information in a non-transparent manner, *i.e.*, the packets that traverse the network in this reserved channel are electronically interpreted in each node, to allow for node configuration and resource reservation.

This thesis presents the state of the art related to the OBS paradigm and focus on some known OBS problems, proposing solutions. The OBS problems that are addressed are related to burst loss and from a network and logical layers perspective, pose some new challenges. Routing algorithms are constrained due to the impossibility of performing hop-by-hop routing in OBS, since the intermediate nodes must route the data bursts in a transparent manner. Quality of service and burst priority, scalability of

the architectures and the node management algorithms, burst assembly algorithms and the traffic grooming are some of the topics that require solutions in OBS networks, in most cases needing a new approach as the traffic management is assumed to be performed at the edge of the network and not on a node basis, as in regular electronic networks. Some of these problems are addressed in this thesis while others are not as they fall out the scope of this research programme.

Along this thesis, a burst is said to be lost when there are not enough network resources available to effectively route the burst from the ingress node to the egress node in the network. There are several approaches to minimize burst loss rate in OBS networks. We address this issue from a multiple perspective: we propose a new OBS network architecture, two new routing algorithms, a new management algorithm for the core nodes and a new burst assembly algorithm.

The scope of the research work reported in this thesis encompasses the assessment of the performance of OBS networks as defined by the current state of the art and the proposal of novel solutions. The performance assessment of those solutions is achieved by simulation and for this purpose a previously built simulator was adapted. At the end of this chapter, a detailed list of the main contributions for the advance of the state of the art in the area of OBS networks, from the point of view of the author, is presented.

1.2. Problem statement and goals of the research

The statement of the problem that defined the research programme is based on the following assumptions:

1. IP networks are widespread.
2. The transition from IP version 4 (IPv4 [9]) to IP version 6 (IPv6 [10]) is currently starting to take place, and IPv6 is expected to be the dominant version of the IP protocol in a near future.
3. Flexible high capacity optical networks will constitute the main technology for the core transmission network infrastructure, in response to the need to

support the dynamic next generation Internet scenarios and applications such as triple play services, voice over IP, IP television, peer to peer applications and so on.

4. The full optical packet switching (OPS) [11] technology is limited by the inexistence of viable optical memory with random access and adequate size and granularity, by the inexistence of adequate optical logic and by the time needed to configure a current optical switching element and finally, by the problems related to the synchronization of data in several channels at the input of the switching matrix.

Having this in view, this research programme addresses the following issues:

- Firstly, to study the transmission in an effective way of very large groups of IP packets in an OBS network, namely by trying to form data bursts compliant to IPv4 datagrams and IPv6 packets including IPv6 Jumbograms [12], while maintaining the backward compatibility with IPv4 networks;
- Secondly, to improve the OBS network technology in order to optimise the network resource utilization and to achieve better burst transmission ratios;
- Third and finally, to present a novel and integrated solution that can be adopted and deployed by the industry.

In conclusion, the goal of the research activities was to define a mean to perform optical burst switching using IPv4 and IPv6 packet bursts, including the definition of a framework combining the assembly of bursts using tributary IPv6 and or IPv4 packets with a new OBS architecture.

1.3. Organization of the thesis

This thesis is organized in seven chapters. The present chapter, devoted to the introduction, describes the thesis focus and scope, the problem statement and the goals of the research activities, the organization of the thesis and the main contributions of the thesis to the advance of scientific knowledge.

Chapter 2 presents the state of the art of Optical Burst Switching networks. It begins with the presentation of the optical burst switching paradigm followed by the discussion of main OBS architectures, including the basic architecture of OBS networks, the architecture of Dynamic Wavelength Routed OBS, and other architectures. The discussion of the generic burst switching concept and of the main burst assembly algorithms are followed by the presentation of one way resource reservation protocols, also discussing the void filling or no void filling and the immediate or delayed resource reservation strategies. The main approaches to contention resolution are discussed, namely, prioritization, optical buffering, burst segmentation, burst and traffic grooming, routing strategies and algorithms and wavelength assignment algorithms. The proposals for IP over WDM, IP over OBS and TCP over OBS are presented and a summary ends this chapter.

Chapter 3 presents main routing algorithms and strategies for OBS. It includes sections on static and dynamic routing, and on dynamic routing strategies for OBS, where deflection routing is discussed. Two new routing algorithms for Optical Burst Switching networks, the static shortest path Extended Dijkstra routing algorithm and the Next Available Neighbour dynamic routing algorithm are presented. This chapter also presents two new performance assessment metrics, called route balance and route symmetry, and the performance of the Dijkstra and of the Extended Dijkstra are evaluated using these metrics. A summary ends this chapter.

Chapter 4 presents conclusions on burst traffic based on IPv4 and IPv6, including the main burst assembly algorithms, namely, Maximum Burst Size, Maximum Time Delay and Hybrid burst assembly. A method to convert IPv4 traffic to IPv6 is discussed. The results of the performance assessment for several burst assembly algorithms using real traffic are presented and conclusions on the efficiency of the burst assembly algorithms are drawn. The chapter also presents and discusses relevant conclusions on OBS networks performance metrics such as burst loss, packet loss or byte loss ratios. A summary concludes this chapter.

Chapter 5 presents the burst assembly machine concept named IP Packet Aggregator and Converter (IP-PAC) and its proposed application environment in a

generic IP network architecture. It also presents a new burst assembly algorithm, dynamically adaptable to incoming traffic characteristics. The connection of IP-PAC machines to a core network is described, including the communication method between IP-PAC nodes and IP-PAC and non-IP-PAC nodes. The performance assessment of the IP-PAC machine concept is presented. This chapter also presents the discussion and main conclusions on the IP-PAC concept. A summary ends this chapter.

Chapter 6 presents the details of the new OBS architecture named C^3-OBS, including the use of IP-PAC machines as burst assembly nodes. The details of the Common Control Channel, the Local Network Model, the Travel Agency algorithm and the definition of time for a well informed network are presented. This chapter also proposes the use of network domains as a mean to enlarge scalability of C^3-OBS networks and presents the simulator built to research and assess the performance of the new C^3-OBS architecture. The performance of the C^3-OBS and OBS architectures are assessed and compared for several regular and irregular topologies. This chapter ends presenting the main conclusions.

Chapter 7 presents an overview of the main findings and contributions of the thesis and suggests new research directions for future work.

Annex A presents details about the implementation of the simulator built to model the OBS and C^3-OBS networks, its simulation parameters, a topology sample configuration file and a sample output file.

The reference section presents the detailed information for the references cited in this thesis. These are numbered sequentially by order of appearance in the English text of this thesis. This section concludes this thesis.

1.4. Main contributions for the advance of the scientific knowledge

This section presents, in the opinion of the author, the major contributions for the advance of the scientific knowledge resulting from the research work reported in this thesis.

The first contribution of this thesis is the proposal and performance evaluation of a new routing algorithm for optical burst switching networks, named Extended Dijkstra Routing Algorithm. The study of this algorithm includes the definition of a new set of routing metrics used to assess and compare the efficiency of routing algorithms in optical burst switching networks and is presented in Chapter 3. This new algorithm and its performance assessment were presented at The Second International Conference on Systems and Networks Communications (ICSNC 2007) [13], which had an acceptance rate of 34%.

The second contribution of this thesis is the proposal and performance evaluation of a non-deterministic dynamic routing algorithm named Next Available Neighbour (NAN) Routing Algorithm. This routing algorithm is complementary to existing routing algorithms and is presented in Chapter 3. This algorithm is in filing process as an European patent by Nokia Siemens Networks AG [14] and an extended version including its performance assessment has been submitted for publication in a journal [15].

The third contribution is the assessment of the impact of the *de facto* 1500-byte packet size limit on the IPv6 traffic profile, which is described in Chapter 4. This study was presented for presentation at The International Conference on Information Networking 2008 (ICOIN 2008) [16] having received the Best Paper Award. An extended version of this paper was submitted for publication in a journal [17].

The fourth contribution is a new Traffic-Aware Burst Assembly Algorithm, which is described in Chapter 4. This algorithm is included in the international patent WO 2007/118594 A1 on the IP-PAC machine, registered by Nokia Siemens Networks [18]. This algorithm was presented at The 2nd International Conference on Distributed Frameworks for Multimedia Applications 2006 (DFMA 2006), which accepted about 60% of 67 submissions [19]. An extended version of this paper was presented at The 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2006) [20], which accepted 47 papers out of over 132 submissions, *i.e.*, an acceptance ratio of about 36%.

The fifth contribution is the study of the relevance of performance assessment metrics in optical burst switching network, *i.e.*, the definition of the relevance of Packet Loss versus Burst Loss versus Byte Loss in OBS networks, which is presented in Chapter 4. This study was presented at The 5th International IFIP-TC6 Networking Conference (Networking 2006) [21], which accepted 20% of about 440 submissions.

The sixth contribution is the new IP-PAC machine concept, which is presented in Chapter 5. This machine concept is described in the international patent WO 2007/118594 A1 on the IP-PAC machine, registered by Nokia Siemens Networks [18]. An extended version of this patent including the performance assessment of IP-PAC machine was submitted for publication in a journal [22].

The seventh contribution of this thesis is the proposal, definition and performance assessment of a new OBS architecture, named Common Control Channel Optical Burst Switching (C^3-OBS) Architecture, which is described in Chapter 6. This contribution was first registered as the international patent WO 2006/072406 A1 [23], by Nokia Siemens Networks. The concept and performance assessment of the C^3-OBS architecture were presented in The Fifth International Conference on Networking (ICN 2006), which accepted 40% of 390 papers [24]. An extended version of this paper was published in the Elsevier Journal of Optical Switching and Networking [25] and further results were also submitted for publication in another journal [26].

The eighth contribution is the proposal and definition of an extension of the C^3-OBS node architecture to the use of C^3-OBS Network Domains, which is presented in Chapter 6 and has been submitted for publication in a journal [27].

Chapter 2.

Optical Burst Switching Networks

2.1. Introduction

The Optical Burst Switching network paradigm (OBS) was initially proposed by Yoo, Jeong and Qiao in 1997 [28-30]. The motivation for OBS is described by Xiong, Vandenhoute and Cankaya in [31], following a very straightforward line of reasoning: the increasing need of high volume bandwidth by the Next Generation Networks (NGN) such as future Internet is met with the deployment of Dense Wavelength Division Multiplexing (DWDM) technologies in existing transport and backbone networks. Electronic routers are expected to collapse to such a traffic volume and there have been great efforts in building hardware-based high-speed electronic IP routers. Actually, there are IP routers in the Terabit per second class, *e.g.* the Cisco 12816 [3] can manage an aggregated traffic up 1280 Gb/s, but the DWDM technology can deliver several Terabits per second and the count is increasing with the ability to pack more data channels into one fibre, so there is a mismatch between line capacity and router capacity, hence a bottleneck is to be expected. The construction of Terabit/s electronic IP routers is subject to high costs and is not regarded as a viable solution to core networks. On the contrary, optical routers supporting IP over WDM through OBS will have better scalability and will be simpler since IP over WDM avoids some redundancy found in other transport technologies, such as IP over ATM and SONET/SDH.

The remainder of this chapter is organized as follows: section 2.2 introduces the generic burst switching concept. Section 2.3 presents the optical burst switching network concept, section 2.4 describes the main OBS architectures and the main burst assembly algorithms are discussed in section 2.5. Section 2.6 is devoted to the most significant resource reservation protocols and section 2.7 discusses the most relevant resource contention resolution approaches. IP over WDM, IP over OBS and TCP over

OBS architectures are presented in sections 2.8, 2.9 and 2.10 respectively. Section 2.11 with the summary concludes this chapter.

2.2. Burst switching

Data packet aggregation process was independently proposed by Haselton [32] (September 1983) and Amstutz [33] (November 1983) as a way to benefit from the statistical multiplexing effect, or as initially described, "improving circuit-switching channel efficiencies" [32]. Haselton presents burst switching in a very enlightening manner: "(…) Burst switching utilizes elements of both packet- and circuit-switched techniques. Both burst switching and packet switching use header techniques. However, packet disciplines use fixed message lengths while burst disciplines use unlimited message lengths. Both circuit and burst latch channels. Circuit switching however, inefficiently latches channels for the duration of the call. Burst switching latches only during the voice or data burst interval, latching and unlatching many times within the calling interval, thus reducing packet-switching (header) overheads and dramatically improving circuit-switching channel efficiencies. (…)" (*in* [32]). For Amstutz, the burst included a 4 byte header, as bursts were meant to travel in time division multiplexing (TDM) scheme and differed from packet switching because bursts were sent interleaved with other bursts in TDM, in opposition to packet transmission done using full link bandwidth. This single difference accounted for the performance improvements, namely by increased header efficiency (the reduction of total header size for packets when compared to header size of bursts) and also by improved bandwidth efficiency, namely for voice traffic [33].

The burst concept was later re-introduced in Optical Networks, contributing to the Optical Burst Switching Network paradigm, in the sense that the header of the aggregated packets is now a control message travelling in a dedicated channel in the network. Thus, in OBS the burst does not require a header since it is not meant to be optically processed for switching or routing. Burst assembly is presented in section 2.5 and in Chapter 4. From a more general perspective, bursts are formed when a number of data packets, regardless of its format, are assembled into a single aggregate, with or without final encapsulation of the constituent data packets. Also, this aggregation is not

format specific since from a conceptual point of view, packets are treated as a formatted group of bytes. The only true requirement to burst assembling is that its disassembling is viable, *i.e.*, the retrieval of the original groups of bytes (packets) from the burst must be done without data loss in a manner that allows further data interpretation. In burst switched networks, bursts are the smallest switchable unit.

The IP-PAC machine described in Chapter 5, presents a proposal for burst assembly and foresees encapsulation of the burst in an IPv6 or IPvFuture[1] packet when the destination node is another IP-PAC machine. The advantages of encapsulation are a consequence of the possibility to handle burst inside the network as if it was a single packet and thus to make it suitable for further switching and addressing techniques, such as, for example, OBS with Next Available Neighbour routing, as described in section 3.4.2, or inter-domain forwarding, as used in C^3-OBS domains. The disadvantage is that the length of the burst is increased by the size of the IP envelope packet header (for IPv6 this would be at least 40 bytes).

Issues related to burst switching have been addressed by several authors previously, such as burst size limitations [34, 35], self-similarity changes caused by the aggregation [36, 37], the relation between burst size and burst loss ratios [21, 36], the format of the burst traffic inside an OBS network [38], the efficiency of burst assembly algorithms [19, 20, 39] and *et cetera*. The main motivation for all the research in burst assembly and burst traffic seems to be that the better the behaviour of the traffic in a network is understood, the more efficiently the issues in management and planning for the network can be addressed and thus more efficient becomes the network.

2.3. The optical burst switching network concept

A computer network is "an interconnected collection of autonomous computers. Two computers are said to be interconnected if they are able to exchange information" [40]. These may be classified accordingly to its geographical extent, to the nature of its

[1] IPvFuture is used as a generic term to represent any future version of the Internet Protocol (IP).

transmission medium or following a number of other criteria. On the geographical coverage, networks are usually termed as Wide-Area Networks (WAN), Metropolitan Area Networks (MAN), Local Area Networks (LAN), Personal Area Network (PAN) and Body Area Network (BAN) as their area of operation decreases. As to the network hierarchy, networks are classified as core networks, metropolitan and access networks, being that core networks connect a group of metropolitan networks and these last serve a set of access networks which ultimately connect users. The span of core networks is very large, *i.e.* core networks are usually continental or inter-continental networks. As to the medium of transmission, a network that uses an optical fibre to transmit data signals is an optical network.

The data packets or bursts travel in the fibre in an optical form. Arriving at an intermediate node, they are either converted to electronic form to be interpreted, switched and forwarded, or they are kept in its optical form and forwarded to the next node. The ideal transport of data by switching and routing packets in a purely optical manner is not viable as there is no optical logic available yet, and it is not foreseeable when such technology will be available. This optical logic includes the lack of true random access optical memory and the logic devices responsible for optical synchronization of the packets or bursts. The conversion to electronic form is not desirable as the operation of high speed packets in electronic form is expensive and slow when compared to its optical transmission.

Optical Burst Switching networks use optical fibres as a transport medium, and are viewed as the next best thing near to Optical Packet Switching (OPS), this later being the switching and routing of data packets in a purely optical manner. OPS is the paramount of optical networking, the grail every company in the industry tries to conquer. But OPS at multi-Gigabit speed ratios requires optical logic, something that is not yet fully mature and / or available.

The OBS paradigm circumvents the all-optical packet processing and switching problem by attempting network resource reservation using a control packet (CP) also known as Control Message, Burst Control Message, Burst Control Packet, Burst Header Message, Setup Message or Burst Header Packet, that crosses the network prior to its

corresponding data burst and is electronically interpreted at each switching node. This CP is optically sent into the network in a predefined channel used only for CP traffic, named control channel, and each CP undergoes optical-electronic conversion at each core node it must cross. The OBS versus OPS trade-off is the speed of OPS for the feasibility of OBS. It must be noted that the CP may be viewed as the header of the burst and so, the ensemble CP and burst would constitute a well-formed very large packet. OBS is an adaptation of ABT [41] (ATM Block Transfer), a standard for burst switching in ATM by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T).

Simpler optical circuit switching architectures, such as the "The Lightnet Architecture" [42] presented in section 2.6, show some potential as the advances in the technology allow the number of data channels in a fibre to increase. Actually, Wavelength Division Multiplexing equipments allow for about 160 data channels per fibre, although in 2005 the Nippon Telegraph and Telephone Corporation reported to have tested 1000 data channels in its 126 km testbed between the Kyoto Keihanna and the Osaka Dojima laboratories [43]. The availability of a high number of data channels, hypothetically each transmitting data at rates that range from 2.5 to 100 Gb/s in commercial or pre-commercial equipments, allows the provisioning high amounts of bandwidth for the core network infrastructure. Yet, as in any circuit switched architecture, the amount of unused bandwidth adds to the arguments in favour of the need of OBS and other more advanced switching paradigms stated previously here (*e.g.* OPS) and so, technologically OPS and OBS rank as the first choice when compared to circuit switching technologies.

One of the main problems with such OBS networks is the burst loss – this problem arises when two or more nodes try to reserve concurrently a resource along the paths of the bursts. In OBS networks, a resource is interpreted as a data channel, a wavelength converter or even switching resources at the switching matrix in the node. If contention to resource usage occurs, the burst is said to be dropped or lost. The minimization of the burst drop probabilities in a given network has been subject of intense research and there are a significant number of proposals to implement low

probability blocking networks, as discussed in section 2.7 - Approaches for contention resolution in OBS.

Figure 1 shows a schematic representation of an OBS network. Separately depicted are the control channel and the data channels, also referred to as lambda channels or λ-channels, an expression that derives from its underlying transmission technology, wavelength division multiplexing (WDM). In Figure 1 one can also see as an OBS node consists of a signalling engine and a switching matrix. Links are depicted connecting nodes; these are usually bidirectional and so, to reduce optical interference, each link is usually made of two different optical fibres – one conveying the input traffic to the node and other conveying the output traffic from the node. Each link comports at least a data channel and a control channel and may aggregate several fibres. Usually, it is assumed that link and fibre are equivalent concepts.

In OBS a node can be core or edge [44, 45] and each of these types play different roles – core nodes are expected to be simple and keep their activities to routing only and edge nodes are expected to perform other services, like creation of the CPs and all its related tasks, burst assembly and disassembly and do not route bursts. In Figure 1 the core nodes are placed inside the OBS network cloud and edge nodes interface with the client networks (two client networks are depicted). While edge nodes do not route bursts and are responsible for the burst assembly / disassembly and the CP creation, core nodes route the burst itself in a totally transparent way, implementing what is often called “all optical domain switching”.

A sample edge node consists of a device that interfaces with one or more client networks and with the OBS network. These client networks may be IP networks, Ethernet networks, ATM networks, and so on. The edge nodes perform burst assembly / disassembly and also the tasks that relate to routing the burst inside the OBS network, to allow for the creation of the control packet [45]. This control packet and the burst are then multiplexed into the output fibre in direction of a core node, or inversely, the control packet and the burst are demultiplexed from the input fibre and the burst is disassembled and its constituent packets routed into the client network. A burst is

therefore a set of data units (*e.g.* IP packets, Ethernet frames, ATM cells, *etc.*) that are grouped together by the edge node, following burst assembly rules and constraints.

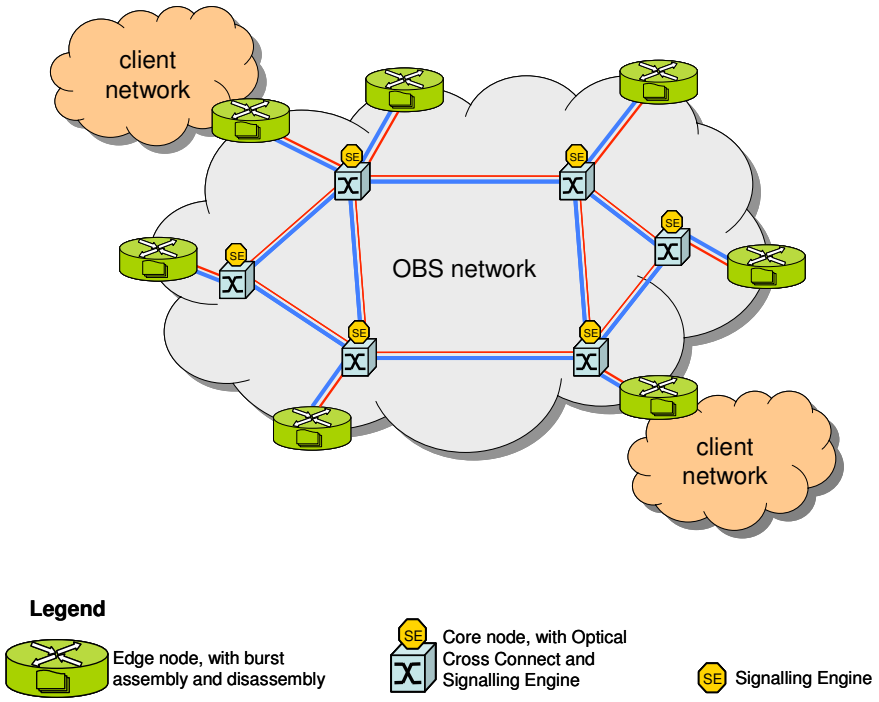


Figure 1 – Schematic representation of an OBS network.

Figure 2 shows a possible functional scheme for an edge node. There are two main features in this scheme: it does not have an optical cross connect matrix, as it does not perform optical switching, and it has two modules for burst assembly and disassembly that interface with tributary networks. Similarly to the core node, described forward, the edge node interfaces also with the fibres that connect the node to the OBS topology (see Figure 1). Input traffic from the tributary client networks is received by the interfaces that connect the node to these networks, and is assembled into bursts. In this process, the Signalling Engine manufactures the Control Packet and later commands the burst assembly module to transmit the burst into the switching matrix, where it will be switched to the appropriate output port in the appropriate fibre. Inversely, when the Signalling Engine receives a Control Packet announcing the arrival of a burst, it configures the electronic switching matrix to deliver the burst, now in electronic form, to the burst disassembly machine at the appropriate output port and

interface. Amongst companies that manufacture edge nodes are Alcatel, Ciena and Matisse Networks [46-48].

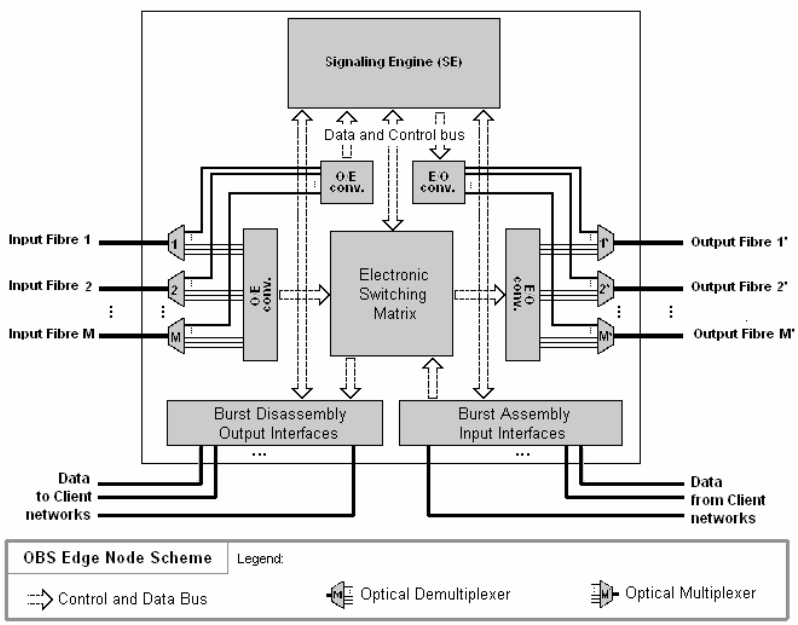


Figure 2 – A schematic representation of an edge node.

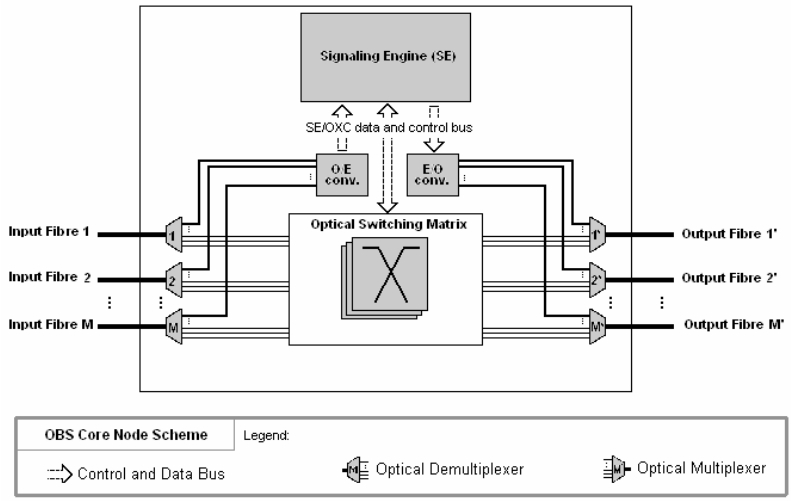


Figure 3 – A schematic representation of a core node.

A sample core node consists of a device that interfaces with several input and output fibres. The data channels and the control channel are de-multiplexed at the entry in the node and the signal the control channel carries is converted from Optical to Electronic form (O/E conversion) and interpreted at the Signalling Engine (SE). The SE is responsible for the implementation of the resource reservation schemes at the node, for the decision to drop of the incoming data burst and ultimately, for the control of the switching matrix [44]. While it is common that the SE is depicted as a separate part of the switching matrix, some authors (*e.g.* [49]) prefer to include it in the switching matrix. One will address it as a separate entity of the switching matrix, although both the SE and the switching matrix are part of the OXC, also loosely named OBS node.

Figure 3 shows a possible scheme of an OBS core node. Other architectures are possible, as for example "Labeled Optical Burst Switching for IP-over-WDM Integration" presented by Qiao and Staley in [50] or "Architecture for Optical Burst Switched Networks: Common Control Channel OBS - C³-OBS" proposed by Garcia *et al.* in [23]. Amongst companies that manufacture Optical Cross Connect nodes (OXC) are MEMX, Alcatel, Ciena and Matisse Networks [46-48, 51], while others such as Texas Instruments [52] produce components for the OXCs.

Some OBS test-beds have been deployed. Recently, Kim, Cho *et al.* [53] presented results for a WDM three ring OBS network with data channels of 2.5 Gb/s. Authors claim the demonstration was successful, allowing the operation of this network in a flexible manner either in circuit-oriented and packet-switched modes. Also recently, Sun, Hashiguchi *et al.* [54] published results of the creation of a testbed for an OBS network with three edge nodes and one core node. Authors used this network to transmit simultaneously two video streams to three clients. Edge nodes used two WDM channels and the core node used four WDM channels at 1.25 Gb/s. The bandwidth of the control channel was 100 Mb/s. JET protocol was used as the authors considered it would allow higher bandwidth utilization. According to the authors of [54] this was the first implementation and demonstration of a network wide test-bed with comprehensive OBS functions.

2.4. Optical burst switching network architectures

2.4.1. Basic architecture of an optical burst switching network

OBS basic architecture is based on the premise that data is aggregated into bursts and transported from an ingress edge node to an egress edge node in the network, by setting up a short-life light path in the network in such a way that the burst finds the path configured when it crosses the network nodes. This light path is set in such a way as to maximize the utilization of the resources of the network, *i.e.* the light path is configured only during the strictly necessary time for the burst transmission, after which it may be explicitly or implicitly destroyed. If the light path is to be explicitly destroyed, then either ingress or egress node will issue another CP with a message that will remove the configured status for that data channel in each of the nodes, otherwise, in the implicit release scenario, each node will compute (in the case of estimated release) or assume (in the case of reservation for a fixed duration) the time after which the channel is again free and available [55].

Some other basic architectural assumptions of OBS [56] are as follows:

Out-of-band signalling: The signalling channel undergoes electro-optical conversion at each node to allow CP data electronic interpretation. This means that in a fibre carrying *n* WDM channels, only *(n-1)* channels are used for data and one channel is used as the control channel, whose function is to carry the CPs (with signalling, QoS, routing and status functions and so on).

Data transparency: Data is completely transparent do the nodes in the network, as only the ingress and egress nodes need to be aware of aggregation and disaggregation schemes, data transmission rates, signal modulation and so on. A burst switched network acts as a time manager for a particular data channel, keeping the nodes oblivious to data handling details, this approach bringing obvious numerous advantages. OBS makes no assumption as to the type of data and so it schedules periods of time where the resources will be occupied.

Data and respective control packet are separated by a time delay: As the network messages in the CP are electronically interpreted in each node, thus requiring optic-electronic conversion, the total averaged transmission speed of the CP is slower than of its corresponding burst. As the light path needs to be configured in the node for the burst to route through, it is necessary to send the CP with some time in advance regarding the burst. This time, named *Offset time*, or T_{Offset} is calculated as a function of the time the CP needs to enter, be interpreted and exit the Signalling Engine (SE) in the OXC, named T_{Setup}, the number of nodes the CP/burst will cross in its path from the ingress to the egress node and the time taken by each node to configure the switching matrix in the OXC, named T_{OXC}.

Network intelligence at the edge: Burst assembly and disassembly, routing decisions are kept in the edge nodes – core nodes in the network are kept simple.

Asynchronous Functioning: Following the simplicity inherent to the OBS paradigm, nodes do not need to be synchronized.

A basic node architecture comprehends several input and output fibres, each with a number of WDM data channels, each fibre carrying its control and data channels. Figure 3 depicts the functional scheme for an OBS node with an arbitrary *m* number of input and output fibres.

There are a number of variations on the original OBS architecture, as for example the ones that:

- use of Fibre Delay Lines (FDL) as buffers, *e.g.* [57, 58], as presented in section 2.7.2;
- implement burst segmentation, *e.g.* [58, 59] as presented in section 2.7.3;
- perform burst and path grooming [60, 61], as presented in section 2.7.4;
- integrated burst assembly and disassembly with electronic buffering and burst add drop at nodes *e.g.* [50] and presented in section 2.4.3.

2.4.2. Dynamic wavelength-routed optical burst switching network architecture

Dynamic Wavelength-Routed Optical Burst Switched Network Architecture (DWR-OBS) was proposed in 2002 by Düser and Bayvel [62, 63]. It is a compromise between the Tell-and-Go OBS (TAG OBS) and Tell-and-Wait OBS (TAW OBS), as it proposes a node whose function is to act as a reservation request broker to the network. DWR-OBS elects one node in the network to evaluate the resource reservation requests from the edge nodes. This node, called Central Node, issues back *acknowledgement* or rejection messages in response to the requesting nodes, thus managing all the network resources. Analysis performed in [63] show this architecture can cope with until 115 nodes, thus making it suitable for medium size networks, according to the authors. This limitation rises because of the computational load posed on the central node, which must process all the requests from the all the nodes. Another limitation of this network is the burst assembly time, which must be long enough as to allow the request to travel from the ingress node to the central node and back. One of the common criticisms formulated to this architecture is that a single point of decision is failure sensible and that the collapse of such a node (or its connecting links) would render ineffective the whole network for the period of time until a new Central Node becomes active.

In [64] a pre-booking mechanism is presented, derived from the DWR-OBS architecture. Authors claim that for a 90 ms end-to-end delay with a 10^{-4} bit loss tolerance, the pre-booking mechanism yields approximately twice the traffic the DWR-OBS.

2.4.3. Other architectures

Labeled Optical Burst Switching (LOBS) as proposed in [50, 65] is viewed as a natural extension of the multi-protocol label switching (MPLS) framework for OBS [66, 67]. In this architecture the MPLS functionality serves as an integration layer between IP and the WDM, as shown in Figure 4. LOBS provides path provisioning, traffic and resource engineering, network survivability and several other features related to the MPLS framework.

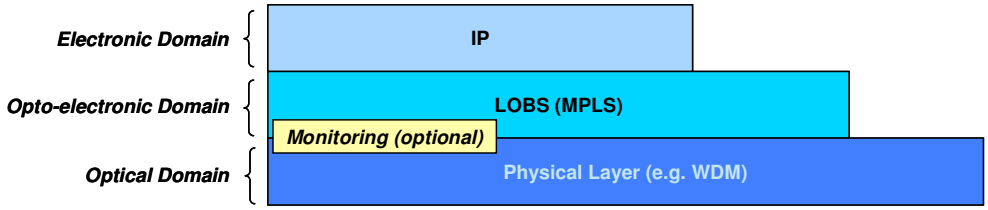


Figure 4 – An integrated IP over WDM architecture as proposed in LOBS [65].

In [65] a LOBS node architecture is proposed, in which incoming bursts may be locally disassembled and again assembled and reinserted into the network. MPLS messages are used to control burst switching, bandwidth reservation and mainly serve to reduce the complexities associated with defining and maintaining a separate optical (burst switching) layer. Additionally [65] also suggests that from LOBS based networks, migration and internetworking with optical packet switching will be easier.

The "Lightnet Architecture" as proposed in [42] is presented in section 2.6 since it may be viewed as a no-reservation protocol. This architecture implements light paths using the availability of WDM data channels, trying to maximize the wavelength continuity constraint along the source-destination paths, in order to minimize switching and processing effort inside the network. Figure 5 shows an example of the light paths for a 3 node ring network.

Chapter 6 presents a new architecture named Common Control Channel Optical Burst Switching C^3-OBS [23-25], which uses the information in the CPs to locally maintain a local model of the network reservation requests, implementing a distributed control model of the network.

2.5. Burst assembly of IP traffic

Data packet aggregation or burst assembly is a process where individual data entities such as IP packets, Ethernet frames, ATM cells, etc. are grouped together before the resulting data conglomerate, also termed burst, is sent into the network structure. The burst may be re-encapsulated (or not), depending on the supported network scenario. The encapsulation of the burst by an additional envelope adds routing and

processing capabilities to this mega-packet whose payload is the burst, *e.g.* the case where the burst constitutes the payload of an IPv6 packet or Jumbogram. If the burst is not re-encapsulated, then it has to be transmitted in a transparent manner to the network, because there is no available routing information in the conglomerate of bytes.

The nature and origin of its constituent data packets is not relevant to the burst assembly principle. The Burst Switching concept only requires the other end of the transmission channel to operate a complementary burst disassembly process, which retrieves the original constituent packets in order to route them further into the destination sub-networks.

Burst Assembly algorithms are constraint driven and fall into three categories:

1) Maximum Burst Size (MBS) [68];

2) Maximum Time Delay (MTD) [69];

3) Hybrid Assembly (HA) [70, 71].

Other burst assembly algorithms, *e.g.* considering classes of services, are typically built upon the aforementioned basic types. These basic burst assembly algorithms are presented and evaluated for real IP traffic in Chapter 4.

Some burst assembly algorithms address Class of Service (CoS) issues from the burst point of view. This is done in two different ways:

1. Packets that are associated with a given CoS are assembled to a specific burst queue and the resulting burst is assigned that given CoS, since all its constituent packets have the same CoS level [72];
2. Each burst contains packets associated with several CoS, usually ordered in some specific manner, *e.g.*, highest priority packets are placed at the head or tail of the burst or in a way that high priority packets are place at the centre of the burst [59, 73, 74].

There are problems associated to each of these burst assembly approaches, the more common of which is the complexity associated to the management of several

classes of service queues per destination address. In a simple calculation, if a network has 10 nodes (*e.g.* GÉANT 2 has 34 nodes [1]) and allows for 3 classes of service, this means that each node must maintain 27 queues following approach 1.

In networks which admit burst segmentation, a part of the burst may be dropped in case of resource contention, *e.g.* in OBS, where the burst travels in an all-optical form, transparent to the switching nodes. Vokkarane *et al.* proposed a generalized burst assembly algorithm [74] that places higher priority packets in the burst head, to allow possible burst segmentation and tail loss, *i.e.*, a bust includes packets with different priority constraints, sorted in a special order.

Dolzer proposed the Assured Horizon framework [75] for OBS networks. This framework includes a special bandwidth reservation scheme between ingress and egress point for each forward equivalence class, the policing of that bandwidth by the burst assemblers at edge nodes and to enforce it and also a traffic shaper / dropper at each core node, as the bursts are classified as Conformant or Non-conformant accordingly to QoS parameters. This framework was evaluated in [75] and addresses issues related to the implementation of QoS aware OBS.

A Round-Robin Burst assembler is proposed in [76] by Tachibana *et al.* This assembler accepts IP packets from a tributary network and sorts them into queues depending on their final destination. The several resulting queues are emptied in a round-robin manner, given the fixed time the algorithm takes to cycle all the queues. A queue is skipped if it is empty and kept if there are no available output resources. Authors claim that keeping the dequeuing at fixed intervals is more efficient than dequeuing at exponentially distributed intervals in an OBS sample network.

Long, Tucker and Wang [77] proposed a new framework to support DiffServ in IP over OBS. In this scheme, packets with different class of service tags are queued separately, with each queue adopting different adequate threshold values.

In 2004, Rodrigo and Götz presented a random burst assembly algorithm [78]. In this algorithm, each time a packet ingress the burst assembly queue, a Bernoulli random number generator defines whether the burst must depart or not. This leads to pure

poissonian burst arrival process and thus produces traffic that complies with previous research assumptions.

Considering CoS issue, some resource reservation protocols define different T_{Offset} times according to the priority of the bursts [79, 80], which in turn may be calculated either using *e.g.* a weighted or linear average of its packets stated CoS. Other resource reservation protocols allow for burst segmentation (see section 2.7.3) and in this case, it makes sense to keep the highest priority packets together in the area of the burst that is expected to be most successful to reach its destination.

In [73], Qiao proposes a hybrid burst priority (HBP) scheme for LOBS (yet applicable in OBS). In HBP, the priority of the burst is computed using a weighted average of the CoS field content of its packets. Inside the burst, HPB uses a special ordering scheme termed Nutshell, since the highest priority packets are placed in the centre of the burst.

2.6. Resource reservation protocols

2.6.1. Types of resource reservation protocols in OBS networks

The purpose of resource reservation protocols, also called wavelength reservation or resource reservation protocols [55] is to define how the OXC and the OBS network itself functions from an operational perspective. Resource reservation protocols are responsible for the definition, operation and maintenance of the local resource reservation state databases and for the flow of the control messages (namely its timing) inside the OXC.

There are three types of resource reservation methods in OBS: no reservation [42], unidirectional reservation, also said Tell-And-Go (TAG) reservation [81], bidirectional reservation, also said Tell-And-Wait (TAW) reservation [81]. There are some variants over these basic types, such as the Intermediate Node Initiated (INI) signalling [82] and the Dual Control Packet Signalling (DCP) [83]. Please note that most authors (*e.g.* [84, 85]) consider only two basic signalling methods, the TAW and

the TAG, probably because OBS may be viewed as an adaptation of ABT [41] and the TAG and TAW schemes were already used in ABT [55].

In the no reservation scheme, the burst is sent into the network without the preceding Control Packet (CP), much as in the circuit switched paradigm – if the circuit is set, then each burst can be forwarded without need to perform switching along its path. This approach uses the WDM technology, in particular, the availability of several data paths (lambda channels) in the same physical link to increase user available throughput and to simplify switching. A light path (data channel) is an optical path established between two nodes in the network, not necessarily adjacent, created by the allocation of the same wavelength throughout the path. Once the light path is established, intermediate nodes do not perform processing, buffering, or Electronic to Optic form (E/O) conversions. The use of light paths to establish circuits and thus transport packets reduces the overall network buffering and processing requirements when compared with a conventional store and forward network. This approach was initially devised as a novel optical architecture, termed "The Lightnet Architecture" [42]. On the basis of this architecture lies an integrated packet and circuit switching solution, as packets are routed over adjacent light paths and circuits are established using the available data channels on paths, for the circuits duration.

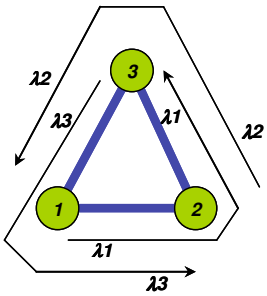


Figure 5 – Lightnet Architecture sample with three homogeneous lightpaths.

Figure 5 shows sample light path demands for a ring network. The efficient management of light path networks focus on solving two problems: firstly, wavelengths are a scarce resource and thus it is necessary to establish light paths efficiently in terms of the number of wavelengths used; secondly the requirement of establishing a light path using the same wavelength throughout the path introduces a potential bandwidth

waste, when compared to a light path establishment where this wavelength continuity constraint is not addressed, *i.e.*, if wavelength converters are available and used. A third problem arises with light path network operation, as the time needed to establish a light path circuit introduces additional bandwidth waste [42].

In TAG OBS each burst is preceded by a corresponding Control Packet (CP). Moreover, the burst ingress node does not wait for an acknowledgement (ACK) from any node and sends the burst after a given time, called *Offset Time* or T_{Offset}, defined as the interval of time, in the ingress node, between the first bit of the CP and the first bit of the burst [30].

In TAW OBS, the ingress edge node expects an *acknowledgement* from the egress edge node and issues the burst only after this acknowledgement. Expectedly, TAW offset time is bigger than its TAG equivalent. The trade-off between TAW and TAG is more offset time for fewer bursts dropped.

In either TAG or TAW OBS, T_{Offset} is a function of three variables: the number of nodes the burst must traverse, the time needed for the configuration of each node (assumed to be equal in all nodes), termed T_{Setup}, and the time needed for the configuration of the switching matrix in the OXC, termed T_{OXC}. Of course, the no-reservation model may be viewed as a TAG model with a T_{Offset} value equal to zero.

Control Packets (CP) contain the information that allows the functioning of the network defined resource reservation protocol [81]. The CP carries the following data:

- Ingress node address;
- Egress node address;
- Path, consisting of the addresses of the nodes the CP and the burst must visit or traverse, or a label that clearly identifies that path;
- Burst departure time from the ingress node;
- Burst length.

A different structure is proposed in [86], as depicted in Figure 6. Here, the routing information is carried by the label and wavelength ID fields, in particular, the

label encloses information related to the MPLS mechanism [87, 88] and the wavelength ID states the input port of burst.

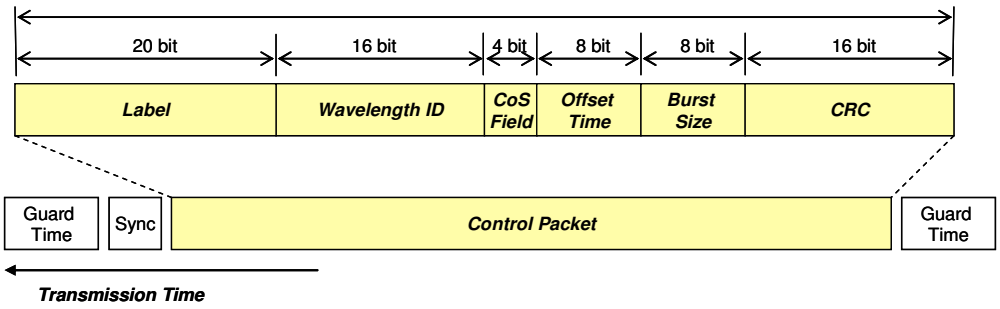


Figure 6 - Control packet structure as proposed by Oh and Kang in [86].

Other network functions such as neighbour discovery, routing information, network resilience related tasks and so on may be implemented either by the CP in the control plane, or by regular IP packets travelling inside bursts, addressed to the nodes (assuming that each node has an IP address for management and supervision functions).

Currently studied resource reservation protocols are, in chronological order of publication:

- Just Enough Time (JET) [30]
- Horizon [89]
- Just In Time (JIT) [90]
- JumpStart [56, 91]
- JIT$^+$ [84]
- Enhanced JIT (E-JIT) [92]

The goal of a resource reservation protocol is to define the manner in which the SE is to handle the OXC resource reservation. These well known protocols may be classified as to their approach to resource reservation, *i.e.* these protocols may perform immediate or delayed reservation and may perform void filling or not. Figure 7 shows a scheme that depicts each protocol as to their ability to handle resource reservation. JumpStart is placed on several classes because it can mimic any of the other protocols, *i.e.* it behaves differently in respect to its configuration parameters; JIT$^+$ is a delayed

reservation protocol, but it accepts two reservations at the most, as opposed to JET that accepts an unlimited number of reservations in a data channel. Note that there is no protocol performing immediate reservation and void filling, as this does not make sense, because if a channel is reserved up until a time, no other information is available as to allow a void filling process.

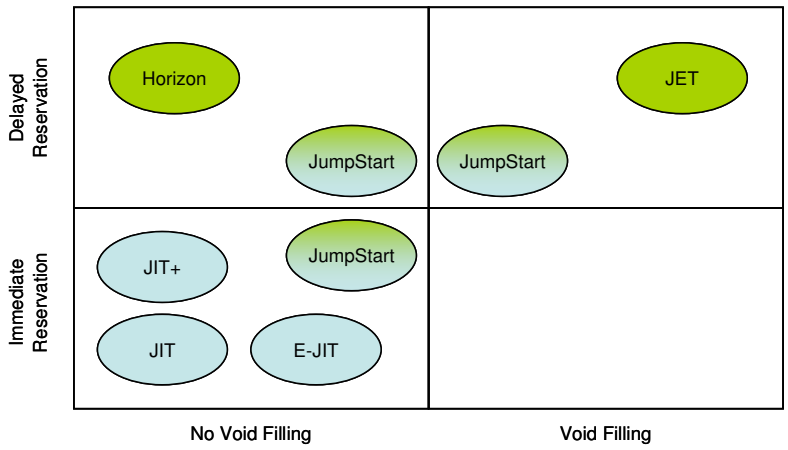


Figure 7 – Classification of resource reservation protocols.

Theoretically, void filling and delayed reservation protocols are much more efficient that non-void filling and immediate reservation protocols *e.g.* JIT. This assessment is questionable, since more complex protocols handle more complex databases and thus require more computational power, which in turn may degrade the performance of the OXC and of the network [84]. In fact, research seems to point out that their performance is similar when T_{Offset} and T_{OXC} are set to realistic values and thus implementation should prefer simpler protocols instead of more complex ones [55, 84].

The signalling on the network is performed by a control packet (CP), sent by the ingress edge node. This CP visits each node on the selected path and attempts the resource reservation. This attempt may or may not require *acknowledgement* to the ingress node and if this is the case, an ACK CP is issued by the egress node (or by an intermediate node) and travels back to the ingress node. If no acknowledgement is required, the OBS is said Tell-And-Go (TAG) [81], otherwise, is said Tell-And-Wait

(TAW) [81]. There are two known variations to TAG and TAW OBS – Intermediate Node Initiated Reservation (INI) [82] and Double Control Packet (DCP) [83].

TAG-OBS sends the burst into the network without acknowledgement of any kind and T_{Offset} is calculated as shown in (1), following the principle depicted in Figure 10 and Figure 11. One sees that

$$T_{Offset} = T_{OXC} + n.T_{Setup} \tag{1}$$

where n is the number of core nodes the burst / CP will cross in the defined burst path and T_{OXC} and T_{Setup} have been defined previously.

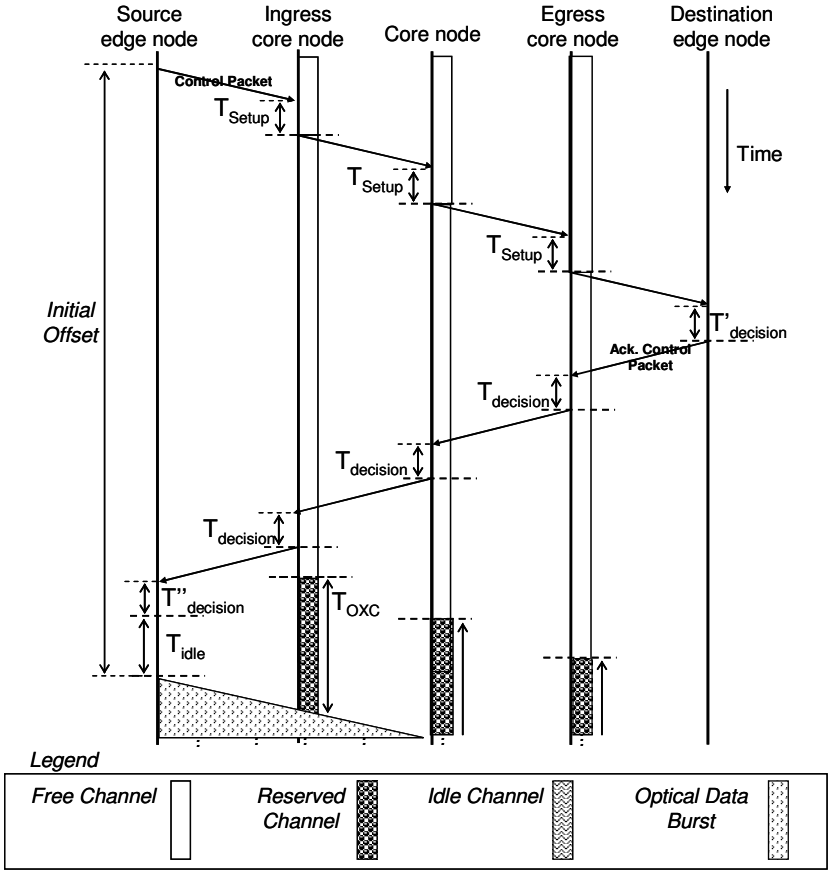


Figure 8 – Schematic representation of messages and burst transmission for a Tell and Wait OBS network with five nodes.

In TAG OBS (Tell and Go OBS), the burst is sent into the network without any feedback from the nodes that the resources are available and thus, it may drop at any of its path nodes. When comparing the performance of TAW OBS and TAG OBS, the burst drop probability for TAG OBS is lower for long networks (with hundreds of km) and slow optical devices or for metropolitan networks (with shorter links) with faster optical devices [85].

In TAW OBS (Tell and Wait OBS), the ingress node waits for an *acknowledgement* message from the network confirming that the resources have been successfully reserved. Usually, delayed resource reservation is foreseen, because immediate reservation would cause the reservation time of the channel to be bigger than demanded by the burst and ultimately, increase the inefficiency of the network.

In case of delayed reservation, (2) shows the equation for the calculation of T_{Offset} and as it can be seen from Figure 8, follows

$$T_{Offset} = T_{OXC} + 2n.T_{Setup}. \quad (2)$$

If immediate reservation is used, equation (2) computes the worst case, assuming that the T_{OXC} is bigger that $2n.T_{Setup}$ and also that T_{Setup} is always bigger than $T_{decision}$, $T'_{decision}$ and $T''_{decision}$. Currently, T_{OXC} is about a few milliseconds and T_{Setup} is around a few tens of microseconds [46, 47, 51, 55].

$T_{decision}$ is the time the node takes to receive, perform O/E conversion, decide that this is an *acknowledgement* message and reinsert this CP in the control channel in a optical form (E/O conversion); $T'_{decision}$ is the time the destination edge node takes to receive, convert O/E, decide to accept the reception of the burst and finally create the acknowledgement CP (ACK CP in Figure 8), issuing it in optical form to the control channel and finally, $T''_{decision}$ is the time the source edge node takes to receive the ACK CP, convert it and process it and decide to issue the burst, if the CP has a positive acknowledgement. Please note that in TAG OBS, this final decision does not make any sense, since the decision to issue the burst into the network is part of the definition of

TAG OBS – the burst is always to be inserted into the network, even if it will drop in the next hop.

For the next calculations, one can assume that the worst case is that

$$T_{decision} = T'_{decision} = T''_{decision}. \text{ all equal to } T_{Setup,} \quad (3)$$

because T_{Setup} is expected to be higher than any $T_{decision}$ as the first requires at least a database query.

Although Figure 8 is not shown at scale and taking (3) into consideration, one can see that if T_{OXC} is such that

$$T_{OXC} < (2n+2).T_{Setup} \quad (4)$$

then T_{Offset} can be calculated as follows

$$T_{Offset} = (2n+2).T_{Setup}, \quad (5)$$

because the configuration of the OXC can occur while the CP is travelling to the egress node, plus the correspondent ACK CP travels back to the ingress node. This is of course highly dependent on the number of nodes the CP is expected to travel – it may happen that node at the first hop in the path decides that it has no resources to allow the burst to pass at the desired time and data channel.

If

$$T_{OXC} > (2n+2).T_{Setup}, \quad (6)$$

then T_{Offset} is

$$T_{Offset} = T_{OXC} + nT_{Setup}\,. \tag{7}$$

Combining equations (5) and (7), one can say that the worst case value for the Offset time in TAW OBS is

$$T_{Offset} = \max\left\{ (2n+2).T_{Setup}\,,\; T_{OXC} + n.T_{Setup} \right\}. \tag{8}$$

Figure 8 shows the reservation scheme for delayed reservation mechanism. In this figure, T_{idle} is depicted at the source edge node to allow the correct formulation of the figure with equation (8). If $T_{OXC} < (2n+2).T_{Setup}$ (expression (4)), then T_{idle} may be zero.

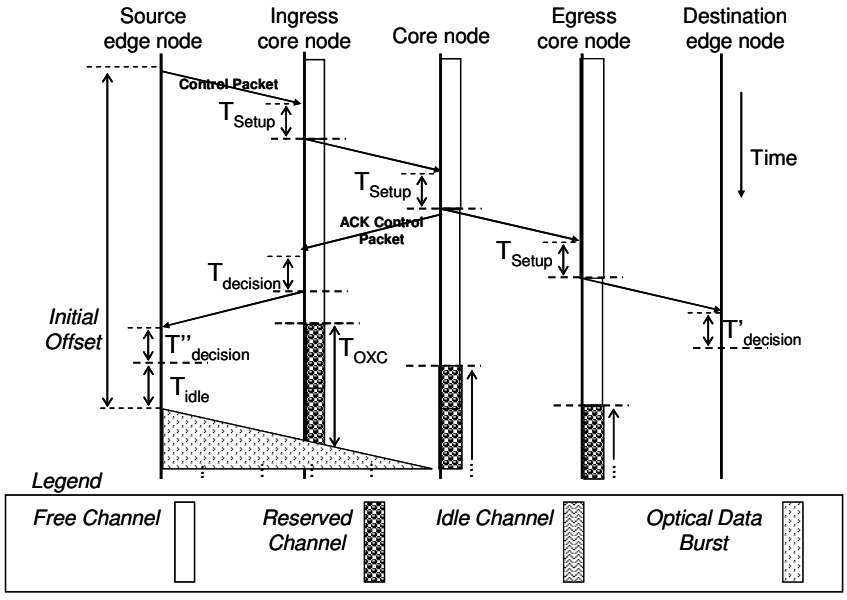


Figure 9 – Schematic representation of messages and burst transmission in an Intermediate Node Initiated (INI) Reservation OBS network with five nodes (configuration of OXC is not shown for all the nodes).

In Intermediate Node Initiated Reservation (INI) [82], one of the nodes in the path is responsible for the *acknowledgement* message. Figure 9 shows how INI works in a sample five node network (three core nodes), establishing the middle node as the intermediate node. The proposal of INI stands on the assumption that having half way guarantee on resource reservation is better than to have no guarantee at all. The TAW offset is reduced to half (considering equal length links and middle intermediate node) and the transmission experiences less burst drop due to reduction of the burst drop in the first hops [82], as it is a well known fact that burst loss ratio increases with path length, a QoS problem known as lack of fairness [93].

Following Karanam, Vokkarane *et al.* in [82], the INI offset time is calculated as follows:

$$T_{Offset} = \max\left\{ (n+1).T_{Setup} , T_{OXC} + n.T_{Setup} \right\} \quad (9)$$

where n is the number of core nodes the CP and the burst must cross and T_{Setup} and T_{OXC} have the usual meaning. Equation (9) assumes that the middle node is the intermediate node responsible for the ACK message, although INI does allow any node along the path to assume that role, independently of being the first or the last in the path. If the intermediate node is defined as the last, INI falls to TAW OBS. To allow coherent formulation between Figure 9 and equation (9), T_{idle} is also depicted.

In [94] it was proved that following a service isolation scheme, for a given user-defined burst priority, high priority bursts would be more successful if its offset time was bigger than low priority bursts.

DCP reservation [83] is set to solve the problem with small or low priority bursts, adding another level of channel rescheduling by issuing two different CP. The first CP carries all the information about the burst except its input channel. Its function is to initiate the Fibre Delay Line (FDL) buffering mechanism, thus allowing the node to rearrange the FDL buffers, if the arriving burst can not be fitted in the latest available unused channel (LAUC algorithm [31], see section 2.7.6 - Wavelength assignment algorithms). The second CP carries the burst ID and its input channel, being the burst ID

common to both CPs. If the pre-reservation stage is successful, the node forwards the first CP to the next node and issues a confirmation (the second CP) to the next node. This confirmation already knows the output channel for the burst on this node and coincidently, the input channel for the burst in the next node. Li *et al.* in [83] claim that DCP achieves 5 to 10 times less burst drops when compared to single CP OBS for network loads of 75%.

2.6.2. Immediate versus delayed reservation

When attempting to reserve a network resource, *e.g.* a data channel or a wavelength converter, the SE may use one of two techniques – either the resource is set as occupied starting from the moment its availability is queried by the Control Packet (CP) or the resource is defined as occupied only during the time the burst needs to use that resource. The first approach is termed immediate reservation, while the last is termed delayed reservation. Figure 10 and Figure 11 show how immediate and delayed reservation affects the status of a given data channel in the networks. Relating this figure with Figure 3, the ingress core, the core and the egress core nodes could be any of the 6 nodes depicted in Figure 1 and the source and the destination edge nodes would be users (in a wide sense) that being attached to the network, are responsible for creating (assembling) the bursts and its control packets (as source node) and disassembling the bursts (as destination node).

In a delayed reservation (Figure 11), if the channel is free, the protocol may try to fit an extra burst in between reservations. In an immediate reservation scenario, the data channel remains idle from the time the CP is interpreted and thus, any other CP that tries to reserve that same data channel will be unsuccessful, causing its burst to drop. Note that in an immediate reservation scheme like the one shown in Figure 10, after the node has configured its OXC, the data channel is idle but not free – no other reservations can be attempted on this data channel. In a delayed reservation, the channel is free until the SE decides to configure the OXC, which happens just before the corresponding burst is expected to arrive.

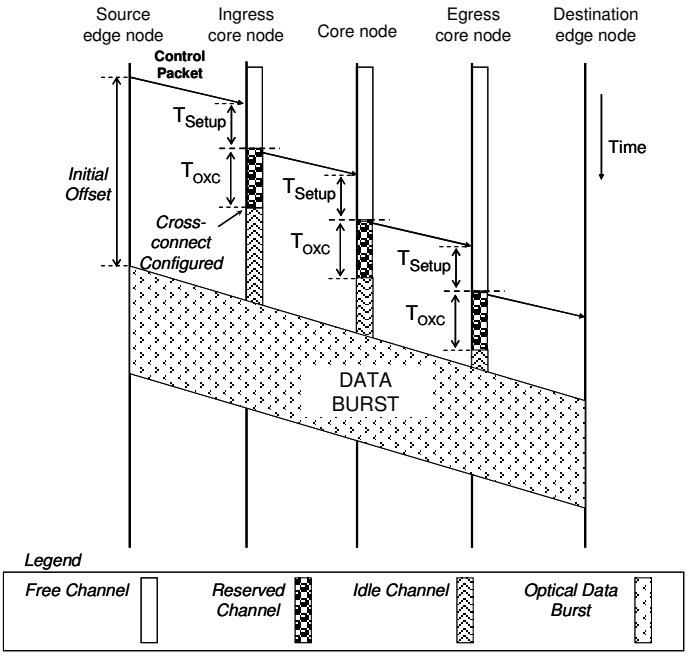


Figure 10 – Schematic representation of the signalling messages in an immediate reservation protocol (*e.g.* JIT, JIT^{+} and JumpStart) OBS network with five nodes.

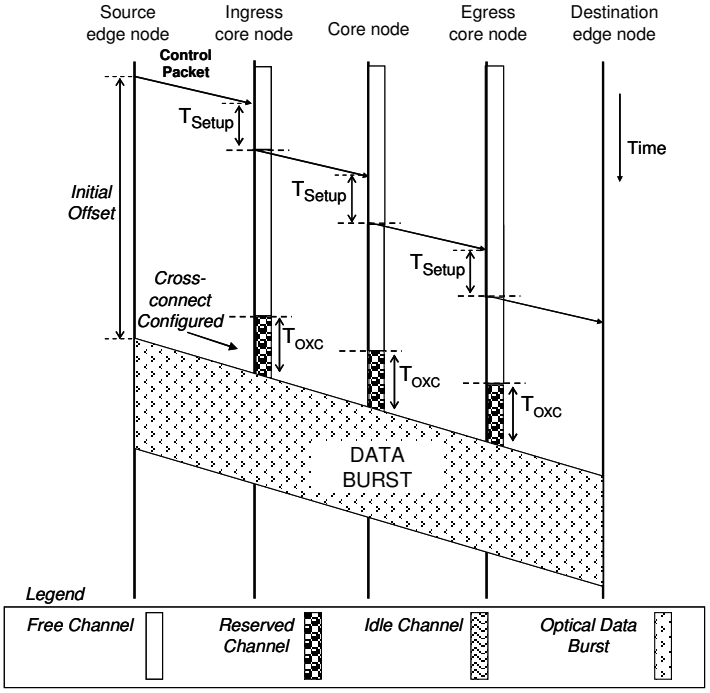


Figure 11 – Schematic representation of the signalling messages in a delayed reservation protocol (*e.g.* JET, Horizon and JumpStart) OBS network with five nodes.

2.6.3. Void filling versus no void filling

When using delayed reservation, a protocol is said to implement void filling if it attempts to reserve resources for a burst transmission over the free space on a data channel that has already other burst reservations. JET attempts void filling, consisting this in a query to the SE database, trying to find a suitable interval of time in any data channel (if wavelength converters are available at that time) for that particular burst.

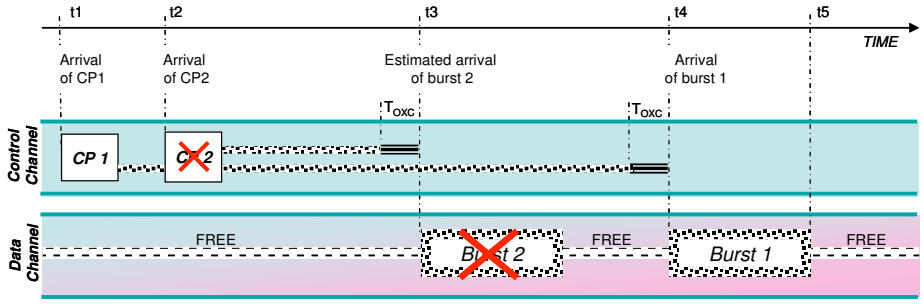


Figure 12 – Scheduling two bursts using no-void-filling signalling protocols (*e.g.* Horizon, JIT or JIT^{+}).

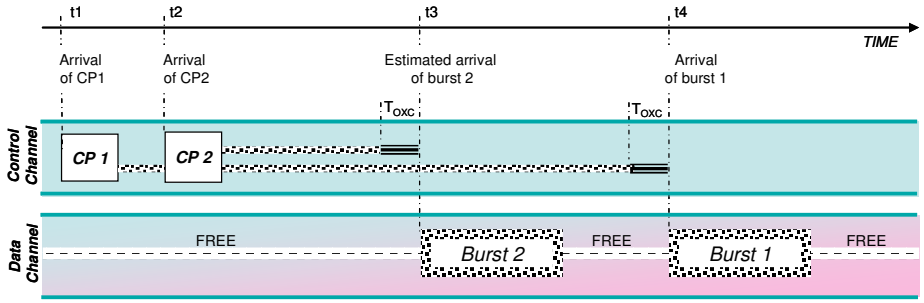


Figure 13 – Scheduling two bursts using void-filling protocol JET.

Figure 12 and Figure 13 depict the same scenario: two bursts (labelled *Burst 1* and *Burst 2*) are trying to use the same output resource in a node. For each of these bursts, a CP was issued (*CP 1* and *CP 2* respectively) and *CP 1* arrives to this node before than *CP 2*. As T_{Offset} for *Burst 2* is smaller than T_{Offset} for *Burst 1, Burst 2* is expected to arrive sooner to this node. In a no void filling scheme like Horizon, only the first CP will make its reservation good, although there could be space for other bursts to

be scheduled. Note that at time t_1+T_{Setup} the node expects *Burst 1* to arrive at time t_4 and before that, the data channel is free. In a void filling scheme like JET, the node will try to find a free data channel space in order to allow the expected arrival time of the burst to be smaller than the latest arrival minus the T_{OXC} (it still has to configure the OXC correctly before the previously scheduled burst arrives).

Figure 12 shows how Horizon, JIT or JIT$^+$ behave and Figure 13 show how JET behaves rearding void filling. In these figures, the data channel is said to be free when no burst is using it, although its conventional state would be FREE or RESERVED or IDLE, depending on the used protocol. As the data channel status is dependent on the type of reservation performed, Figure 10 and Figure 11 show its correct conventional state.

2.6.4. Just Enough Time (JET) resource reservation protocol

JET was proposed by Qiao and Yoo [30] in 1999 and works as follows:

> a given output data channel is reserved to a burst if the arrival of that burst happens after the time horizon for that data channel, ***or*** the arrival of the burst happens in a free time of the data channel and its length plus the T_{OXC} happen before that free time ends. The time horizon is defined as the time of arrival of the latest burst plus its length.

JET allows the implementation of a non-First Come First Served (non-FCFS) service. If all T_{Offset} times are similar, then, the serving order of the bursts in the OXC tends to be similar to the arrival of its respective CPs. Figure 13 shows how two consecutive bursts are scheduled over the same data channel following the JET protocol. In this figure, *CP 1* refers to *Burst 1* and *CP 2* refers to *Burst 2*. In this example, T_{Offset} of *Burst 1* is much longer than T_{Offset} of *Burst 2*. Using JET, one can see that *Burst 2* still finds a free channel space to allow for resource reservation, although its CP arrives after the CP for *Burst 1*. Void filling algorithms are used to optimize the occupancy of each data channel. JET does no poses any restriction on the number of scheduled reservations and thus, its internal database may grow beyond a desirable size. Figure 11 shows a sample JET operation.

2.6.5. Horizon resource reservation protocol

In 1999, Turner [89] proposed the Horizon protocol. This is a delayed reservation protocol that does not perform void filling. Turner used the name *Horizon* because there is a time horizon associated to each data channel, before which no reservation can be made. This time horizon is defined as the earliest time that the channel is known to be free.

In Horizon, reservation is done as follows:

> an output data channel is reserved to a burst only immediately before the first bit of the burst is expected to arrive; if upon arrival there is no available data channel, then the CP is rejected and the burst is dropped.

In another interpretation,

> a data channel is reserved to a burst if the arrival time of that burst is greater than the time horizon for that data channel; if upon arrival of the CP the arrival of its burst is expected sooner than the horizon of the data channel, then the CP is rejected and the burst is dropped.

As Horizon does not perform void filling, bursts are served as they arrive. When a burst is scheduled, the horizon of the data channel is set to the arrival time of burst, plus its duration plus T_{OXC}. So in Horizon, a burst can only be scheduled if its first bit arrives after the departure of the last burst plus the OXC configuration time.

2.6.6. Just in Time (JIT) resource reservation protocol

Wei and McFarland proposed JIT in December 2000 [90]. JIT implements an immediate reservation scheme without void filling.

JIT works as follows:

> an output data channel is reserved to a burst as soon as its CP arrives; if the data channel cannot be reserved, then the CP is rejected and the burst is lost.

JIT is purely FCFS and the channel is said to be idle (but not free) as soon as a successful reservation is made (see Figure 10). JIT is the simplest reservation mechanism, as it does not requires the maintenance of a database other than an variable storing the value for the time after which the channel is free.

2.6.7. JumpStart resource reservation protocol

The project Jumpstart started in 2002, financed by the American Research Development Association (ARDA), the North Carolina State University (NCSU) and the MCNC, a research institute founded in 1990 by the General Assembly of the State of North Carolina, all from the United States of America (USA). The JumpStart protocol [56, 91], proposed in 2002 by Baldine, Rouskas, Perros and Stevenson, is based on JIT, but adds the following characteristics:

- Allows QoS implementation
- Has Multicast support
- Allows label switching
- Allows for persistent connections (light paths)

In the JumpStart protocol, the edge node issues a message to the ingress node when the burst is waiting for transmission. If the ingress node can switch the burst, it replies to the edge node with an acknowledgement (ACK) message and forwards to control packet to the core nodes in the network. If the ingress node can not switch the burst it issues a negative *acknowledgement* (NAK) to the edge node and the burst is dropped.

JumpStart uses the messages in the CP to define the behaviour of the OXCs in the network. The message can be used to define the way the path is reserved, performing *estimated release* or *explicit release*. In the *estimated or implicit release* scheme, the resources are released as soon as the burst is assumed to have been transmitted (Figure 14). In the *explicit release* scheme, the edge node issues a new control packet to release the previously reserved network resources (Figure 15). While the circuit is established (in a light path-like manner) the source edge node can send any number of bursts to the destination edge node.

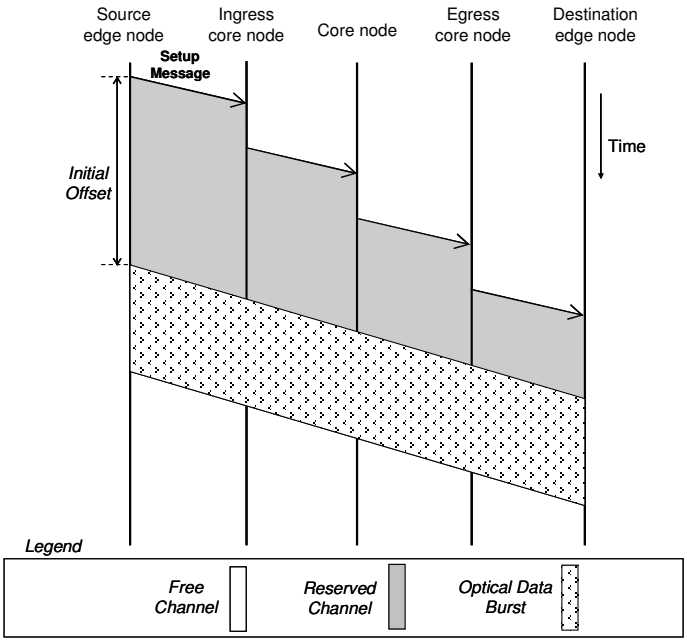


Figure 14 – Schematic representation of messages and burst transmission depicting implicit or estimated release in an immediate reservation scenario for an OBS network with five nodes.

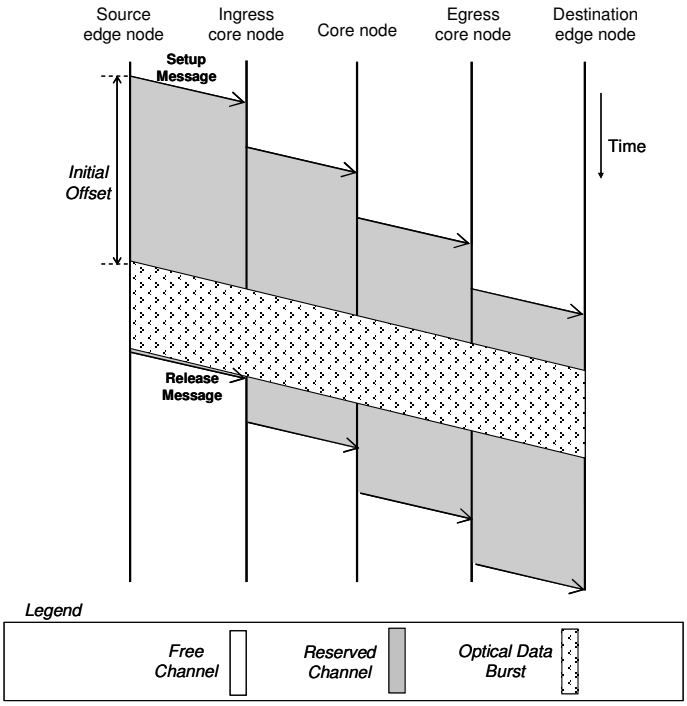


Figure 15 – Schematic representation of messages and burst transmission depicting explicit release in an immediate reservation scenario for an OBS network with five nodes.

2.6.8. JIT+ resource reservation protocol

JIT+ was proposed in 2003 by Teng and Rouskas [84] and is designed as an improvement to JIT as follows:

> a data channel is reserved for a burst if the arrival time of that burst happens after the time horizon of that data channel and if that data channel has at the most another reservation.

JIT+ does not attempt void filling and tries to improve JIT performance allowing at the most two reservations on a given wavelength. While JET, Horizon and JumpStart allow for an undefined number of reservations over a data channel, JIT+ keeps this number to two – thus simplifying the underlying database structure and its operational algorithms. Figure 16 shows how JIT+ handles three successive reservation attempts. It can be seen that although the channel could accommodate the third burst, its CP is rejected and the burst is dropped. The third burst would have been accepted in Horizon or JET.

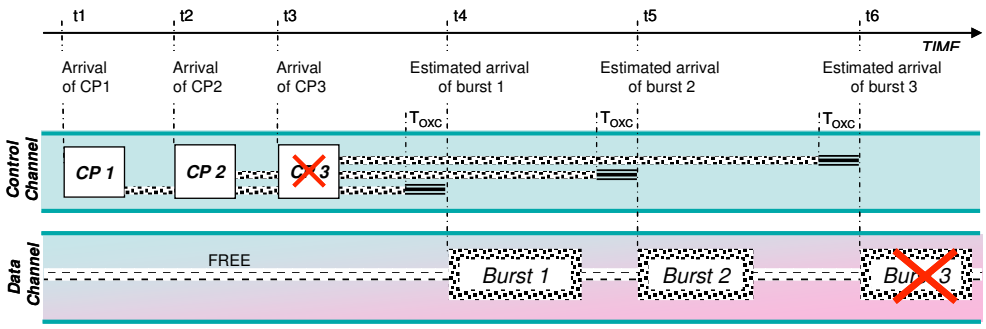


Figure 16 – Scheduling three burst in JIT+.

2.6.9. E-JIT resource reservation protocol

E-JIT stands for Enhanced Just-in-Time and was presented by Rodrigues *et al.* [11, 92]. This protocol is based on JIT and takes advantage of its relative simplicity, while keeping the advantages shown in JIT+ by allowing at most two simultaneous reservations. E-JIT improves the JIT protocol by implementing an improved data

channel scheduling scheme, decreasing the period of time in which the data channel remains in "reserved" status. This results in optimized channel utilization and in a potentially reduced burst loss probability.

JIT considers that the resources are free to a new reservation when the data channel status is *Free*. Yet, this can be detrimental to networks that operate very small bursts, as these need more CPs and request more configuration tasks in the nodes. OBS takes advantage of fewer switching and setup events achieved by the statistical multiplexing resulting from the aggregation of the packets which share a common set of constraints at the edge nodes (burst assembly tasks), but in the limit, if the transmission in an OBS network as to be done for very small bursts, *i.e.*, almost in a packet by packet basis, OBS performance will converge to that of regular O/E/O networks. Thus, for OBS networks, the ratio of number of bursts over number of transmitted packets in the burst is desirably much smaller than 1. Burst assembly efficiencies are discussed in Chapter 4, section 4.4.

The increased efficiency of E-JIT results from a better understanding of node reservation timings. E-JIT assumes that a channel is reservable (as opposite to free in JIT) if the OXC configuration can start immediately after the last bit of the burst has passed the node. Figure 17 shows E-JIT operation, depicting two reservations and a failed attempt to reserve the third burst, as JIT$^+$ would do. It can be seen that, under JIT or JIT$^+$, the control packet to attempt reservation for burst B would only be accepted after time t_6. In E-JIT, as the time of arrival of the CP for Burst B plus the T_{Setup} is higher than the estimated time of departure of the previous burst from the node (in Figure 17, Burst A departs the node at time t_6), the reservation request for Burst B is accepted.

The performance evaluation of this new protocol was performed in [92] and its performance is expected to increase for networks with high dominance of shorter bursts, as these process more CPs and thus benefit from the new protocol improved channel efficiency management.

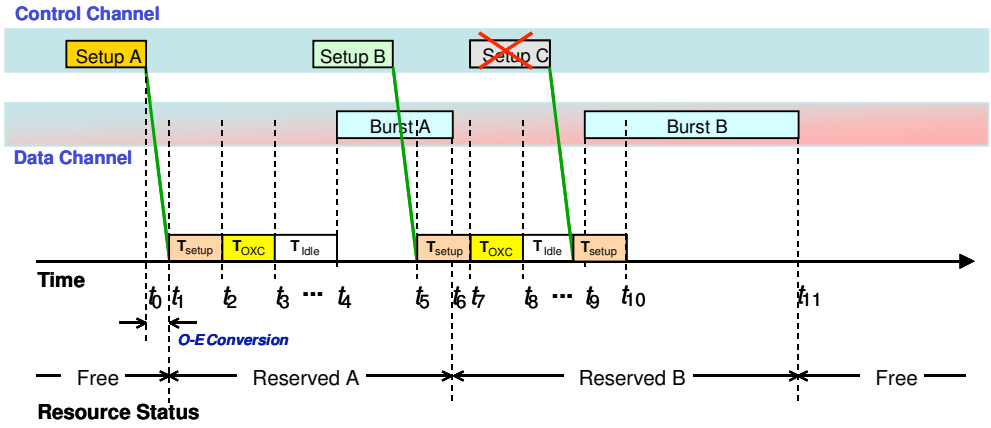


Figure 17 – Operation of E-JIT resource reservation protocol (rejecting a burst).

2.7. Approaches for contention resolution in OBS

As one of the major problems in OBS networks is contention resolution, some of the devised strategies to solve it are presented here. The strategies may be implemented independently of the OBS architecture and thus deserve a separate analysis, although many of these approaches fall out of scope of these thesis.

2.7.1. Prioritization

Apart from strict contention resolution on OBS, there are a number of strategies which focus on prioritization of burst traffic, aiming to improve the probability of success of higher priority bursts. The main reason to give different priorities to different bursts is that some traffic is less sensible to loss and delay than other. If two bursts try to reserve the same resource, the highest priority one should be allowed to complete the reservation. Generally speaking the resource reservation protocols are priority agnostic. There have been several proposals of resource reservation protocols concerned if the Quality of Service (QoS) issues in OBS. In 2000, Yoo and Qiao proposed pJET [94], which assigns larger T_{Offset} times to higher priority bursts in order to obtain better resource reservation probabilities. Other schemes (*e.g.* using channel differentiation [95]) have been object of research but are out of the scope of this thesis.

2.7.2. Optical buffering

If optical buffering was mature enough to allow a random delay memory, optical packet switching could be partially implemented. As it is not yet possible to use fully random optical buffering, this approach is limited to the use of Fibre Delay Lines (FDL). Typically an FDL is a special type of fibre which allows a limited fixed delayed transmission of the optical signal. This delay is proportional to the nature of the fibre and to its length and as an example, a 2 km standard FDL is needed to produce a 10 s delay [96]. Also and due to optical signal regeneration issues, authors in [96] expect a reachable maximum delay of 260 s per burst.

To allow a mean of comparison, Table 1 shows the transmission times of bursts for various burst sizes at different transmission speeds. A FDL as presented in [96] could store a complete 1 MB burst transmitted at 40 Gb/s.

Table 1 – Burst transmission time (in s) for different transmission ratios and different burst sizes.

	2.5 Gb/s	***10 Gb/s***	***40 Gb/s***	***100 Gb/s***
9 KB	29.491	7.373	1.843	0.737
64 KB	209.715	52.429	13.107	5.243
1 MB	3 355.443	838.861	209.715	83.886

The use of FDL has been explored as it can potentially improve the network throughput and reduce the burst loss probability [97]. As the use of FDLs may increase the delay between the burst and its control packet, signalling and reservation schemes need to be revised and updated. Gauger describes two different FDL scheduling mechanisms [98]: *PreRes* and *PostRes*. In the *PreRes* scheduling mechanism, the request to reserve a FDL buffer is made as soon as the burst control packet is processed and the node finds that there is no available wavelength on the required output port. In this scheme, the T_{Offset} is increased by the amount of delay the FDL provided. In the

PostRes scheme, both the burst and the control packet are delayed before the assessment of the availability of resources and thus the original T_{Offset} is preserved.

In [99] Li and Qiao propose the combined use of electronic buffers and FDLs to minimize burst loss at edge and core nodes of an OBS network, respectively. This patent claims to be applicable to Labeled Optical Burst Switched (LOBS) [65], OBS and OPS networks. It reports that contention and loss are reduced because of the effect caused by the local electronic buffering of the bursts and also because at an intermediate node, bursts that are already in transit may be delayed long enough to avoid contention using FDLs.

2.7.3. Burst segmentation

Burst segmentation [58] is a form of optimization of the available network bandwidth, as this mechanism falls under two different categories: QoS strategies and contention resolution. It follows the principle that is better to receive part of a burst than not to receive the burst at all. If a burst is currently transmitting and there is a request to reserve its resources, then part of the burst may be dropped. This schemes aims to drop the part of the burst that contains the packets that have low priority, allowing the high priority packets in the burst to still be transmitted. In [59], Vokkarane and Jue proposed a combined scheme using segmentation and deflection (see section 3.4.1) of the overlapped part of the burst, in order to further increase the efficiency of the network.

2.7.4. Burst and traffic grooming

Burst and traffic grooming are strategies to further improve OBS network performance. Burst grooming has been described as the alignment of the several bursts that have the same destination or share a large part of the path but possibly not the same source, and are transmitted close in time [60]. If a burst is to be a fully groomed, then the ingressed node will wait a short amount of time before another burst with the same final destination and it will align this new burst in the same output channel with the already transmitting one. Partially groomed bursts only share part of the path and thus, their final destination might not be the same. This scheme may require the use of some form of optical buffering such as a Fibre Delay Line (FDL).

Traffic grooming [61] is an approach to a problem that can be formulated as follows: given a network configuration and a set of path requests with different bandwidth requirements, it is necessary to define a set of light paths as to simultaneously maximize the success of path requests and minimize the configuration effort on network resources. This approach is related to the path accommodation strategy presented in the next section.

2.7.5. Routing strategies and algorithms

Routing Algorithms (RA) are responsible for the selection of the optimal path and can be classified following a series of key features. The selection of suitable paths for traffic allows the prevention of traffic bottlenecks in the network, that ultimately cause packet / burst delay and or loss. When a network device needs to compute the next hop or the full path for a given data packet (or stream of packets, in the case of a circuit definition), some metrics are used to define the optimal property of the candidate paths. These metrics may include number of hops (or path length), the reliability, the available bandwidth, the load applied to the path, the delay the path will inflict to the packets, or even the monetary cost of using that particular path. If all constraints are considered identical for each of the possible hops in the network, only one constraint remains as a selection criterion – the length of the path (or number of hops, if all links are supposed equally lengthy). This length will be a function of the remainder constraints. For instance, it will be proportional to the monetary cost of the path, inversely proportional to the path reliability, etc. The Dijkstra algorithm [100-102] is the *de facto* standard for shortest path route selection, mostly due to its simplicity and efficiency. Popular network simulators as *ns-2* [103], *OWns* [104], *OBSim* [105], *OIRC OBS-ns Simulator* [106] and others rely on the Dijkstra algorithm to compute the static paths that are used throughout the simulation process.

Chapter 3 provides an overview of current routing algorithms and strategies for OBS networks, and presents the new Extended Dijkstra, the Next Available Neighbour complementary routing algorithm and the Travel Agency Algorithm.

Path accommodation is a technology proposed by Nagatsu *et al.* in 1996 [107, 108]. By using wavelength division multiplexing and wavelength routing, Path Accommodation is a heuristic algorithm that tries to define wavelength paths or virtual wavelength paths in the network. In the wavelength path scheme one wavelength is assigned to each path from beginning to end and in the virtual wavelength path scheme the wavelength assignment is made link by link for each path, which may require some wavelength conversion at the intermediate OXCs. These two schemes are shown in Figure 18 and Figure 19. In these figures, the beginning and end of the path are showed with a rounded connector and each colour or pattern represents a given wavelength. Path Accommodation is a somehow derived approach of the Lightnet Architecture [42] presented in section 2.6.1.

Path Accommodation was initially defined as a heuristic algorithm with the goal of minimize the use of wavelength conversion at the network nodes and as a result, it contributes to produce the Path Accommodation phenomena, *i.e.* in traffic-aware architectures like C^3-OBS it is possible to schedule bursts in such a manner that path can be defined as to minimize the configuration effort on core notes, thus reusing some of the paths previously defined, even if this means one or two additional hops into the local path the bursts has to travel. The Travel Agency algorithm presented in detail in section 6.2.4, allows for such operation, for example when the selection of the path with least configurations is done (see also section 2.7.4). As network traffic increases that is expected that Path Accommodation allows for a decrease in the effort for configuring the OXCs.

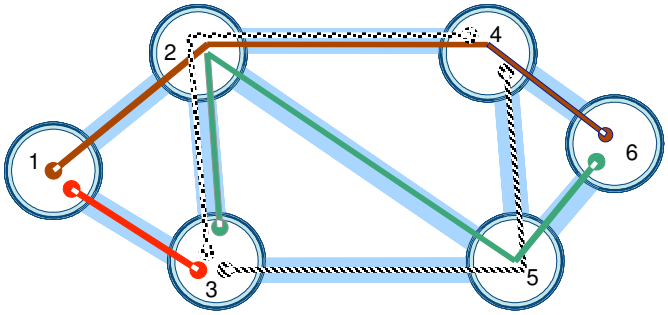


Figure 18 – Wavelength Path scheme showing five paths using different wavelengths.

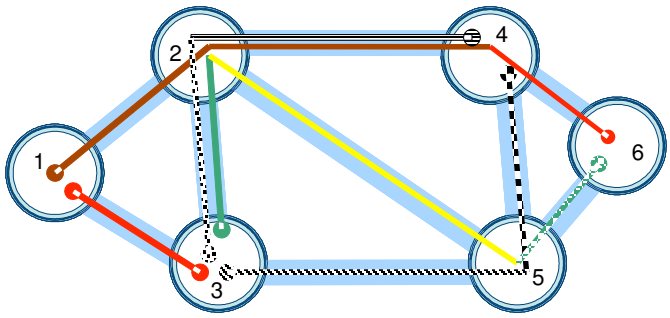


Figure 19 – Virtual Wavelength Path scheme showing five paths using different wavelengths in the same path.

2.7.6. Wavelength assignment algorithms

Wavelength assignment algorithms have been proposed as a way to overcome the simplifying assumption that OBS networks are capable of full wavelength conversion [109]. Full wavelength conversion is an important feature in OBS networks since it removes the wavelength continuity constraint, although wavelength conversion is not yet fully available [110]. Without wavelength conversion a node can only forward an incoming burst to an output port *if and only if* the wavelength carrying the burst is available on that output port. An adequate choice of wavelengths for a burst transmission helps to reduce the amount of resources needed in the transmission, *e.g.* the wavelength converters, where contention may also occur. Wavelength assignment algorithms fall into two categories: Non-Adaptive Wavelength Assignment and Adaptive Wavelength Assignment, sometimes also said Static and Dynamic Wavelength Assignment schemes [110].

2.7.6.1. Non-adaptive wavelength assignment

First-Fit and Random Wavelength Assignment schemes are well known and have been extensively studied in the context of wavelength routed networks [111]. In First-Fit the available wavelengths are labelled arbitrarily and listed in increasing order of label value. This order is static and equal to all the network nodes. When a node tries to send a burst, it sweeps the ordered list of wavelengths until it finds a free wavelength; if no wavelength is available then the burst is dropped. Random wavelength assignment

scheme tries to allocate a randomly selected wavelength from among the known free wavelengths. If there are no free wavelengths then the burst is dropped.

If the network does not perform wavelength conversion then the First-Fit wavelength assignment scheme performs worst than the Random wavelength assignment scheme because the probability of choosing any free wavelength is lower when the wavelength is randomly selected and so wavelength conflict in core nodes is less likely to succeed [109].

To minimize the poor performance of First-Fit Wavelength Assignment schemes, algorithms that use partitioning and assignment of start wavelengths have been used [109]. These algorithms are said to be traffic engineering approaches to the wavelength contention resolution problem as it assigns a different set of wavelengths to each network node or group of nodes and thus minimizes the overall combined interference level in the wavelength selection by the network nodes. The assignment of start wavelengths to different nodes also allows a better distribution of the effectively used wavelengths if the network traffic load is homogeneous in the network.

2.7.6.2. Adaptive wavelength assignment

In Adaptive Wavelength Assignment schemes, each node adjusts the wavelength selection accordingly to the received network feedback. A common mechanism to implement Adaptive Wavelengths Assignment is to assign a priority level to each wavelength. At any given instant the priority of the wavelength reflects the likelihood that the burst transmission on this wavelength will be successful and these priorities are updated periodically with the feedback from the network. A simple mechanism to manage the priority of a wavelength set is to increase the priority of a wavelength when it transmits successfully a burst and inversely to decrease its priority if the burst was dropped in the network. As all the nodes in the OBS network use the same algorithm, the priorities of different wavelengths are dynamically adjusted when the network operates.

Simpler Adaptive Wavelength Assignment schemes are related to local node usage. In this class one can find the Least Used and the Most Used wavelength selection

schemes. In these schemes the wavelengths are graded according to their frequency of use. If the Least Used scheme is adopted, the selected wavelength will be the one showing least usage and inversely, if the Most Used scheme is adopted the selected wavelength is the one that counts more transmissions.

In [112] Xu, *et al.* propose a number of adaptive algorithms to handle wavelength assignment with or without FDLs, such as Min-SV and Batching FDL. These authors also assess the computational complexity of the most common algorithms, comparing the performance of their proposals with Latest Available Unused Channel with Void Filling as presented by Xiong *et al.* in [31].

2.8. IP over WDM

The motivation to transport IP packets over wavelength division multiplexing (WDM) networks is the simplification of the transport and data layer as shown in Figure 20. While conventional copper cables need signal regeneration at each kilometre when operating at the bandwidth of 100 Mb/s, WDM technology allows a number of wavelength channels (currently 400 channels are under research, although there are reports of experiments with 1000 channels) in a link at 10 Gb/s rates for several tens of kilometres [113]. Liu summarizes the advantages of IP over WDM transport scheme as follows [113]:

- WDM networks can support the growth of Internet traffic by exploiting existing fibre infrastructure;
- The majority of data traffic across networks is IP. The future trend of data traffic still will be IP, *e.g.* Voice over IP (VoIP), P2P applications, etc.;
- IP over WDM (IPoWDM) inherits the flexibility, security and versatility of the IP protocol stack;
- IP over WDM may achieve on demand dynamic bandwidth allocation;
- IP over WDM guarantees vendor service and equipment interoperability via the IP protocols;

- The experiences of IP and ATM integration show that IP and WDM allow a closer integration for efficiency and flexibility.

Comparison of operational expenditure costs (OPEX) and capital expenditure costs (CAPEX) for several transportation scenarios including IPoWDM (or IPoDWDM for IP over Dense Wavelength Division Multiplexing) was performed by several authors, in particular, Batchellor and Gershel in [114]. These authors compared the CAPEX and OPEX for several transportation scenarios and for the national European network and the US Core network (see [114]), namely the Transponder Based Optical Layer (TXP), Next Gen Cross Connects with Packet over SONET (POS) with 10 Gigabit Ethernet interfaces (NGXC/POS and NGXN/10GE respectively) and IP over Dense Wavelength Division Multiplexing (IPoDWDM) with 10 Gb/s wavelength bandwidth and concluded that IPoDWDM presents significant advantages over all other transport scenarios, mainly because of the elimination of the electronic devices from the network data path, *i.e.*, the elimination of transponders, regenerators and electronic switch matrices.

The LOBS architecture proposed by [65] and discussed in section 2.4.3 is an implementation of the IP over the WDM transport principle. Chapter 6 presents the proposal of an approach for an OBS network using the IP over WDM approach.

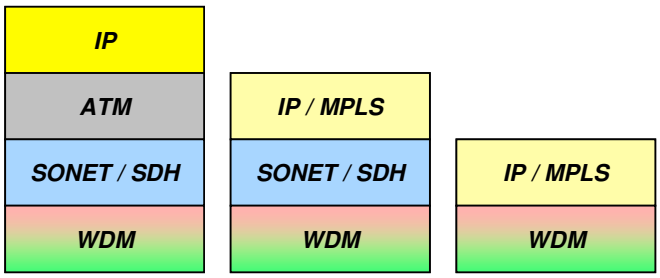


Figure 20 – Approaches for IP over WDM.

2.9. IP over OBS

Recently Farahmand *et al.* [115] proposed IP over OBS. Figure 21 depicts this architecture and compares it with the OSI model [116]. This approach assumes that

burst assembly (Packet Aggregation and De-aggregation - PAD) is performed at network level as depicted in Figure 21. At the Data Link layer, the Burst Framing Control (BFC) and the Medium Access Control (MAC) correspond to the Data Link layer. Also visible in this figure is the proposed model for the control plane. Here, the application layer corresponds to Burst Signalling Control (BSC), while Signalling Connection Control (SCC) is performed at the network layer. The Signalling Frame Control (SFC) is done at the Data Link level.

The BFC receives the aggregated packets from the upper layer and frames them suitably as shown in Figure 22. Figure 23 and Figure 24 show a generic control packet structure and a burst header packet structure, including QoS and routing information. It is not clear in [115] what the authors mean by the O&M acronym, other than this field will carry information about network management and signalling information.

The IP-over-OBS approach is much inline with the initial burst switching approach as stated by Haselton [32], in the sense that it distinguishes the "header" from the "packet payload".

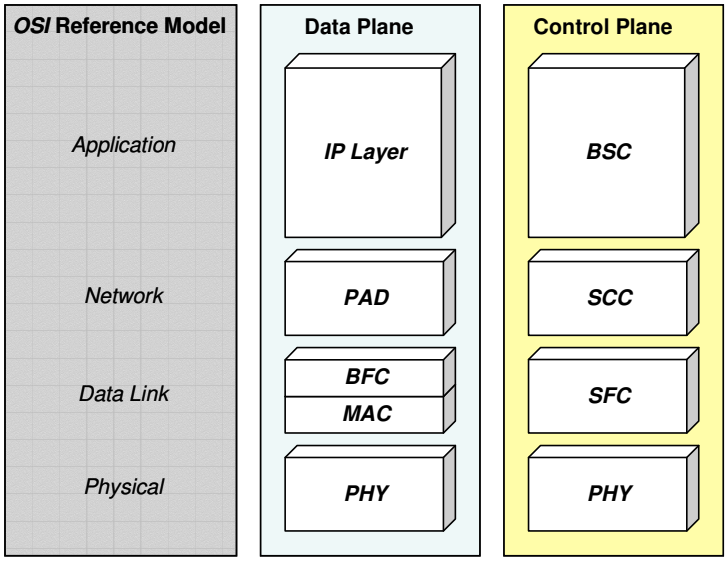


Figure 21 - IP-over-OBS proposed hierarchical model [115].

Figure 22 - IP-over-OBS proposed data burst structure [115].

Figure 23 - IP-over-OBS proposed control packet generic structure [115].

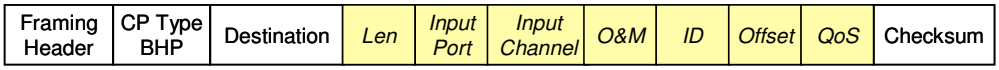


Figure 24 - IP-over-OBS proposed Burst Header Packet structure [115].

2.10. TCP over OBS

Transmission Control Protocol (TCP) [117] is the most frequent protocol in Internet communications, accounting for more than 90% of the transmitted IP packets [118]. The major TCP implementations (TCP Tahoe, TCP Reno and TCP SACK [119]) assume that the underlying transport medium is electronic and that the TCP packets can be buffered for as long as it is needed in order to assure its transmission by the electronic routers along the path of the TCP flow. OBS network transmission can not hold these assumptions; therefore OBS must support TCP in a manner that the performance of TCP based applications does not degrade.

Performance for TCP in OBS was assessed in [120]. In this paper, Gowda *et al.* state that the TCP performance degradation was severe for OBS networks operating at burst loss ratios as low as 0.3%. The problems that OBS poses to TCP can be summarized in two major aspects: the first one relates to burst loss, and the second one relates to the longer delay posed to packets in assembly queue and in transmission which can cause a TCP problem known as false time out (FTO) [121]. Authors in [121]

propose a new TCP implementation called Burst TCP (BTCP) that can detect FTOs occurring in the OBS network, thus improving TCP over OBS performance.

2.11. Summary

This chapter presents a detailed outline of state of art on optical burst switching. To allow a complete overview, the burst switching proposal as initially defined for ATM was presented and the burst assembly algorithms are briefly discussed from a generic point of view. The Optical Burst Switching concept is presented along with its main architectural approaches. Burst assembly of IP traffic if discussed, presenting the main burst assembly algorithms. OBS resource reservation protocols are presented and the main strategies for contention resolution in OBS. The chapter ends with the presentation of recent proposals for IP over WDM, IP over OBS and TCP over OBS.

Chapter 3.

Routing Algorithms for Optical Burst Switched Networks

3.1. Introduction

This chapter describes the main routing algorithms and strategies for Optical Burst Switched networks. It also presents three new approaches for the definition of paths in OBS networks – a new static shortest path routing algorithm termed Extended Dijkstra Routing Algorithm [13], a new complementary dynamic routing algorithm called Next Available Neighbour [14, 15] and briefly presents the C^3-OBS route definition management algorithm, called Travel Agency Algorithm, both discussed in Chapter 6.

The performance of the new algorithms for several network topologies is also presented and compared with the performance of strict shortest path routing algorithms, in particular, with the Dijkstra algorithm. Sections 3 and 4 present the main static and dynamic routing strategies for OBS, namely, shortest path routing, fixed alternate routing, and deflection routing. This chapter ends with the relevant conclusions.

This chapter is partially based on papers [13-15].

3.2. Main routing algorithms and strategies

Routing is defined as the process of establishing routes, including the task of moving data packets over the routes across a network from a source point to a destination point. This usually involves two distinct phases: the first is to find the optimal routing paths, taking into account a set of rules and constraints; and the second

is the send of the data packets through the network via the previously established path [122]. The critical phase of routing is the selection of the optimal path, due to the complexity and large dimension of network topologies and to the very often large number of rules and constraints that have to be met.

Routing is one of the most critical issues in the performance of OBS networks due to two specific reasons:

1. OBS uses source routing and

2. OBS routing is traffic unaware (also termed source blind routing), whether this is shortest path, fixed-alternate, deflected and so on.

In source routing, the path is defined at the entry nodes and thus the core nodes are not allowed to change the pre-established path. Traffic unaware routing is also a limitation to OBS networks since nodes typically do not perform traffic load prediction and thus are unable to avoid congestion points. Furthermore, even when nodes perform some sort of traffic prediction, nodes are still unaware of the network overall condition. With these limitations, routing and wavelength assignment (RWA) are critical components for the performance of OBS networks.

3.3. Static and dynamic routing

A route is said static if it stays unchanged during the transmission of the routed packet, or from a wider point of view, if the route stays unchanged during the transmission of the routed packets that have the same source and destination addresses. This is the case of OBS networks, as during the transmission of a burst in an OBS network, it is impracticable for a core node to redefine a new route for the burst. The redefinition of a route by a core node would normally require the addition of extra time to the T_{Offset} in order to allow the nodes in the new path (possibly longer than the initial one) to make the necessary resource reservations. When a burst is set to cross an OBS network, its path is predefined in the ingress node and the corresponding CP carries the path information. Using the network topology depicted in Figure 25, consider the case

where a burst enters the network at node 3 and must exit at node 6. As shortest path routing is used, the selected route will be 3 ➔ 5 ➔ 6 (meaning node 3, then node 5, finally node 6). T_{Offset} will be calculated considering that there are 3 hops in the path. Let us suppose that when the CP reaches node 5, there are no available output resources, while, in the other hand, resources to output the burst to node 4 are available. Deflecting the burst to node 4 would allow its successful transmission, but this deflection means that an extra hop was added to the path and thus, T_{Offset} would have needed to include an additional T_{Setup} to allow node 6 to make the necessary resource reservations, as the new path now would be 3 ➔ 5 ➔ 4 ➔ 6. If node 5 could delay (buffer) the burst long enough as to insert the extra time in the CP-burst gap, then the burst could possibly reach its destination. In OBS networks, such a delay means the use of Fibre Delay Lines (FDLs) and these pose a number of issues, namely because of two different reasons: the delay they introduce is not adjustable, *i.e.,* a set of bits is delayed exactly the amount of time they require to traverse the length of the FDL and also, the FDL might not be long enough as to delay the whole burst. Also, the excessive use of FDLs may introduce errors in the signal and thus the network would have to monitor the overall use of FDLs for each particular burst.

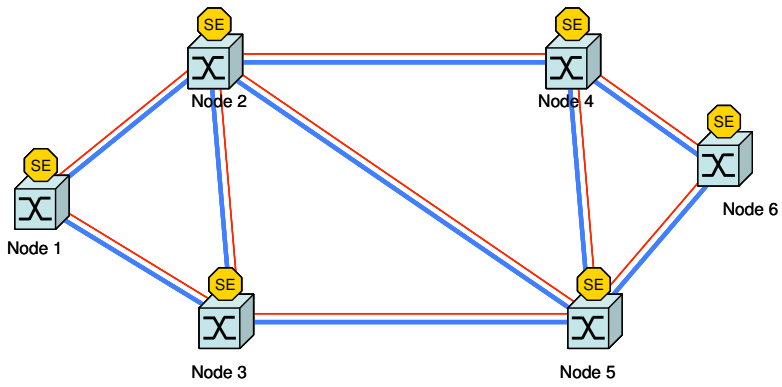


Figure 25 – Schematic representation of an OBS network with 6 nodes and 9 links.

Opposed to static routing is the generically termed dynamic routing, used broadly in packet switched networks. Dynamic packet routing is achieved because in electronic networks packets can be buffered and delayed while the new route is computed and its availability confirmed. There are alternatives to static routing in OBS networks and some of them are presented forward in this section, namely, Deflection

Routing [123], Next Available Neighbour (NAN) Routing [14] and to some extent the Travel Agency Algorithm (TAA) [26]. The later two can be cumulatively used in the C^3-OBS architecture proposed in Chapter 6.

3.3.1. Static routing

As defined before, in an OBS network, static routing happens when the burst path stays unchanged during the burst transmission. Examples of static routing for OBS networks are now presented and discussed their performance.

3.3.1.1. Shortest path

Routing in OBS has always assumed, either explicitly or implicitly, shortest path routes. Shortest path is a very popular routing scheme in packet networks since it guarantees minimum delay and minimum resource utilization. The main problem with shortest path is that it is load unaware and thus, if an overloaded link belongs to a shortest path, it will still be used despite of the possible existence of an available alternative route.

3.3.1.2. Dijkstra algorithm

One of the most popular shortest paths algorithms is the Dijkstra algorithm (DA) [100-102]. DA is an iterative algorithm that given a set of connected nodes and its interconnecting links will return the first available shortest path between any two nodes. Dijkstra algorithm can be described as follows [124]:

> "(…) The input of the algorithm consists of a weighted directed graph G and a source vertex s in G. We will denote V the set of all vertices in the graph G. Each edge of the graph is an ordered pair of vertices (u,v) representing a connection from vertex u to vertex v. The set of all edges is denoted E. Weights of edges are given by a weight function w: $E \rightarrow [0, \infty]$; therefore $w(u,v)$ is the non-negative cost of moving from vertex u to vertex v. The cost of an edge can be thought of as (a generalization of) the distance between those two vertices. The

cost of a path between two vertices is the sum of costs of the edges in that path. For a given pair of vertices *s* and *t* in *V*, the algorithm finds the path from *s* to *t* with lowest cost (*i.e.* the shortest path). It can also be used for finding costs of shortest paths from a single vertex *s* to all other vertices in the graph.

The algorithm works by keeping for each vertex *v* the cost *d[v]* of the shortest path found so far between *s* and *v*. Initially, this value is *0* for the source vertex *s* (*d[s]=0*) and *infinity* for all other vertices, representing the fact that we do not know any path leading to those vertices ($d[v]=\infty$ for every *v* in *V*, except *s*). When the algorithm finishes, *d[v]* will be the cost of the shortest path from *s* to *v* - or *infinity*, if no such path exists. The basic operation of the Dijkstra algorithm is edge relaxation: if there is an edge from *u* to *v*, then the shortest known path from *s* to *u* (*d[u]*) can be extended to a path from *s* to *v* by adding edge *(u,v)* at the end. This path will have length *d[u]+w(u,v)*. If this is less than the current *d[v]*, we can replace the current value of *d[v]* with the new value.

Edge relaxation is applied until all values *d[v]* represent the cost of the shortest path from *s* to *v*. The algorithm is organized so that each edge *(u,v)* is relaxed only once, when *d[u]* has reached its final value.

The algorithm maintains two sets of vertices *S* and *Q*. Set *S* contains all vertices for which we know that the value *d[v]* is already the cost of the shortest path and set *Q* contains all other vertices. Set *S* starts empty and in each step one vertex is moved from *Q* to *S*. This vertex is chosen as the vertex with lowest value of *d[u]*. When a vertex *u* is moved to *S*, the algorithm relaxes every outgoing edge *(u,v)*. (…)" (*in* [124])

The Dijkstra algorithm was designed to solve the single-source shortest path problem for a directed graph with non-negative weights. Real telecommunication

networks fall in this class of graphs, although very often each link is bidirectional and, as so, it must be considered as a pair of opposite direction graph edges.

Open Shortest Path First (OSPF) [125] ([126] for IPv6) is a well known real-world implementation of the Dijkstra algorithm used in network routing. OSPF is a link-state, hierarchical Interior Gateway Protocol (IGP) routing protocol, that uses *cost* as its routing metric. A link state database is constructed following the network topology which is identical on all routers for that network domain. In real networks, the Spanning-Tree Protocol (STP) [127] runs on the network before the OSPF. In a general way, a spanning tree of a graph is a sub-graph which is also a tree that contains all the nodes. In other words, in a network environment, where redundant links are common, the STP causes these links to appear closed for the operation of the network elements, as to eliminate the existence of loops and of duplicate messages, such as *e.g.* neighbour discovery messages.

3.3.1.3. Fixed Alternate routing

In fixed alternate routing (or simply alternate routing as some authors refer to it), each source-destination pair is connected through a finite number of possible burst transit paths [128]. Of course, among these, only one is elected as the shortest path although there may be several candidates. Fixed alternate routing algorithms select one of these paths each time a burst has to be sent from the source to the destination nodes. The selection of one of these paths is made randomly or sequentially, following a determined label order and is subject to the availability of the desired output port in the ingress node. Typically, the size of the set of paths associated with a particular source-destination pair is a function of the connectivity of the network, *i.e.*, the more connected the network is, the larger this set may be. One of the advantages of this algorithm is that it increases the probability of finding a free path for the burst transit. Its drawback is that, while selecting a longer path, the burst utilized bandwidth is bigger and thus may introduce additional burst drops in the network overall operation.

3.3.2. Extended Dijkstra routing algorithm

The main problem with fixed and alternate routing, *e.g.* the Dijkstra Algorithm (DA), is that these schemes are link load unaware, *i.e.*, they are not adaptable to changes in the network load caused by the variability of traffic. On other hand, apart from this limitation, iterative algorithms such as the ones based on the Dijkstra Algorithm tend to overload some links more than others in a specific class of topologies that offer at least two candidates for the shortest path, such as bidirectional ring topologies with an even number of nodes.

Bidirectional rings are a particular interesting class of topologies in optical networks, because in a ring there are two possible paths to the destination node instead of one. Rings are common sub-topologies in existing or planned networks, such as GEANT [1, 129], the COST 239 / European Optical network (EON) topology [130] or the USA backbone network NSFnet. Figure 26 and Figure 27 show the 11 node version for EON and the 14 node NSFnet networks – in both figures, among others, several four node ring sub networks, can be detected, *e.g.* Paris, Brussels, Luxembourg, Zurich for EON and Pittsburgh, Princeton, College Park and Ithaca for NSFnet.

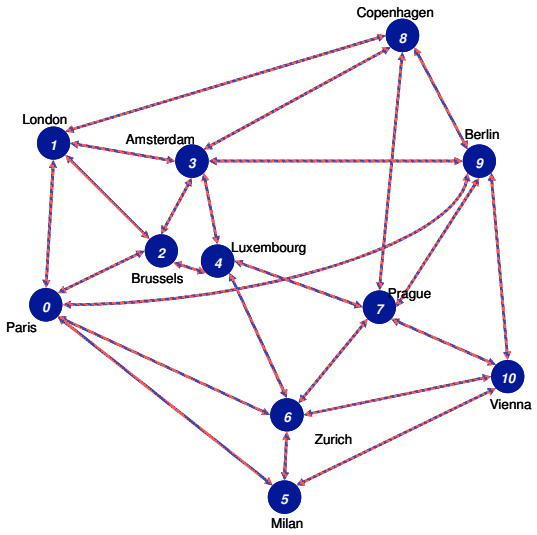


Figure 26 – The 11 nodes COST 239 / EON network.

Due to the iterative nature of the Dijkstra Algorithm, in topologies that contain rings the algorithm tends to place more paths over some links than over others. To better illustrate the nature of the problem, the term path will be used to refer to a set of links and nodes over which packets travel between their ingress and egress nodes in the network and the term route to the event of a packet using a link in such a path. Within this context, a packet follows a path but it only creates a route event on one link of the path at a time. A link contains as many route events as many different paths use that link.

In real world networks, path and load balancing algorithms tend to utilize efficiently all the available connection links, *e.g.* using Equal Cost Multi Path (ECMP) algorithms [125, 131]. But in simulators such as *ns-2* [103], very often the Dijkstra algorithm is the only available routing tool.

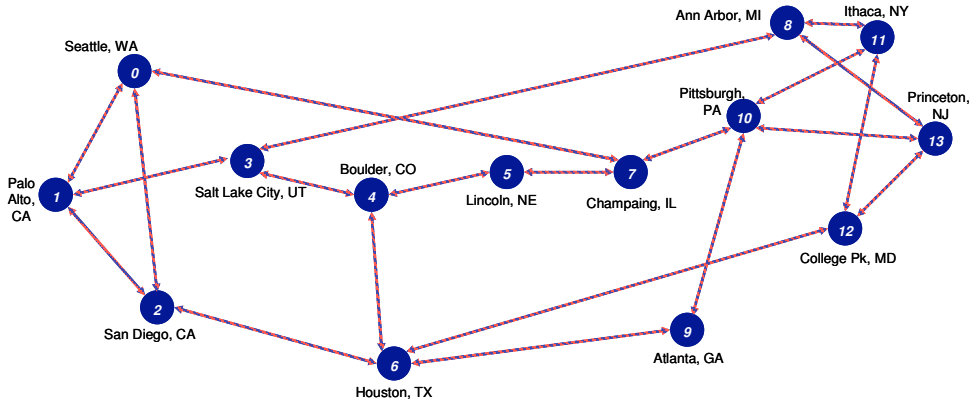


Figure 27 – The 14 nodes NSFnet network topology.

To assess the link overload problem verified in DA and the efficiency of the Extended Dijkstra algorithm, presented in this section, two routing algorithm performance metrics are defined. Consider the scenario for a ring network where each node offers equal traffic load with a uniform distribution for the destination, *i.e.* every communication between any two links is equally probable. In this homogeneous traffic generation scenario, each node sends a packet to another node following a uniform distribution. In a homogeneous traffic generation scenario network, the probability of a packet having a given source-destination pair, when the source node is fixed to $\boldsymbol{s}$, is

$$P(S = s \mid D = d) = \frac{1}{n-1} \tag{10}$$

where ***S*** and ***D*** are the source and destination variables, *s* and *d* are the identifiers of the source and destination nodes and *n-1* is the number of possible destinations, being *n* the total number of nodes.

Following the uniform distribution for source and destination nodes and considering

$$\begin{aligned} & N \overset{def}{=} \{1..n\} \\ & Path(s,d) \overset{def}{=} \{(s,h_1,h_2,...,h_{PL-2},d) : s \in N, \\ & \qquad h_1 \in N \setminus s, h_2 \in N \setminus \{s,h_1\},..., \\ & \qquad d \in N \setminus \{s,h_1,h_2,...,h_{PL-2}\}\} \end{aligned} \tag{11}$$

where ***PL*** is the path length in terms of its number of hops or the number of elements minus one of the ***PL***-tuple Path(***s***,***d***), for the whole network, the number of paths created is

$$\#Paths = 2.C_2^n \tag{12}$$

where

$$Paths = \bigcup_{\substack{s \in N \\ d \in N \setminus s}} Path(s,d) \tag{13}$$

and

$$C_2^n = \frac{n!}{(n-2)! \times 2!}, \tag{14}$$

and where *s* and *d* are elements of the sets of possible source and destination nodes, identified by *1* up to *n* and *C* is the number of combinations of *n* taking two by two. The combination is counted twice because of the bidirectional nature of the communication links. These paths will be distributed over the links that physically implement the

connections between the nodes, so the number of available paths is related to the load on the links and the equilibrium of the communication in the network, considering the homogeneous traffic generation scenario.

A bidirectional link l is defined as a pair (n_i,n_j) where n_i and n_j are two directly connected nodes. Note that $l=(n_i,n_j)=(n_j,n_i)$, although its traffics may be different, *i.e.*,

$$R(n_i,n_j)=\vec{R}_l;\;\; R(n_j,n_i)=\overleftarrow{R}_l \tag{15}$$

where R is the number of routes between nodes n_i and n_j, as defined forward in (16).

Furthermore, L is defined as the set of all links l_i and $\#L$ is its dimension.

If the topology is fully connected, the number of routes that use any given link l in the network, is

$$R_l=\frac{\sum_{i\in L} R_i}{\#L}=2\,, \tag{16}$$

because of the bidirectional nature of the link. If the network is not fully directly connected, *i.e.* if some links of the fully directly connected network are not present, for each link l' (connecting node i to node j) that is missing, its routes $R(n_i,n_j)$ and $R(n_j,n_i)$ will be reassigned to another route over the existing links, introducing asymmetry and heterogeneity in the traffic on the network. The creation of paths over a network topology, *i.e.* the assignment of these routes over network links, performed by an iterative algorithm such as the Dijkstra Algorithm, will always prefer the first available link in the selection list, disregarding concerns of overload and unbalance for that link.

Having this scenario in mind, two additional metrics are defined, balance and symmetry, to allow the assessment of the efficiency of a routing algorithm concerning the evenly distribution of the number of routes and paths over the available links.

Let us define a balanced routing algorithm as one that respects the following condition: for any link l, the sum of number of routes that use l is equal or, at least, very close to the number of routes created for any other link, *i.e.*

$$\vec{R}_l + \overleftarrow{R}_l \approx \frac{\sum_{i \in L} R_i}{\#L}, \forall l \in L \Leftrightarrow \sigma(R_i) < \varepsilon, i \in L, \quad (17)$$

where L is the set of all links and $\#L$ its dimension, σ is the standard deviation for the values of R_i ($i \in L$), ε is a function of the number of routes, desirably small. In topologies where some nodes are more connected than others, the value of ε may be set to some adequate value, because the more heterogeneous the nodal degree of the nodes in the topology, the more difficult will be the choice of balanced paths for all the links.

A symmetric routing algorithm is defined as one that, for any given bidirectional link l, creates routes in a way that the number of routes that use that link in one direction ($\vec{R}_l$) is equal or close to the number of routes that use the same link in the opposite direction ($\overleftarrow{R}_l$), *i.e.*,

$$\vec{R}_l \approx \overleftarrow{R}_l, \forall l \in L, \quad (18)$$

being the desired proximity of these two values dependent also of the homogeneity of the nodal degree of the nodes in the topology.

Cumulatively, a balanced symmetric routing algorithm is defined as

$$\vec{R}_l \approx \overleftarrow{R}_l \approx \frac{\sum_{i \in L} R_i}{2.\#L}, \forall l \in L. \quad (19)$$

The smallest ring topology that is not fully meshed is the 4 node ring network topology, as shown in Figure 28. The number of routes that use the links on a four node ring network following two different implementations of a shortest path Dijkstra like algorithm are depicted in Figure 29 a) and b). Although in this figure the numbering sequence of the nodes was kept (from 1 to 4, clockwise), their order is not relevant for the results, as changing its sequence results in a permutation of the routes on the links.

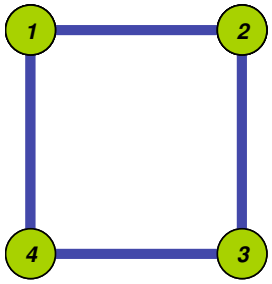


Figure 28 – Four nodes ring network.

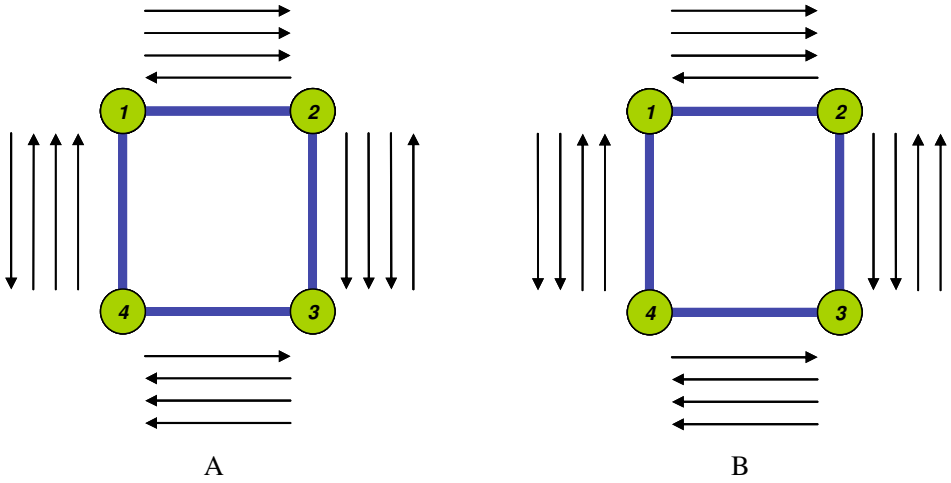


Figure 29 – Dijkstra algorithm routes (version A and version B) in a four nodes ring network.

Having defined this simple network, the DA was run to calculate the number of paths that are created having in consideration all source destination pairs. The result routing table is shown in Table 2. Depicting the paths over the graph of Figure 28, the overload the DA poses on links in network topologies such as rings is clearly visible. As can be seen from Table 2, each of the 12 possible source-destination pairs on a 4 node ring network originates a route. The incidence of these routes over each link is depicted in Figure 29 A. Here each arrow represents a created route that uses this particular link. For some links, there is three times more traffic in one direction than in the other. This is clearly an unwanted scenario as it means that traffic is not homogeneous, although it might be balanced (no bidirectional link has more traffic events than the other). Some implementations of the DA are not as unbalanced. Figure 29 B depicts such case,

although some links still show three times more traffic events on one direction than on the other.

As can be seen in Figure 29 A and B, none of the route load graphs is symmetric. In Figure 29 A, each link is assigned with 300% more traffic requests in one direction than in the other. This is also visible for Figure 29 B for links 1-2 and 3-4. Links 1-4 and 2-3 in Figure 29 B are balanced and symmetric.

Table 2 – Routing table for a ring network with four nodes.

Source	Destination	Path
1	2	1 → 2
1	3	1 → 2 → 3
1	4	1 → 4
2	1	2 → 1
2	3	2 → 3
2	4	2 → 3 → 4
3	1	3 → 4 → 1
3	2	3 → 2
3	4	3 → 4
4	1	4 → 1
4	2	4 → 1 → 2
4	3	4 → 3

The difference in the symmetry has higher impact in ring networks that have an even number of nodes, with the maximum asymmetry occurring for the 4-node ring network. The reason for this fact is that in a network with an even number of nodes, there will be two shortest paths (with equal cost) between two diametrically opposed nodes, *e.g.*, in the network depicted in Figure 29, between nodes *1* and *3* and nodes *2* and *4*. Likewise, if the network has more than four nodes, the routes that overload the links constitute a smaller share of the overall routes created and thus the asymmetry is

higher in a 4-node ring network. If the number of nodes is odd and the links have equal costs, there will be no two equal cost possible paths between any two network nodes.

Figure 30 shows a network with seven nodes and nine bidirectional links. This network is actually composed of three rings with four nodes sharing some of their links. The Dijkstra algorithm was applied to this topology and observed the number of routes that are assigned to each link. The result is expressed in a route matrix, as shown in Table 3. A route matrix is a bi-dimensional matrix that contains the number of routes created in each link connecting any two nodes. As the route matrix in Table 3 is almost symmetric, one concludes that the algorithm produces quasi-symmetric routes. Yet, it can be seen that link *1-2* is used in 7 paths, while link *5-6* is used in only 2 paths.

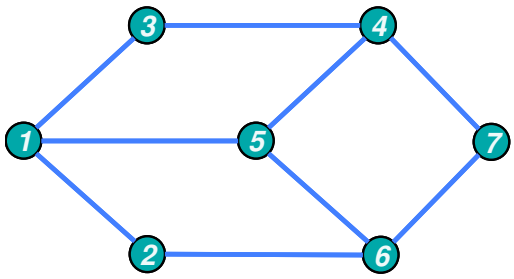


Figure 30 – Seven nodes network with nine bidirectional links.

The performance assessment of an OBS network with this topology, using a simulator that implements the DA, would result in a higher burst loss probability ratio for the nodes whose output links have received more routes. Similar conclusions may be drawn for any other packet or burst switched network that does not enforce tree-shaping algorithms to its topology.

For this topology, the effect of the iterative nature of the algorithm is well observed on the Sum line and column of the route matrix in Table 3 – the links that connect the lower designation nodes show almost three times more route events than the lightest loaded links.

Table 3 – Route Matrix for the seven node nine links topology for Dijkstra.

S\D	1	2	3	4	5	6	7	Sum
1	-	6	7		3			16
2	7	-				4		11
3	6		-	5				11
4			4	-	3		4	11
5	3			3	-	2		8
6		5			2	-	2	9
7				3		3	-	6
Sum	16	11	11	11	8	9	6	

S\D: Source \ Destination.

To define the Dijkstra algorithm, the graph $G = (V, E)$, where V is a set of vertices and E is a set of edges will be used, S denoting the set of vertices whose shortest paths from the source have already been determined and V-S are the remaining vertices. The other data structures needed are: d - an array of best estimates of shortest path to each vertex and p_i - an array of predecessors for each vertex.

The basic mode of operation for DA is:

1. Initialize d and p_i,
2. Set S to empty,
3. While there are still vertices in $V-S$,
   i. Sort the vertices in $V-S$ according to the current best estimate of their distance from the source,
   ii. Add u, the closest vertex in $V-S$, to S,
   iii. Relax all the vertices still in $V-S$ connected to u.

The relaxation process (step 3.iii) updates the costs of all the vertices v, connected to a vertex u, if the best estimate of the shortest path to v by including (u,v) in the path to v can be improved. More complete descriptions of the Dijkstra algorithm are extensively available in the literature.

Having in mind the limitations of the Dijkstra algorithm on the creation of paths on graphs that include rings with an even number of vertexes, a new algorithm that aims to solve them is proposed. For that matter, it is introduced the following concept: in a graph, each node (or vertex) is identified by a unique number, termed the node

identifier, without any special order. The initial Dijkstra algorithm is extended in order to detect possible equal cost routes and use additional conditions based on the nodes identifiers to select on those routes. After the sorting phase (3.i), (*u* is already defined), the Extended Dijkstra algorithm continues as follows:

```
3.i.a. if there is another candidate u' to be the best
       vertex (that is, with equal cost), then
3.i.b. if the sum of the vertex identifiers related to u is
       equal to the sum of vertex identifiers related to u',
       then
3.i.b.1. choose u as the vertex that has the first lowest
       neighbour,
       else
3.i.b.2. if the source-node s of the path is lower than the
       destination-node v of the path, then
3.i.b.2.1. choose u as the vertex which path has the highest
       sum of node identifiers,
       else
3.i.b.2.2. choose u as the vertex which path has the lowest
       sum of node identifiers.
3.i.c. repeat 3.i.a. until there are no left candidates
```

As these additional conditions only run if there are two or more path candidates to shortest path, the Extended Dijkstra still provides strictly shortest paths. Appling the new algorithm to the test case of the four node ring network shown in Figure 29, one has new routes as shown in Figure 31. The routes shown in Figure 31 are balanced and symmetric.

The performance assessment of the new algorithm was performed with a modified version of the simulator previously build and published in [105]. As this simulator was developed for OBS networks, it allows to carry out performance studies of the Extended Dijkstra algorithm regarding the standard Dijkstra algorithm in terms of its impact on this kind of networks.

Two different metrics were obtained through simulation: number of routes created over each unidirectional link and number of bursts dropped in the network. All the network simulation parameters were kept equal, except for the selected routing

algorithm. The simulation parameters are presented in section 6.5. Simulated topologies were the four-node ring, six-node ring, eight-node ring, two topologies with seven and nine nodes made of adjacent four node rings and the eleven nodes COST 239 network. The route matrices for the eight-node ring and the eleven nodes COST 239 and the graphs for the seven nodes (three rings with four nodes) and nine nodes (four rings with four nodes) are presented as well as the performance comparison of OBS networks for several topologies at different load scenarios.

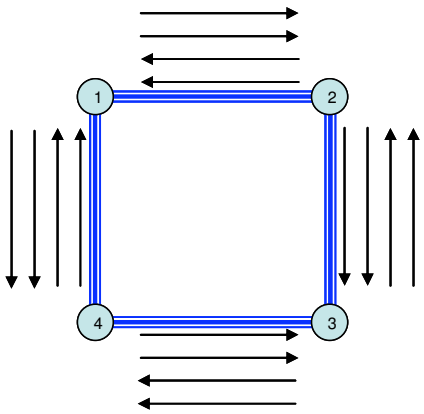


Figure 31 – Routes created by the Extended Dijkstra algorithm in a four node ring topology.

Table 4 – Route Matrix for the eight nodes ring topology for Dijkstra.

S\D	1	2	3	4	5	6	7	8
1	-	6						9
2	8	-	6					
3		8	-	6				
4			8	-	7			
5				9	-	8		
6					10	-	8	
7						10	-	8
8	7						10	-

Table 5 – Route Matrix for the eight nodes ring topology for Extended Dijkstra.

S\D	1	2	3	4	5	6	7	8
1	-	8						8
2	8	-	8					
3		8	-	8				
4			8	-	8			
5				8	-	8		
6					8	-	8	
7						8	-	8
8	8						8	-

Table 6 – Route Matrix for the eleven nodes COST 239 topology for Dijkstra.

S\D	1	2	3	4	5	6	7	8	9	10	11
1	-	2	1	1	1	2					5
2	2	-	1				3		4		
3	1	1	-	2	3		3				
4	1		2	-		1	4			4	
5	1		3		-	3	4	5			
6	2			1	3	-		2		2	
7		3	3	4	4		-		3		5
8					5	2		-	4	3	4
9		4					3	4	-		5
10				4		2		3		-	7
11	5						5	4	5	7	-

Table 7 – Route Matrix for the eleven nodes COST 239 topology for Extended Dijkstra.

S\D	1	2	3	4	5	6	7	8	9	10	11
1	-	6	3	3	2	3					5
2	4	-	2				3		4		
3	1	1	-	2	3		3				
4	4		2	-		2	4			4	
5	4		2		-	2	3	5			
6	3			2	3	-		2		2	
7		2	1	4	4		-		3		4
8					4	3		-	5	2	2
9		4					3	3	-		4
10				5		3		3		-	3
11	6						2	3	2	5	-

The following figures, from Figure 32 to Figure 35, show the simulation results, with the number of created routes near the links.

Figure 36 shows the improvement due to the use of the Extended Dijkstra algorithm instead of the Dijkstra algorithm for three burst loss scenarios, around 50%, around 10% and around 1% for the following networks: four-node ring, six-node ring, eight-node ring and nine-node topology (four rings with four-node each) networks and the COST 239 topology. In all the scenarios, the Extended Dijkstra shows improvement in the burst loss probability. As expected, more loaded networks show a smaller improvement since it is well known that overloaded OBS networks do not have room for additional improvement [23, 30, 132]. Also expectedly, more connected networks show higher improvement ratios than low connected networks for similar network simulation parameters. The value for burst loss probability using Extended Dijkstra when the burst drop probability was around 1% (using DA) for the 9 nodes (8 nodes ring plus central node) is not shown because it is very close to zero.

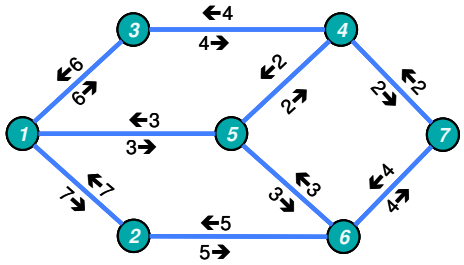


Figure 32 – Number of created routes per link and direction in a seven nodes topology with 3 four-node ring networks sharing links obtained with the Dijkstra algorithm.

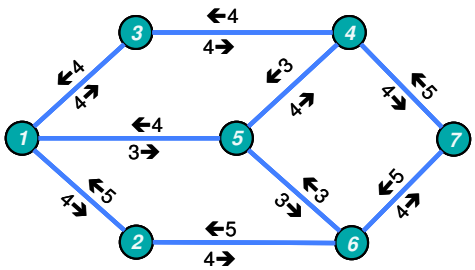


Figure 33 – Number of created routes per link and direction in a seven nodes topology with 3 four-node ring networks sharing links obtained with the Extended Dijkstra algorithm.

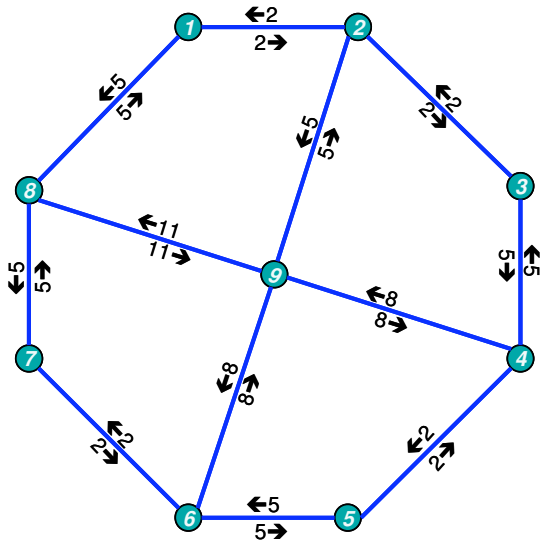


Figure 34 – Number of created routes per link and direction in a nine-node topology with 4 four-nodes ring networks sharing links obtained with the Dijkstra algorithm.

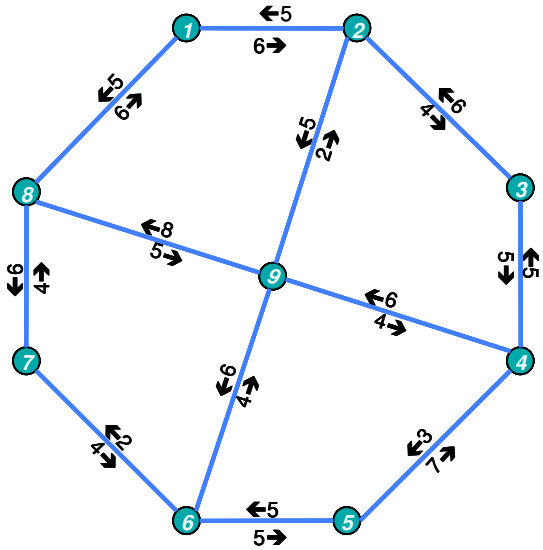


Figure 35 – Number of created routes per link and direction in a nine-node topology with 4 four-nodes ring networks sharing links obtained with the Extended Dijkstra algorithm.

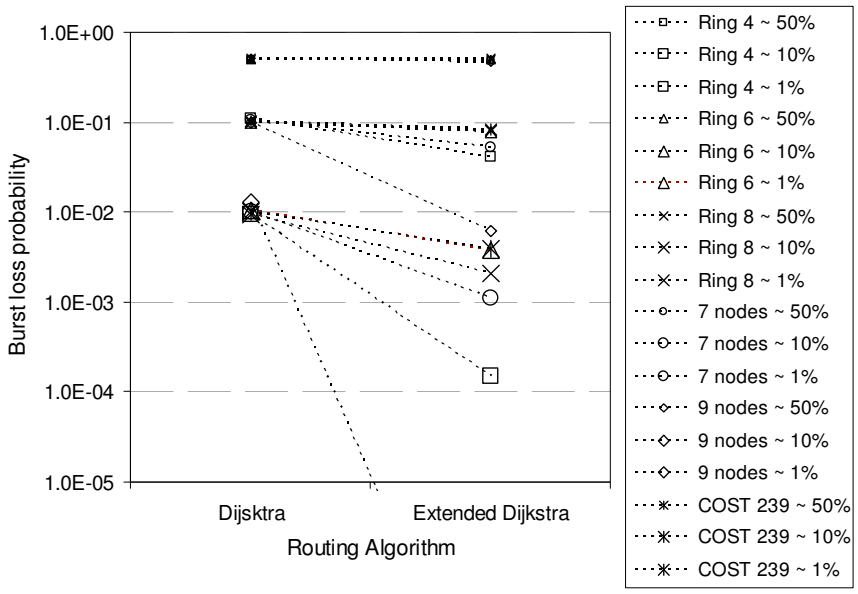


Figure 36 – Burst loss probability versus routing algorithm for three burst loss scenarios (around 50%, 10% and 1%) for four network ring and ring based topologies.

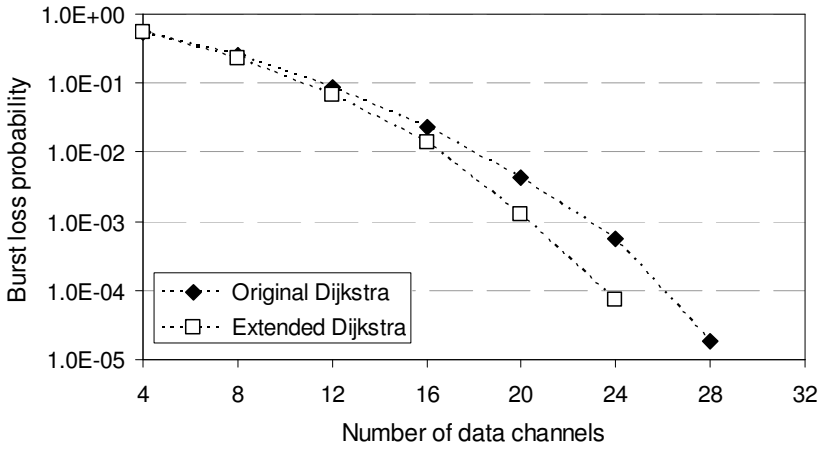


Figure 37 – Comparison of burst loss probability versus number of data channels for the eleven nodes COST 239 topology OBS network, using the Dijkstra and the Extended Dijkstra routing algorithms.

The burst loss probabilities obtained for the COST 239 network topology for the two routing algorithms are shown in Figure 37. As may be seen the change in the routing algorithm is responsible for an improvement of almost an order of magnitude when the network has 24 available data channels (traffic generation rate was kept constant to allow comparison). For 28 data channels, the burst loss probability for the case of routing performed with the Extended Dijkstra algorithm drops to zero.

To compare the balance of the Dijkstra and Extended Dijkstra routing algorithms, following the definition in (17), the standard deviation of the values for the route matrices was measured for all the topologies. Figure 38 shows that, for all the tested topologies, Extended Dijkstra routing produces more balanced routes on links than Dijkstra routing.

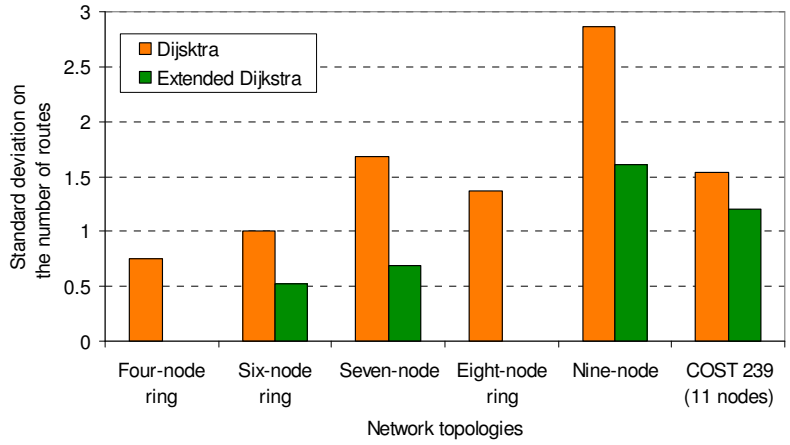


Figure 38 – Standard deviation on the number of routes created over links for the Dijkstra and the Extended Dijkstra routing algorithms, for several topologies.

The combined balance and symmetry of the networks for both routing algorithms, following the definition in (19) was also assessed. Figure 39 shows the differences between the two algorithms for the tested topologies. The two mean routes per link ($\bar{R}/\#Path$ and $\bar{R}/\#Path$, respectively) and the overall mean number of routes per link are shown. For regular topologies, as the four-node and eight-node rings, the Extended Dijkstra algorithm is clearly more symmetric than the Dijkstra algorithm; on the six-node ring both algorithms have a similar performance regarding symmetry. On

the other hand, for the seven-node and nine-node irregular topologies, the Extended Dijkstra algorithm is less symmetric than the Dijkstra algorithm, albeit in the case of the eleven nodes COST 239 topology the difference is negligible. The combination of these results with the results from Figure 36 and Figure 38, clearly show that the balance between all links is more important for the overall performance of the network than the link bidirectional symmetry.

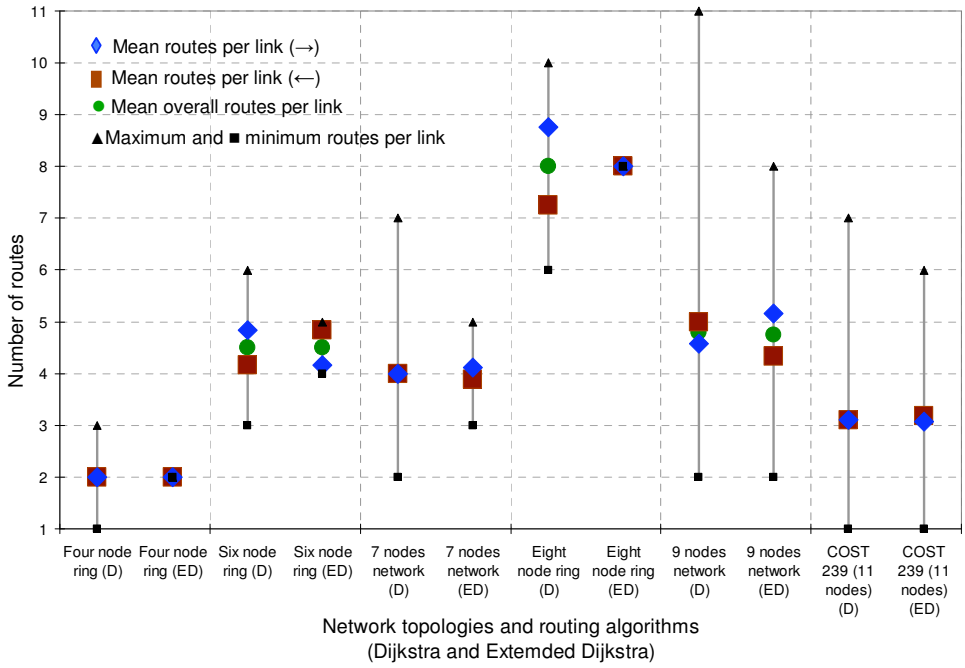


Figure 39 – Symmetry test for the Dijkstra (D) and Extended Dijkstra routing algorithms (ED), showing mean number of routes in each direction and overall mean and maximum and minimum number of routes per link for several topologies.

Although routing in a real environment cannot be restricted to OSPF (an implementation of the Dijkstra algorithm), it was proved that there is still room to improve the initial Dijkstra algorithm, being the Extended Dijkstra our proposal. The Extended Dijkstra algorithm was tested through simulation and its performance metrics, balance and symmetry where compared with the ones for the Dijkstra algorithm. Results

show that for the overall network performance, balance in links is more relevant than symmetry in routes.

Furthermore, the results presented in this section suggest that several published results focusing on performance assessment of ring networks or networks than include rings such as [84, 133], or ones that rely on simulators that use plain Dijkstra routing algorithm, such as the *ns-2* [103], the *OWns* [104], the *OBSim* [105] or the *OIRC OBS-ns Simulator* [106], could be over pessimistic because its static-source-routing shortest path only approach is not adequate when compared with the still shortest path provided by the Extended Dijkstra algorithm. In conclusion, the estimations made regarding the nature and amount of the network resources necessary to achieve a given level of performance on these networks are pessimistic.

Shortest path routing is not the sole mechanism that defines routing in modern networks. Other authors, such as [134, 135], propose heuristic and traffic engineering algorithms that seek to optimize the set of paths used to route static traffic demands, not necessarily returning shortest paths routes. Nevertheless, the starting point in network routing is very often OSPF. The Extended Dijkstra algorithm presented here, still provides strictly the shortest paths and thus may allow a faster convergence to routing table equilibrium in dynamically routed meshed networks.

3.3.3. The Travel Agency algorithm

The Travel Agency Algorithm (TAA) is the main algorithm that supports advanced routing in the C^3-OBS traffic aware network architecture. This algorithm is static from a burst point of view since the full path for the burst is completely predefined in the ingress node, and dynamic from the source-destination node pair point of view, since the path defined for a given source-destination pair may be different each time a burst transmission is requested. When returning the burst path this algorithm takes into account the foreseeable resource reservation status of the network for the predicted time of the burst transmission. This algorithm is described in detail in section 6.2.4.

3.4. Dynamic routing strategies

When two bursts try to use the same resource, *e.g.* a channel in a link, the switching node may perform several actions as to maximize the efficiency of the network. The simplest action is to drop the lowest priority burst, in order to allow the highest priority burst to successfully transmit. Other strategies include drop and retransmit the burst, perform deflection routing [123] or buffering [50] the burst. In the case of burst drop and retransmit, the switching node where the burst was dropped will ask for a retransmission of the dropped burst, by sending a negative *acknowledgement* to the ingress node of the dropped burst. Deflection routing is analysed below, as a mean of introduction of Next Available Neighbour Routing, a novel routing algorithm that combines some features of deflection routing and buffering.

3.4.1. Deflection Routing

In OBS, Deflection Routing [123] aims to use free channels as optical buffers in a resolution contention situation. In the context of electronic networks it was initially described as "Hot-Potato Heuristic Routine Doctrine" in 1964 by Paul Baran [136]. "Hot Potato" acts on messages as if they were hot potatoes and tries to pass them to the next node as fast as possible – ultimately the message will reach its destination because of the destination address tag it carries. The Deflection Routing algorithm starts with the same problem of "Hot Potato" – the lack of memory at the switching nodes (not enough memory, in the case of "Hot Potato"), although its approach is more elaborate, particularly in the case of OBS networks.

Figure 40 depicts an OBS network with Deflection Routing. When node 2 and node 3 decide to schedule a burst to node 6, link 5-6 may impose contention on one of the burst transmissions. If so, instead of dropping one of the bursts, node 5 may try to deflect one of the bursts to the node 4, which in turn will sent it to node 6.

Deflection Routing for OBS networks imposes an additional problem related with the offset time between the CP and the Burst. Deflection can only occur if the total number of hops (or total path length) of the originally planned route is bigger or equal

to the path cost of the deflected route, or if delaying techniques are applied to the burst as a mean to increase the offset between the CP and the burst after the burst has already entered the network, *e.g.* by using FDLs. In the example depicted in Figure 40, the CP for the burst originated in node 3 should have an additional time in order to allow the configuration time for an extra hop in the route.

According to the authors in [137, 138], Deflection Routing may be regarded as a viable solution for contention resolution in lightly loaded optical networks, *i.e.*, in a load scenario where some links (the links that will carry the deflected burst) still admit additional charge. Furthermore, the authors in [137, 138] claim that Deflection Routing can be used conjointly with other strategies, *e.g.* burst segmentation (see section 2.7.3).

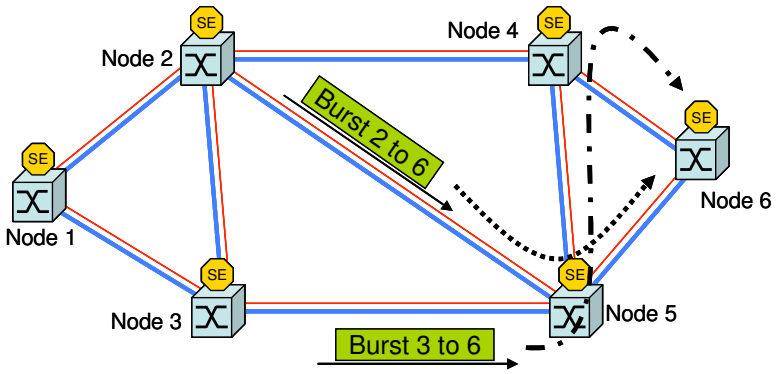


Figure 40 – Schematic representation of an OBS network with Deflection Routing (CPs are not depicted).

3.4.2. Next Available Neighbour (NAN) routing algorithm

The non-stopping need for information exchange leads to a continuous traffic increase and the fluctuation in traffic patterns calls for efficient transport and routing algorithms. The network bandwidth limitations are sometimes met by traffic spikes and when that happens, traffic is dropped. There are several techniques to selectively drop the smallest priority packets or that apply some degree of fairness to the sources that have their traffic dropped. In packet switched networks the traffic may be dropped in an

intermediate node, but in circuit switched or connection oriented networks, the traffic admission policies are often applied at the ingress node. Inside an OBS network, bursts are the smallest data unit, and thus when a burst is blocked, several data packets are lost.

It has been shown that longer paths show higher loss probabilities, *i.e.* the longer the number of hops the packet must traverse, the higher its drop probability is [139]. This problem is a consequence of the need to find, in a cumulative manner, the available network resources from ingress to egress nodes. In an overloaded network scenario, a packet or burst may not reach its final destination and thus be dropped, possibly being reinserted into the network at a later time.

Yet, if the nature of the problem that caused the packet or burst to be dropped was the excessively long or otherwise overloaded or damaged segment of the network, the packet/burst will still be dropped again until new routing is devised and applied. Additionally, in [140], the existence of nodes with a particular behaviour, termed occluded nodes was shown. These nodes act as burst droppers, causing unfairness in the burst transmission.

Having in mind these issues, namely, the penalty caused by long paths and the existence of occluded nodes, a new routing algorithm is proposed. This algorithm is such that in the event of a packet or burst drop, the node will instead forward the packet or burst towards another node as close as possible to the destination of the packet or burst. This algorithm is complementary to existing routing strategies as it is applicable in a burst / packet loss situation, and it is called Next Available Neighbour Routing (NAN routing) [14].

There are at least two possible embodiments to this algorithm:

1. IP / MPLS
2. OBS / C^3-OBS / OPS / OCS (Optical Burst Switching / Common Control Channel OBS / Optical Packet Switching / Circuit Switching).

For OBS, NAN routing acts as follows: when an OBS core node receives a CP and decides that it cannot guarantee the reservation of the resources the CP requests,

instead of sending a negative *acknowledgement* (NAK) CP back to the ingress node, it will manufacture a NAN-CP destined to a NAN node (detailed ahead); when the burst arrives at the node, instead of being dropped, it will be routed to a NAN node, or, eventually, if no NAN node is available and the node allows for this procedure, it will be O/E converted, buffered and later reinserted in the network.

In C^3-OBS, NAN routing acts as follows: if for a given burst, the resources are not fully available along the path of the burst, and will not be available within an acceptable time window, the burst will still be routed as far as possible. The destination node of the burst will be kept as the originally intended egress node, and the intermediate node that receives the burst will act as the neighbour node. In this case, when the node receives the burst, it must O/E convert it and reinsert it in the network at a later time.

For both OBS and C^3-OBS, the approach of splitting the path of the burst and relay it onto a node that had not initially been planned to handle it, brings benefits as it circumvents these two aforementioned known OBS problems: the penalty for long paths and the burst dropping at occluded nodes.

For IP Multi Protocol Label Switching (MPLS) networks, NAN routing works as follows: when following a given MPLS path, if a node finds the exit link unavailable or overloaded, instead of further buffering the packet or dropping it, it will relay it to a node outside the original MPLS path, acting this as the neighbour node. The neighbour node will, in turn, try to reroute the packet towards its final destination.

A counter or other method, *e.g.*, a method based on the generation time of the packet or burst is added to the NAN routing information in order to avoid endless re-insertions or loops in the packet / burst route in the network.

The neighbour node selection feature is performed by a function that takes into account several network resource status variables, such as (for example but not exclusively): fewer hops to the destination, biggest available bandwidth on the path to the final destination, or shorter predicted delivery time to the final destination. One possible alternative for this selection function is the random choice of the node which

will act as the neighbour node. In the simulations presented forward, the selection function was defined as purely random.

In NAN routing for IP networks, the IP packets are specifically addressed as usual and no manipulation is made on the addresses of the packets. If a node finds that the network resources needed to the next hop are unavailable, this node may decide to redirect the to-be-dropped burst / packet to any trustworthy node near to its final destination. It will be the responsibility of this neighbour node to reinsert the packet / burst into the network as to allow it to reach its final destination.

It should be noted that NAN routing is not IPv6 anycasting [10], since anycasting is performed on a given set of machines (namely routers, not hosts); also, anycasting is mostly non-deterministic, *i.e.*, two anycast requests may end up in different machines. NAN routing is also non-deterministic from a network point of view in the sense that the choice of the neighbour node may be performed circumstantially (*e.g.* per packet arrival), but may be deterministic from a node point of view, because the selection of the NAN node may be subject to a number of rules and constraints.

Also, NAN routing differs from IPv6 anycasting in the sense that the neighbours that receive the packets need not to belong to the same sub-network, *i.e.* the only requirement to NAN routing is that there has been a previous trust agreement to NAN route between the machines, being this trust agreement implicit (in the case the machines under the same administrative domain) or explicit (in the case the machines are managed by different entities), and in this later case, there will be a mechanism associated with the exchange of trust credentials, or more simply, a trust table that is securely exchanged among all nodes in a given neighbourhood.

Please note that the NAN routing algorithm may be cumulatively used with other routing strategies or algorithms, such as fixed-alternate routing or deflection routing, both described previously in sections 3.3.1.3 and 3.4.1.

Consider the following example for MPLS networks (see Figure 41): Node 1 is sending packets to node 6 using the Virtual Private LAN Service (VPLS) set up via the links 1-2, 2-4 and 4-6. Node 4 finds link 4-6 unavailable (*e.g.* overloaded or with a

failure status). Instead of dropping traffic or buffering it long enough until the link is up again, node 4 will use link 4-5 to send the MPLS traffic. Node 5 will then try to resend this traffic to node 6.

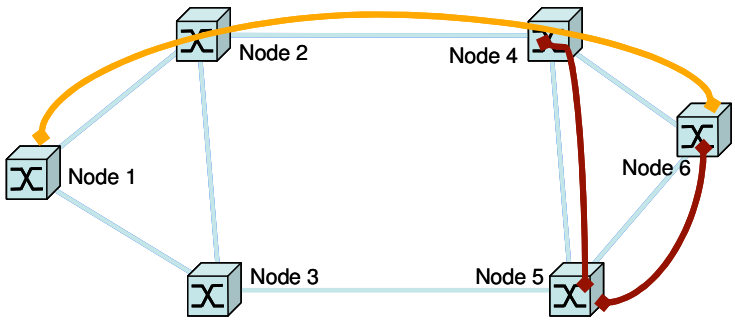


Figure 41 – Sample MPLS network topology with three sample virtual routes.

Note that in MPLS traffic, although the addresses in the packets are not manipulated, the MPLS address tags are. Upon reception, the egress node has the responsibility to check whether the packets final address is one of their sub-networks or not. If not, the egress node will try to resend these packets to their ultimate destination.

Presenting an example based on the network topology shown in Figure 25 and considering a traffic aware architecture (*e.g.* C^3-OBS): consider that node 1 sends a packet / burst to node 6 (see Figure 25), following path 1-2-4-6, but links 4-6 and 5-6 are busy. Node 4 is not overloaded and link 4-6 will be available in an acceptable time, but not links 1-2 and 2-4, which will become occupied. Given this situation, node 1 may decide to use neighbour node 4 while keeping the packet / burst addressed to node 6. As soon as the link 4-6 (or 4-5 and 5-6) becomes available, the "guest" packet / burst buffered in 4 can be resent to its egress destination.

Figure 42 depicts NAN routing for an OBS network when node 2 and node 3 schedule a burst to node 6, using concurrently the available data channel in link 5-6. In this case, node 5 decides to relay the burst to be dropped burst to node 4, which receives and buffers it electronically (MEM block in Figure 42). The buffered burst does not

need to be interpreted and may be reinserted into the network as soon as the resources are free (or are estimated to be free). Another implementation of NAN for OBS networks would be to use node 5 as the neighbour node, if the node is unable to route the burst further in the network. The neighbour node which receives the burst is then responsible to reinsert the burst in the network towards its final destination.

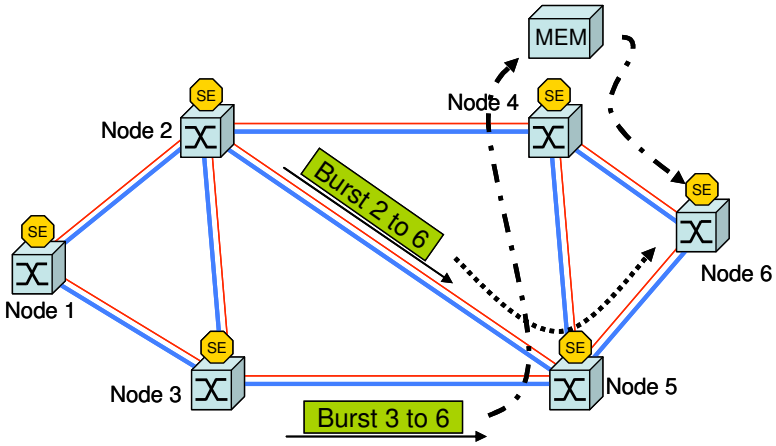


Figure 42 – Next Available Neighbour routing example (CPs are not depicted).

In either case, the burst experiences additional delay due to the O/E/O processing, which in its electronic form may include an amount of buffering time.

This additional delay can now be compared with the delay a burst experiences when it has to be retransmitted. In the case of an OBS architecture and assuming a NAK is sent by the switching node where the burst was dropped to the ingress node signalling the burst drop, the end-to-end delay the burst experiences in a successful transmission and considering only one burst drop in node k along the path of length n is

$$T_{Delay} = 2.T_{Offset} + k.T_{Setup} + T_{E/O} \tag{20}$$

where

- T_{Delay} is the delay the burst experiences from the instant it is ready to be transmitted to the instant in which is transmitted the second time;

- $T_{E/O}$ is the time the ingress node takes to insert the burst in the optical channel, including the time the ingress node takes to interpreter the CP with the NAK message; this time may also include the time spent in the creation of the NAK CP, slightly smaller than T_{Setup}, as T_{Setup} also accounts for the O/E conversion.

In the case of traffic aware architectures such as C^3-OBS, the scenario is different since in this architecture the burst does not enters the network if it is known to drop at any point in the path. In these cases, NAN routing explicitly sends the burst to a neighbour node – and thus the initial control packet is created having as destination the neighbour node, although it is flagged as NAN routing. The neighbour node has then to reinsert the burst (with a new CP) in order to allow it to complete its route to the egress node. In this case, the delay is given by:

$$T_{Delay} = T'_{Offset} + T''_{Offset} + T_{E/O} \quad (21)$$

and

$$T_{Offset} \leq T'_{Offset} + T''_{Offset} \quad (22)$$

where

- T_{Offset} is the offset time required for a burst to complete the path, between the ingress and the egress node;
- T'_{Offset} is the offset time calculated for the route between the ingress and the neighbour node and
- T''_{Offset} is the offset time calculated for the route between the neighbour node and the egress node.

In the particular case where the NAN route coincides with the initially planned route for the burst, as depicted in the example of burst travelling from node 3 to node 6 in Figure 42, T_{Offset} will be

$$T_{Offset} = T'_{Offset} + T''_{Offset} \quad (23)$$

Comparing (20) with (21), we conclude that in similar conditions, *i.e.*, when the burst path does not suffers from deflection, NAN routing time for traffic aware networks is smaller than NAN routing for OBS networks by $T_{Offset} + k.T_{Setup}$.

There are a number of advantages gained by the implementation of the NAN routing in regular OBS, namely:

1. Ingress node buffers are emptied more rapidly than if the burst had to wait for the availability status of the resources.
2. As the number of total hops for the burst travel is split in two or more smaller paths, the probability of finding adequate available resources increases and inversely, the burst loss probability decreases.
3. The burst can be effectively hosted or buffered (electronically) for longer periods of time.
4. NAN routing may be considered as a walk around feature in link failure situations.
5. If a neighbour node can add content to the packet / burst that is waiting to be resent to the destination node, then several bursts can be merged into one, or, several packets can be sent into the data channel close together, benefiting from an additional statistical multiplexing effect, *i.e.*, with NAN routing it is possible to increase the hits for traffic grooming [61].

For C^3-OBS networks, there are two additional advantages:

1. The burst is assembled at the ingress edge node and disassembled at the egress edge node, but it may experience several O/E/O conversions in the routing process when subject to NAN routing. Although it might be considered that this increases the overall delay of the burst transmission, this feature has to be reversely interpreted – based on the information available in the ingress node Local Network Model (LNM), this node may compute and compare the delay the

burst caused by keeping it in the ingress node until the resources are available, with the delay of an extra O/E/O conversion at the destination egress edge node neighbour.

2. In large networks, where the network diameter is so big that it either renders ineffective the propagation of the control packets or makes the reservation of long paths very difficult, NAN routing may be used to allow the sending of bursts through an intermediate node.

Main drawbacks of NAN routing are the following:

1. Increases the effort posed on both the ingress edge or core node, which must decide whether or not perform NAN routing to a packet / burst that is known to be dropped.
2. Increases the effort posed on the neighbour node, by having to receive, interpreter and forward a hosted packet/ burst that has other destination.

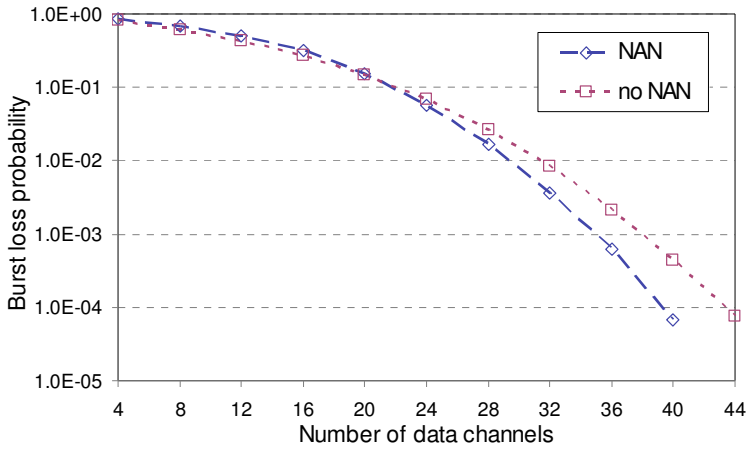


Figure 43 – Performance comparison for an OBS network with Next Available Neighbour routing and with shortest path routing only for the NSFnet topology with 14 nodes and 1 km links, showing burst loss probability versus number of data channels.

Simulation results for NAN for the C^3-OBS architecture are presented in Chapter 6. Figure 43 presents the results for NAN routing in OBS networks, for the 1 km fixed size link 14-node NSFnet topology (see Figure 27) and when the link selecting function was defined as random, *i.e.*, when a burst was to be dropped, it would be randomly sent to an available link, or if no link was available (the case of a node with two links) it would be buffered at that node.

One can see that, expectedly, when network resources are scarce (chart area fewer than 24 available data channels), NAN routing is detrimental to the network performance. This is expected because at this level of resource availability, the network trying to route data beyond its transport capacity and thus each forced burst will cause the drop of one or two other bursts that would otherwise route through.

When the network resources become more available, NAN routing is able to increase the network efficiency by almost an order of magnitude for 40 available data channels, for example (for 44 data channels, NAN routing shows near zero burst losses compared to 10^{-4} of plain shortest path routing in OBS networks).

3.5. Summary

This chapter was devoted to the routing and path selection algorithms and strategies for OBS networks. The routing algorithms are classified as static or dynamic and the main algorithms in each class are presented.

Two new concepts are also presented: the Extended Dijkstra algorithm, a new shortest path static source routing algorithm that chooses paths that have equal costs by selecting the nodes according to its address or identifier tag, resulting in a more balanced traffic routing scenario for networks that have rings, and the Next Available Neighbour routing for traffic aware networks (NAN), a new complementary routing strategy that routes burst or packets that would otherwise be lost in a non-deterministic manner.

To measure the routing patterns created by the new Extended Dijkstra algorithm, two metrics that allow ranking the symmetry and the balance of the routes that are created over the links in the network have been devised. The performance of these new algorithms is assessed, discussed and compared with the strict shortest path routing Dijkstra algorithm for optical burst switched networks.

Chapter 4.

Burst Assembly Algorithms and its Performance Evaluation for IPv4 and IPv6

4.1. Introduction

In this chapter the nature of burst traffic is analysed as a way to better understand its behaviour. As an intermediate step, the understanding of the behaviour of burst traffic and its mechanisms led us to the analysis of tributary IP traffic, also under study in this thesis. This research was the starting point to the invention of the new C^3-OBS architecture.

Inside an OBS network, traffic is composed of bursts which in turn are made of aggregated data packets such as IP packets [9, 10], Ethernet frames [141], ATM cells [142] and so on. Although the burst concept is completely format and encapsulation agnostic, our approach to OBS networks interprets bursts as aggregates of IP packets, so all the research presented here considers IP traffic as the tributary traffic.

This chapter starts with the presentation of the main burst assembly algorithms, often also referred to as packet aggregation algorithms. The performance assessment of the burst assembly algorithms is done using real IPv4 data traces. The inexistence of real IPv6 data traces led to the definition of a method to convert the existing IPv4 real data traces into IPv6 data format.

The part of the research focusing on burst assembly, including its IP tributary traffic, led to two findings: the first one was the finding of a problem related to the widespread use of Ethernet encapsulation of IP packets, the 1500 byte packet size *de*

facto limit [16]. The second one was the assessment of the adequacy of burst loss metrics for OBS networks. The chapter ends with the summary of the main conclusions.

This chapter is partially based on the international patent [18] and on papers [16, 17, 19-22].

4.2. Main burst assembly algorithms

The underlying principle in burst assembly is that a burst assembly queue gathers and manages the packets that have a common destination and a common set of QoS constraints. Following this principle, packets that are destined to a given node and that have low priority are assembled in a different queue than those who have the same destination address but are tagged as high priority, *e.g.*, packets that are marked as email or news content (*i.e.* SMTP, POP3, NNTP) may have a higher burst assembly time threshold than packets that are TCP or RTP (for a complete set of protocols in IP packets please refer to [143]).

Several exceptions have been proposed in opposition to this principle, mostly having in view that the traffic may not be frequent enough to allow the desired efficiency of the burst assembly algorithms, and these are mainly applicable to OBS networks where burst fragmentation is considered. In this situation, when the burst transmission faces contention in the network, some parts of the burst may still be allowed to continued, while others may be dropped or deflected. In this case, the burst reassembly, interpretation and ultimate disassembly are always a responsibility of the destination edge node. In these algorithms, packets inside the burst are not placed in a First-In-First-Out (FIFO) order, but instead they are grouped according to the stated priority of each packet, *i.e.*, the packets are placed in the middle, beginning or end of the burst (see section 2.7.3).

4.2.1. Maximum Burst Size assembly algorithm

In the Maximum Burst Size (MBS) assembly algorithm, the incoming data packets are assembled consecutively into a burst, until its size exceeds the defined

threshold [68]. When this occurs, the last data packet overflowing the current burst starts a new burst, while the current burst is transmitted through the network. Figure 44 shows the flowchart of the MBS algorithm. In this algorithm, the selection of the burst queue for the recently arrived packet includes creating a new burst queue if necessary. For some authors, *e.g.* [144], the overflowing packet is assembled into the existing burst, which may generate bursts that are bigger than the burst size threshold.

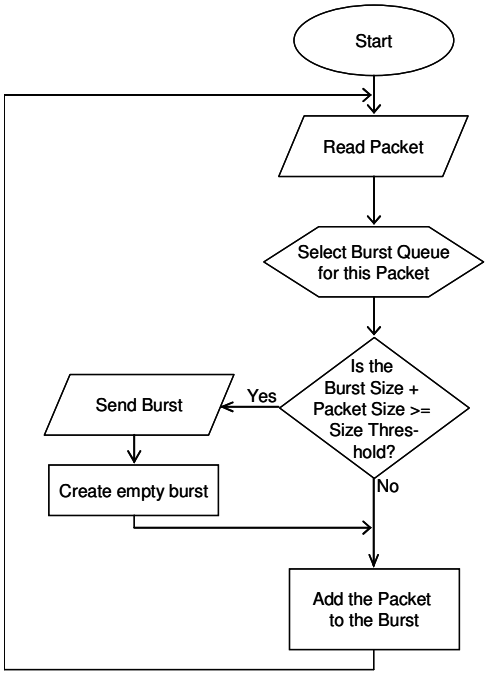


Figure 44 – Flowchart for the Maximum Burst Size algorithm.

4.2.2. Maximum Time Delay assembly algorithm

The Maximum Time Delay (MTD) assembly algorithm was devised to prevent situations where, while using the MBS algorithm, the rate of incoming packets is so low or the arriving packets are so small that it takes an unacceptable amount of time to fill up a single burst, resulting in excessive transmission delay for the aggregated packets. The MTD algorithm checks for the time difference between the time of arrival of the first packet to enter the burst queue and the current local time. The burst is sent into the network as soon as that time difference exceeds the maximum delay time defined, independently of the size of the burst and the number of aggregated packets [69]. Figure

45 shows the flowchart for the MTD algorithm. In this algorithm, the selection of the burst queue for the recently arrived packet includes creating a new burst queue if necessary. If the burst is too small, some authors consider a minimum burst size [69] and propose the padding of the burst until the minimum size is met, with the purpose of diminishing the variability of the bursts size and thus smooth the traffic shape inside the network.

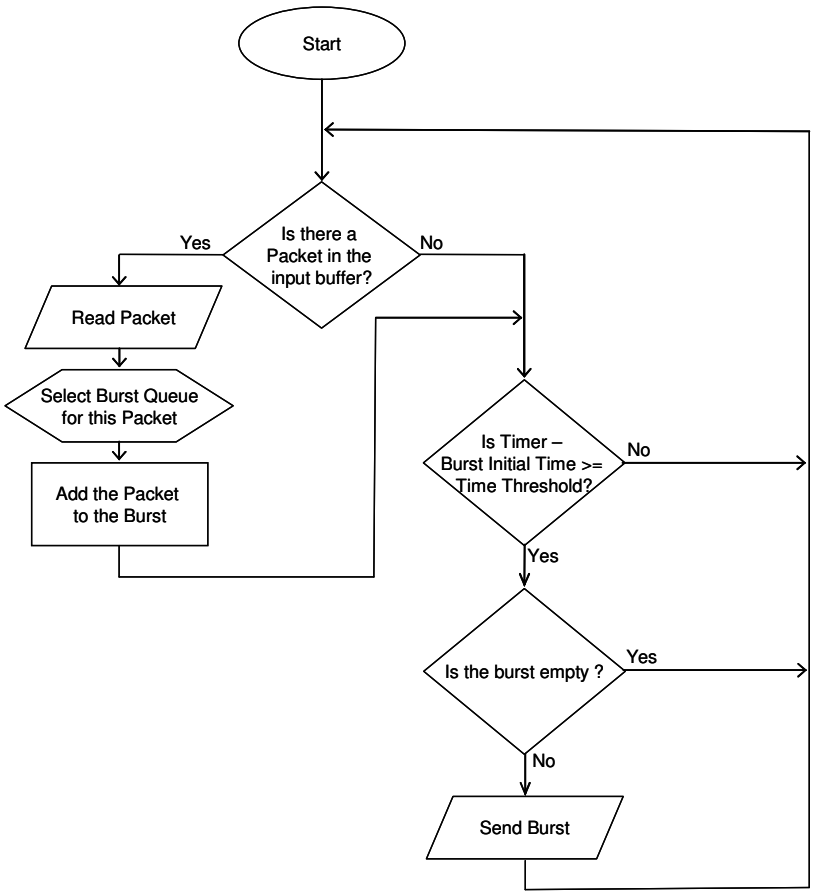


Figure 45 – Flowchart for the Maximum Time Delay algorithm.

4.2.3. Hybrid Assembly algorithm

If the traffic flow rate is too high or the incoming packets are big, the MTD algorithm may end up creating bursts that are too big. In order to prevent the shortcomings of the MBS and the MTD, a Hybrid Assembly (HA) algorithm was devised. In this assembly scheme, both thresholds – time and size – are considered

simultaneously. Incoming packets are assembled into the burst until either one of the threshold conditions is met [70, 71]. Figure 46 depicts the flowchart for this algorithm. In this algorithm, the selection of the burst queue for the recently arrived packet includes creating a new burst queue if necessary.

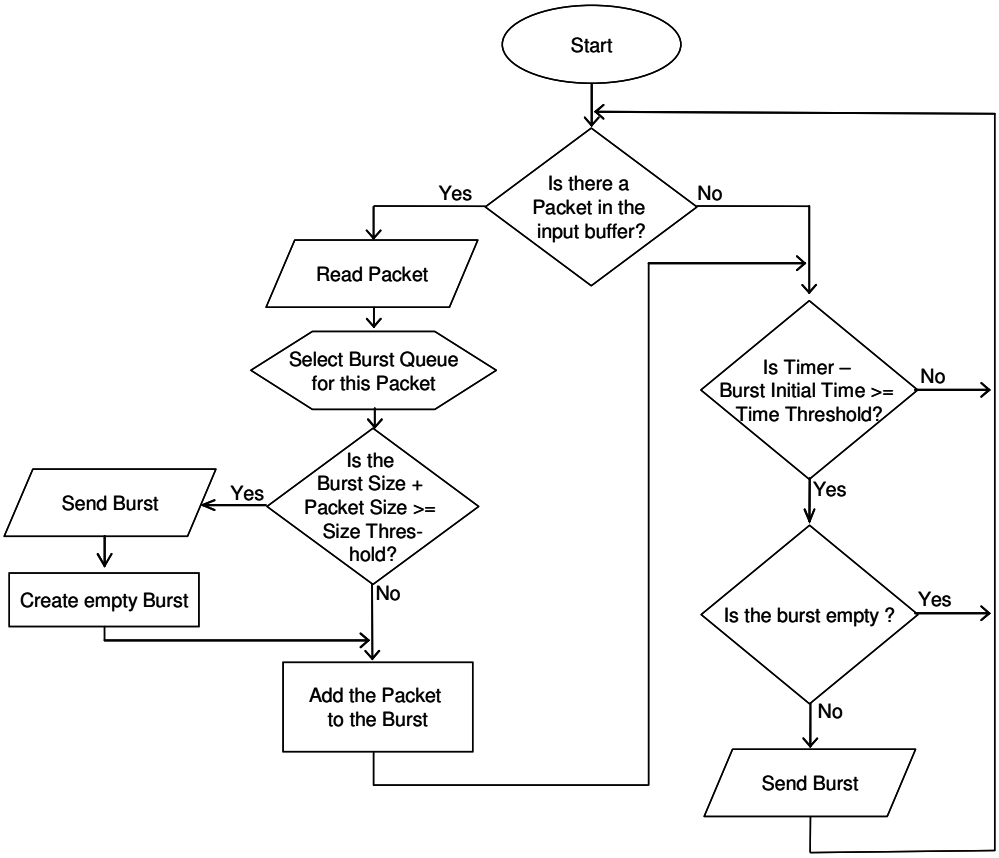


Figure 46 – Flowchart for the Hybrid Assembly algorithm.

4.3. Real IP traffic

The understanding of the nature of data network traffic has been of utmost importance to researchers as well as to the industry since network equipments are engineered and tested through simulation and to simulate network traffic in a realistic way, it is vital to learn and understand traffic characteristics and behaviour.

To the best of our knowledge, there is no study on IP version 6, maybe because there is no abundant multi-application IPv6 network traffic available as opposed to IPv4 based networks, which are pervasive and widespread. Since no IPv6 traces were available, nor could be realistically manufactured because of the scarcity of real and widespread IPv6 networks, IPv6 data streams were generated using a reverse tracking algorithm over existing real IPv4 traffic. The details of this algorithm are presented in this section.

For the analysis of the IPv4 traffic, data traces downloaded from NLANR [145] where used, containing the traffic captured on a series of servers [146], within a known time window.

4.3.1. Real IPv4 traffic

The data recorded from NLANR was obtained in [147]. This data is presented in files that record data packet traces in a time stamped header (*tsh*) format.

The *tsh* file format stores the payload stripped data packets, time stamped at their acquisition. The typical IPv4 data header is extended by the timestamp field (4 bytes for second timestamp and 3 bytes for microsecond timestamp), expressing the timestamp of the captured data packet relative to the 1st of January 1970. Moreover, the IP header is also extended with TCP information, comprising Source and Destination ports, Sequence and *Acknowledgement* numbers and other TCP specific information. The standard format of the *tsh* data packet header [148] is shown in Figure 47.

In order to assure IP address security, the Source and Destination Addresses disclosed in the IP *tsh* packet header section are hashed to preserve the anonymity of the original machines. However, the IP hashing algorithm is designed in such a manner that it preserves the IP address space density, thus class A servers shall always have lower hashed IP number than class D machines. The source code for the IP address hashing procedure is available from the NLANR website (see [145]). Packet payload is not recorded but its size is accounted for in the *Total Length* field of the *tsh* record (see Figure 47).

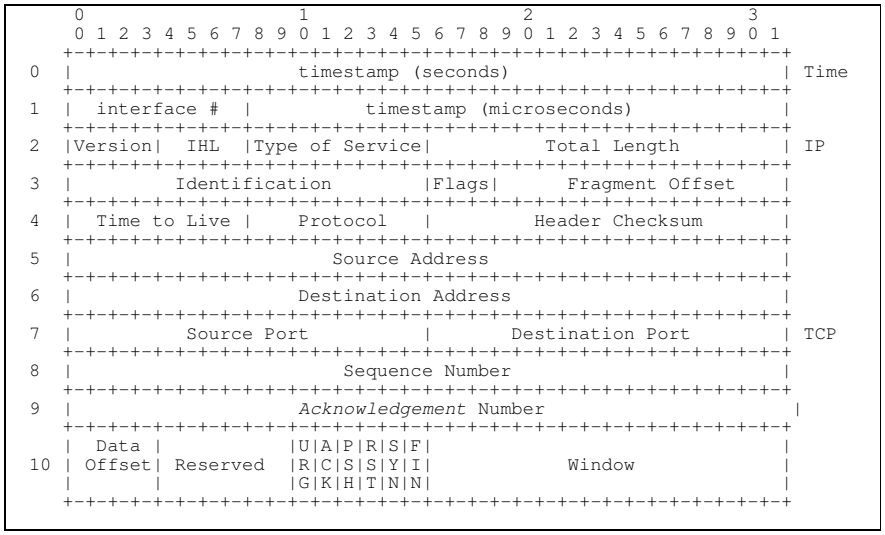


Figure 47 – Internal format of the *tsh* data packet format from NLANR [148].

4.3.2. Examined data set

The used data set comprises 268 data traces obtained from NLANR data server and collected for several servers in the USA between the 8th and the 10th August 2006. The examined data traces include in total over 51 million packets covering various network conditions [145]. Selected sites were the *AmericasPath* exchange point at Miami, Florida (AMP files), the Front Range *GigaPOP* which is a consortium of Universities, non-profit corporations and government agencies that cooperate in an aggregation point called the FRGP in order to share Wide Area Networking (WAN) services (FRG files), the University of Memphis, which has a POS (Packet over SONET) logical OC-3c link to Abilene's KSCY at Kansas City (MEM files) and the Texas universities GigaPOP at Rice University, Houston, Texas (TXS files). Table 8 shows a summary of the characteristics and configurations for the selected batch files; more detailed information is available in [145]. Figure 48 shows a partial packet size graph for a TXS data trace.

Table 8 – Characteristics of data collection sites.

Site	*Link type*	*Framing*	*Encapsulation*
AMP	OC12c	ATM/AAL5	LLC/SNAP
FRG	OC12c	POS	CHDLC
MEM	OC3c	POS	CHDLC
TXS	OC3c	ATM/AAL5	LLC/SNAP

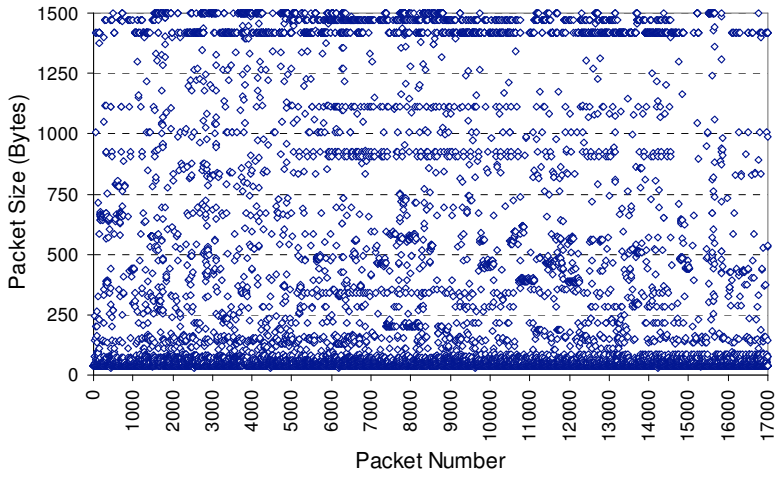


Figure 48 – IPv4 packet size (partial graph for the TXS-1123774395 trace).

4.3.3. Generated IPv6 traffic

When converting the IPv4 packets in the data traces into IPv6 format, two different methods were followed: on the first method, it was assumed that the 1500 byte limit exists for each individual packet and thus packets have to comply with this maximum size. This method is described in section 4.3.3.1. On the second method, described in section 4.3.3.2, the approach was slightly different as there was the need to consider the following hypothesis – if an application generates a payload of, for instance, 1600 bytes and the MTU is 1500 bytes, this payload will be encapsulated in two packets, but it is not mandatory that the first packet will be 1500 bytes long and the second will be 180 bytes long (40 bytes from the IPv6 header plus 1460 bytes payload in the first packet and 40 bytes from the IPv6 header plus 140 bytes of the remaining

payload), as the payload can be evenly split for the two packets (840 bytes in each IPv6 packet) and in this event the header replacement performed by the first method will not result in MTU overflow for any of the packets. The approach followed considering this hypothesis was to try to retrieve the original payload distribution at each source, as to allow the manufacturing of packets in a suitable manner. For this second method, the payloads were grouped together following a full source and destination key and afterwards encapsulated in packets for MTU sizes of 1500 bytes, 9 KB and 64 KB. The 9 KB limit is related to the limit of Ethernet Jumboframes [149] and the 64 KB limit is naturally related to the maximum size of a basic IP packet.

The observed 1500 byte size limit is not related to IP or TCP limitations and our best assumption is that this size limit has two different sources. First, it is related to path and/or link MTU (Maximum Transmission Unit), as described in [10, 150-153]. The value of 1500 bytes seems to be commonly used for link MTU [154] and appears to be a legacy value, as more recent path/link MTU discovery RFCs (Request for Comments) suggest [154] (see also [155, 156]). Second, but probably more important, this limit is related to the Ethernet frame size. Local area networks are mainly Ethernet based and these networks are the largest tributaries of traffic to core networks. Naturally, network equipment is designed and configured to efficiently handle traffic from the tributary networks, thus the 1500 byte MTU. This appears to be a vicious-circle type of problem, since local area networks do not implement technologies that increase the packet frame size to prevent packet partitioning and network equipments keep MTUs to values that match the maximum packet size of the underlying networks. Although in this research the scenarios where the 1500 byte limit was removed were assessed, it seems clear that IPv6 networks that are implemented over Ethernet will still bare this burden.

After the application of each the remanufacturing algorithms, two metrics were applied to the original and reprocessed traces: relative overhead on the number of bytes and relative overhead on the number of manufactured packets. As the results for the size overhead are not relevant, only part of them are presented here.

This procedure allowed us to infer how the IPv6 traffic would be like if the packet size limitation was 1500, 9216 and 65536 bytes and also, to assess the risk of

equipment overloading with unnecessary data packets which are otherwise generated because of the *de facto* 1500 bytes size limit.

Finally, the IPv6 generated traffic followed the Original Payload Retrieval Algorithm described forward, with an aggregation threshold of 447 microseconds.

The departure point for this research using real IPv4 traffic, and the results obtained through the analysis of the traffic traces, led to the conclusion that IPv6 traffic characteristics will be similar to the IPv4 traffic, mostly because of the restrictions posed by the Ethernet frame size, possibly only scaled by the increase of number of transmitted packets.

4.3.3.1. Header replacement

The first approach is clearly the worst case scenario, as it keeps the 1500 bytes limit and remanufactures the data traces on a packet by packet basis, *i.e.*, the remanufacturing of the data traces was made by a simple replacement of the 20 bytes long IPv4 header by a 40 bytes long IPv6 header. By performing this header replacement, the IPv4 packets that sized 1500 bytes already, now became 1520 bytes long IPv6 packets thus causing an overflow for the network MTU settings. To comply with the MTU limit, the payload is split in two packets and thus a 1500 bytes long IPv4 packet originates two IPv6 packets, the first one with 1500 bytes and the second one with 60 bytes, 40 from the additional header and 20 from the previous packet overflow (see Figure 49). Eventual additional headers that were present in the original packet were not considered for this new packet, without loss of generality.

This procedure was applied each time the header replacement caused the creation of a packet that was bigger than 1500 bytes, *i.e.*, to all IPv4 packets whose size was bigger than 1480 bytes. As it can be observed intuitively in Figure 48 there is a significant part of the total number of packets that are close to the 1500 bytes upper limit.

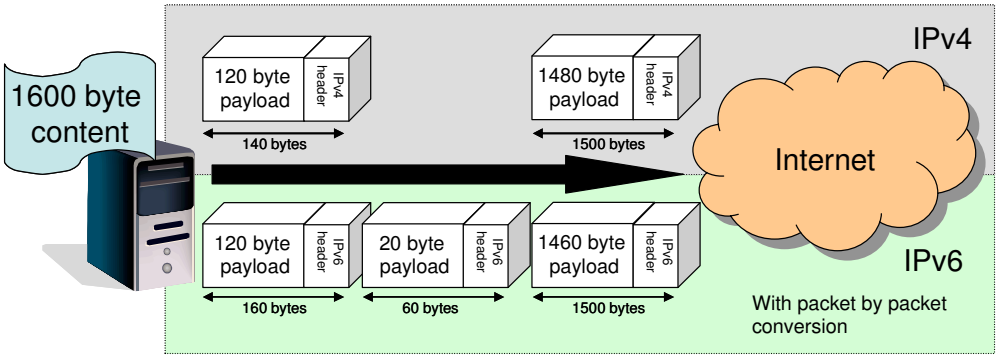


Figure 49 – Sample IPv4 to IPv6 packet conversion under the Header Replacement algorithm.

4.3.3.2. Original payload retrieval

On the second approach, it was attempted to discover the original sizes of the payloads of the packets, mostly because it is admissible that a payload of 1600 bytes does have to generate IPv4 packets that are 1500 and 140 bytes long respectively – it can rather create two IPv4 packets that are 820 bytes long each (800 bytes payload plus 20 bytes for the IPv4 header), in which case the application of the first method will not create MTU overflow for none of the packets (see Figure 50). To implement this, the packets were reverse-engineered to retrieve its original payload distribution, *i.e.*, assuming that if two or more packets had the same IP address and port for both the source and destination (here referred to as full source-destination key) and if they have a close enough timestamp, then they may have been created by the same unique event on the host application and thus the resulting payload was split at a lower layer because of ruling restrictions, such as *e.g.* the Ethernet payload size or the MTU. As for the time threshold used for the assumption of "same originating event" the values used were 100 s, 500 s and 1 ms. This set of time thresholds was defined having in mind the desired maximum delay posed to the packets and also following the work in [157]. These thresholds were applied to the selected traces for the original 1500 bytes MTU and subsequently with the 9 KB and 64 KB size limits. The 9 KB limit is related to the limit of Ethernet Jumboframes and the 64 KB limit is the maximum size of a base IP packet.

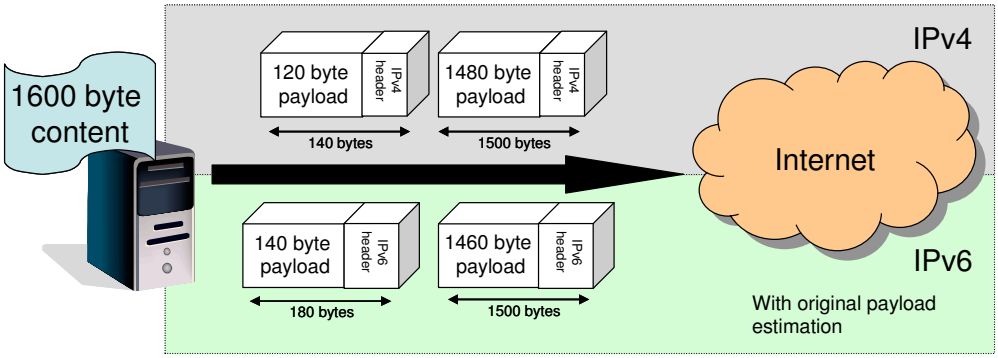


Figure 50 – Sample IPv4 to IPv6 conversion under the Original Payload Retrieval algorithm.

4.3.4. Results

For each selected trace collecting point, nine files were randomly selected, sampling data for more than 42 GB of traffic. The maximum, minimum and average original number of packets in the trace files is presented in Figure 51. The FRG traces record the biggest traces and thus account for the higher share of traffic in the analysed traffic. Still, as shown forward, the results are coherent for each data trace file and are independent of the collection point.

On the first research scenario, it was assumed that the IPv4 to IPv6 change consists in the removal of the IPv4 header and the addition of the IPv6 header. In such a situation, the overall size of the packet increases by 20 bytes and thus some packets will overflow the 1500 size limit. To assess how important this limit is, the packets that were present in the network and actually were bigger than 1500 bytes were counted. In the selected files, the number of packets bigger than 1500 bytes was found to be only $0.52^{o}/_{oo}$. Thus it must be assumed that the network collection sites we selected show a massive presence of Ethernet transport at some point, which is good as it allows overlooking the effect of large packets in this research and to assume, without loss of generality, that all packets are in fact smaller or equal than 1500 bytes.

Figure 52 shows the maximum, minimum and average number of packets bigger than 1500 bytes for each collecting point. The number of packets bigger than 1500 bytes for TXS is zero, as for FRG, the largest trace files, packets bigger than 1500 bytes are, at the most, $0.19^{o}/_{oo}$ of the data; the AMP and the FRG data trace show results very

close to zero. The trace that shows a larger number of packets bigger than 1500 bytes is MEM, but even here this value reaches at the most 0.026% being the average of the trace files of around 0.014%.

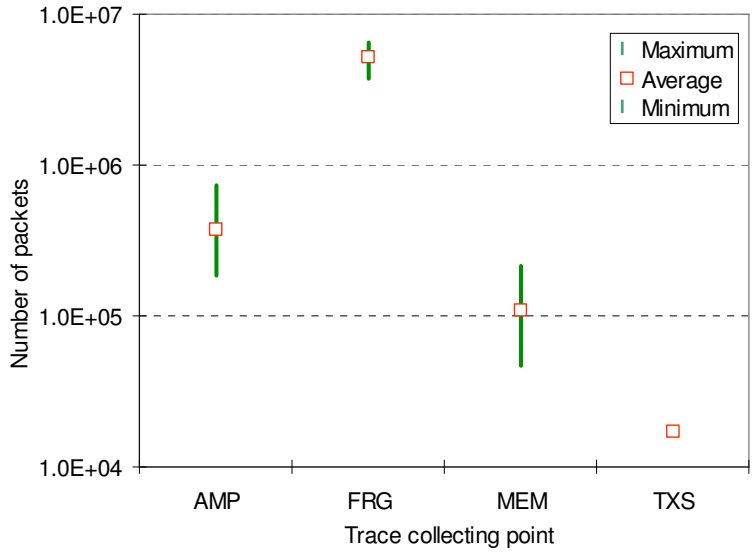


Figure 51 – Maximum, minimum and average number of packets in the selected file traces per network collection point.

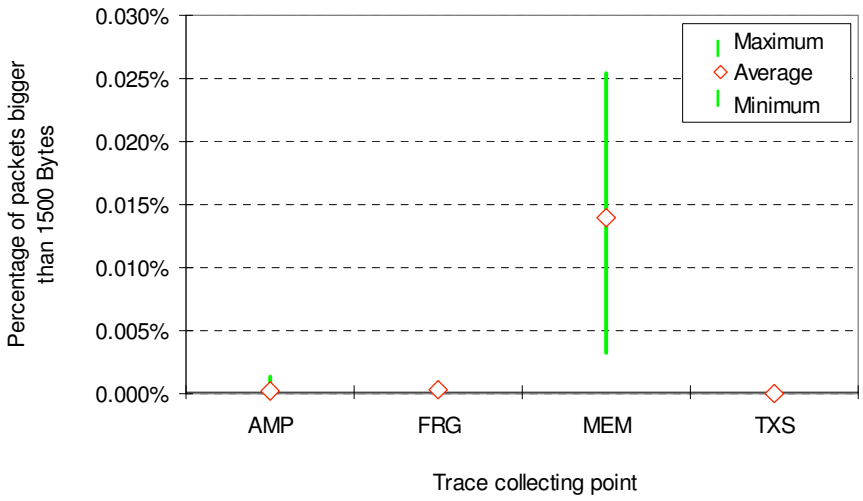


Figure 52 – Maximum, minimum and average percentage of packets bigger than 1500 bytes in the selected traces.

4.3.5. Header replacement algorithm

Following the setup of the first simulation scenario described in section 4.3.3.1, all the packets in the trace files were submitted to a simple operation: the size of an IPv4 header was subtracted to its total size and added the size of an IPv6 header. If the result was bigger than 1500 bytes, a new packet was created with the excess size and a new header was added to this new packet.

By this conversion, the increase in the number of transmitted bytes was not limited to the size difference between the IPv6 and an IPv4 headers, it was also increased by the IPv6 header size times the number of new IPv6 packets created. As it can be seen from Figure 53, the overall increase in the size of the transmitted data, *i.e.*, the increase in the number of transmitted bytes due to the conversion, is expectably small, 1.375%, 2.194%, 0.943% and 1.308% for the AMP, FRG, MEM and TXS traces, respectively.

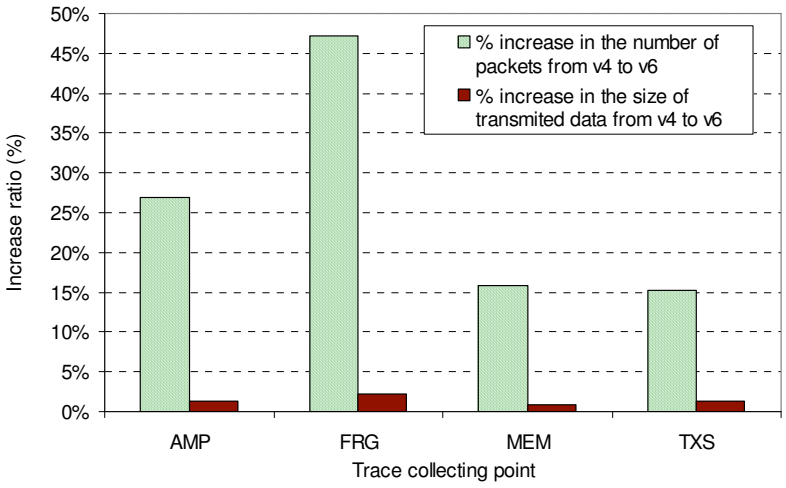


Figure 53 – Increase on the number and size of packets in the selected traces, converted from IPv4 to IPv6 by the application of the header replacement algorithm.

On the other hand, the increase on the number of packets that where created is very high, in the case of the FRG trace, the simple conversion operation from IPv4 to IPv6 would cause more than 45% in average of travelling packets in the network

structure. On the other traces, this value represents at least 15% more packets. For the overall, the increase in the number generated packets reaches an astonishing 45.162%.

4.3.6. Payload reconstruction algorithm

The second simulation scenario, described in section 4.3.3.2, takes into account the original generation of the payload for the data packets. With this, we say that it is conceivable that two consecutive packets – one very big and other relatively small – may have been generated by an end user or an application and thus the methodology applied in the first scenario will not reflect the real changes from IPv4 to IPv6 in the traffic profile. In order to minimize this eventual situation, a special aggregation on the packets was performed, as follows: if two (or more) packets have the same source IP address and port and are destined to the same destination IP address and port and have enough close timestamps, then they may be considered as a single packet. This aggregated packet was built by adding the size of the payloads of the time-neighbour packets and the creation of IPv4 or IPv6 packets was consequently performed by the addition of the suitable header. For the aggregation time thresholds the values of 100 and 500 microseconds and 1 millisecond were selected, as it was previously detailed for the first simulation scenario. For the size thresholds, the selected values were of 1500 bytes, 9 KB and 64 KB, respectively for the original MTU size, the size of an Ethernet Jumboframe and the maximum allowable size for an IP packet.

Results for a MTU of 1500 bytes are shown in Figure 54. Expectedly, the increase in the time threshold that allows the aggregation following the "same originating event" principle shows a decrease in the creation of the number of packets. For time threshold zero, the ratio in the number of created packets is very close to the values obtained in the first algorithm, shown in Figure 53. In Figure 54 it is shown that when the time threshold is smaller to 300 microseconds the ratio of created packets is positive for all traces. For the 100 s threshold, the traces show an increase of around 20%. The results for 500 and 1000 s show already the statistical multiplexing effect resulting from the traffic aggregation [20, 32, 33, 158].

As a side effect of this research and in view of the results shown in Figure 54, we can affirm that the packet creation events at each source have an average decision time between 300 and 700 microseconds, for different traces, as these values correspond to the zeros of the plots. The negative ratios observable at higher time thresholds correspond to the statistical multiplexing effect brought by aggregation and thus are not inherent to the packet creation event. Detailing further more and using a simple extrapolation function in the plots in Figure 54, we found the zeros for the fitting functions to be approximately 494, 446, 293 and 332 microseconds for the AMP, FRG, MEM and TXS traces, respectively. A weighted average for the overall traces returns an estimation of 447 microseconds as the zero of the plot. This value will be use as our best estimative to the time threshold following the "same originating event" principle.

As a result of the previous assertion, we may claim that if the packet creation processes in the tributary machine/application pair is faster the estimated 447 microseconds, the network will have an increase in the IPv6 generated packet number, being this increase at the most around 45%, this later being the result of the weighted average for the delays shown at time threshold value of 0 microseconds.

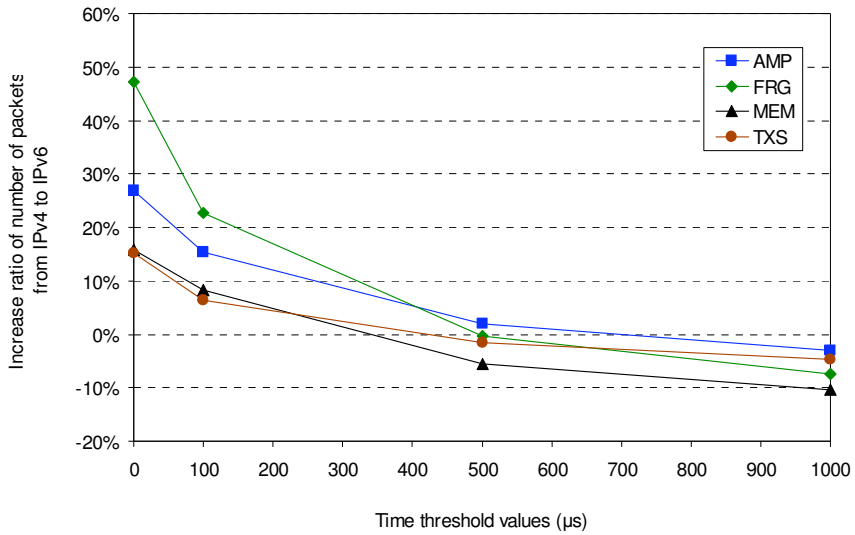


Figure 54 – Percentage of increase on the number of packets for the conversion of IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 µs, for MTU=1500 bytes.

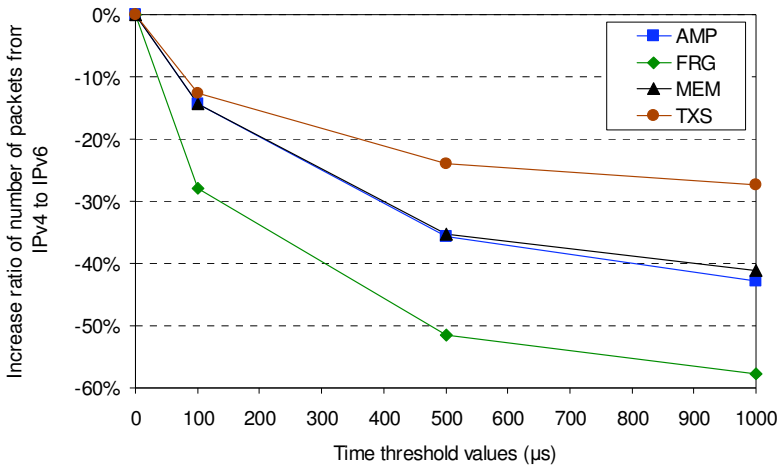


Figure 55 – Percentage of increase for the conversion of IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 µs, for MTU=9K bytes.

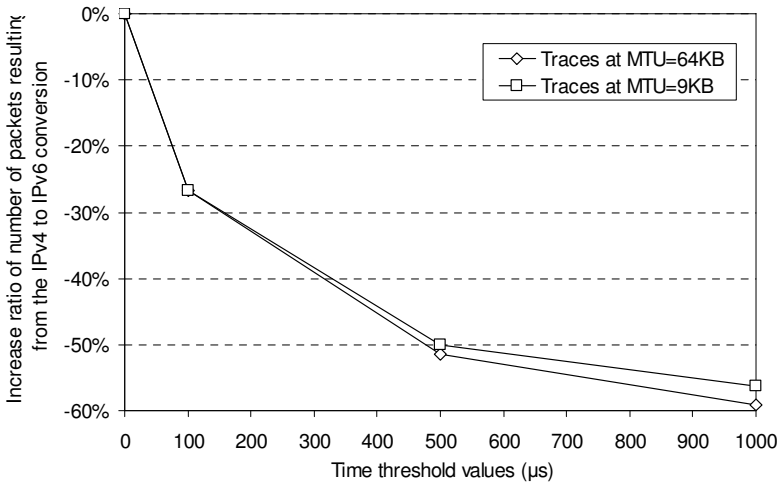


Figure 56 – Weighted percentage of increase on the number of packets for all traces for the conversion from IPv4 to IPv6 packets when time for the "same originating event" is set to 0 (Scenario 1), 100, 500 and 1000 µs, for MTU=9K bytes and MTU=64K bytes.

Figure 55 shows the ratios obtained when the packet size was freed of the 1500 B limit and raised to 9 KB, the size of an Ethernet Jumboframe. Assuming that the time for "same originating event" for the several sources is around 400 microseconds as estimated before, it is visible that this MTU change would cause the number of packets traversing the network to drop around 40%. While this change in the MTU would

probably demand equipments with larger buffers, it would also mean fewer switching operations and thus, faster routing. Also visible is that the most active network points show the highest multiplexing aggregation gains, which is an expected result.

Figure 56 shows that a further increase on the packet size limit from 9 KB to 64 KB would allow only for a marginal benefit in the decrease of the number of generated packets. The fact that only a small increase is achieved when raising the limit from 9 KB to 64 KB is probably a consequence of the way the applications and the hardware on the client machines are designed, namely in terms of buffer size and queue operation, or also possibly related to operating system response times to network request events. It seems that the generality of the computer and network systems involved in traffic generation and routing for the studied traces, cannot provide traffic with sufficiently high throughput as to show data aggregation benefits when the packet size is allowed to be bigger than 9 KB.

We can also conclude that the IPv6 traffic will have a similar shape and behaviour to IPv4, except possibly for a light increase on the number of packets, as the increase in the number of bytes is minimal, and the packet generation times are not changed because these are mostly a function of the underlying applications.

4.4. Burst assembly simulation and evaluation metrics

In this study, IPv4 packets were used as tributary data to the burst assembly process and no encapsulation of the aggregated packets was performed. The disassembly mechanism should thus consider the first 20 bytes of the data burst to be an IPv4 header and proceed to extract that packet from the aggregated data. This step is repeated until no data is left within the burst. In this study no Class of Service (CoS) was considered, mainly because the Type of Service (ToS) field in the IPv4 packets in the data traces does not bear reliable information.

Burst assembly tasks are performed taking into account two burst characteristics: burst size and burst assembly time. When remanufacturing IPv4 into IPv6 packets, we have concluded that the increase in the number of bytes is minimal. Furthermore, the

real packet arrival times were not changed in any of the procedures presented in section 4.3.3, so, as burst assembly algorithms are blind to the number of packets they carry, and following the results presented in section 4.3.3, we conclude that the burst assembly tasks will behave similarly for IPv4 and for IPv6 traffic. Therefore, this section is focused on burst assembly of IPv4 packets.

4.4.1. Evaluation metrics

In recent literature, burst sizes are often defined in terms of their time length, which in turn allows for their measurement in bytes, given a known channel data rate. Since we are using real IPv4 as tributary data for our research, the sizes are defined in bytes (B) and times are defined in microseconds, due to *tsh* format specifications. For the burst itself, a suitable set of metrics was used to evaluate the performance of burst assembly algorithms:

a) average packet delay per burst;

b) average number of packets per burst;

c) average burst size;

d) average burst inter-arrival time.

The average packet delay per burst is important since it provides insight into the value of the mean packet delay imposed by the operation of the burst assembly mechanism. The acceptable delay for a single data packet is one of the main *criteria* when selecting an adequate burst assembly algorithm. The mean number of packets per burst additionally shows how much processing power the disassembly mechanism needs to perform to forward on individual packets. Primarily, the mean number of packets per burst demonstrates the amount of switching effort saved in the network structure due to the statistical multiplexing effect. While the mean burst size is important to determine the degree of occupancy of a network channel, the mean burst inter-arrival time will provide an insight about the ratio at which the bursts flow in the network structure.

Other metrics, *e.g.*, the One Way Delay, as defined in RFC 2679 [159] and the IP Packet Delay Variation Metric, as defined in RFC 3393 [160], produced by the IETF IP Performance Metrics workgroup (IPPM) [161], were not fully adopted because it was intended to keep this study independent of the link speed and these measures imply the knowledge of the link transmission ratio, since both measure the performance from the first to the last bit conveyed over a particular transmission medium.

4.4.2. Burst assembly variables

Burst Assembly was performed with several thresholds. Time thresholds used were 100 s, 500 s, 1 ms, 10 ms and 100 ms and size thresholds used were 9 KB, 64 KB and 1 MB. The first two size thresholds constitute the maximum sizes of respectively the Ethernet Jumboframe and the IP packet. The third threshold was originally set to 4 GB consistently with the maximum size for an IPv6 Jumbogram [12], but it was soon discovered that even 1 MB burst size required burst assembly times in excess of 1 s (for "MEM-1111612868-1.tsh", see chosen trace files below), thereby making the 4 GB size threshold of little research interest. Hybrid assembly was performed for the following six scenarios: size thresholds of 9 KB and 64 KB and time thresholds of 500 s, 10^3 s and 10^4 s.

4.4.3. Results

The results presented were obtained by simulation for the burst assembly process executed for the following three randomly chosen traces: AMP-1107250616-1.tsh, ANL-1109366607-1.tsh and MEM-1111612868-1.tsh. AMP, ANL and MEM stand for AMPATH, Miami, Florida (OC12c), Argonne National Laboratory to STARTAP (OC3c link) and University of Memphis (OC3c link), respectively [145]. Other data trace files were also subject to simulation process, producing results coherent with the ones presented here. Each of the examined trace files records the activity in the selected network point for about continuous 91 seconds. Table 9 shows the activity of each of these network points.

Table 9 – Network Activity for the selected trace files.

	AMP	*ANL*	*MEM*
Packet load (in Bytes)	876 301 540	246 923 708	159 669 572
Time span (in s)	89 836 840	90 921 971	91 918 790
Offered load (in MB/s)	9.754	2.716	1.737

4.4.3.1. Results for Maximum Burst Size

Figure 57 depicts the relative frequency of burst inter-arrival time for each of the studied data traces with the burst size threshold set to 9 KB. As anticipated, the AMP data results in the lowest inter-arrival time plot, since 90% of the bursts arrive within 1555 s or less and this trace contains the highest traffic load. For the other traces, the same limit is observed at 616 s and 12903 s for ANL and MEM, respectively. It is worth noting that the plots do not share the same shape – around 2500 s ANL and MEM switch tendencies – this suggests that burst inter-arrival time depends on the nature of the traffic on the network point where bursts are to be assembled.

Figure 57 does not present the whole burst inter-arrival scale: AMP reaches its maximum at 10001 s, ANL at 30038 s and MEM at 55702 s (for MEM, 99.95% of the burst arrive up until 35103 s), respectively. The same behaviour is also visible for other size thresholds in all the studied traces.

The comparison between burst inter-arrival times for different size thresholds lead to the conclusion that an increase of two orders of magnitude in the burst size (from 9 KB to 1 MB) results in an increase of more than two orders of magnitude of the burst inter-arrival time. This result is depicted in Figure 58. It also shows the influence of network offered load on burst inter-arrival time. Here, it is assumed that the nature of the traffic in the three network points is comparable and the offered throughput was calculated using values from Table 9. When the network load increases by a factor of five, the average burst inter-arrival time decreases by an order of magnitude. This behaviour is also verified for several burst size thresholds.

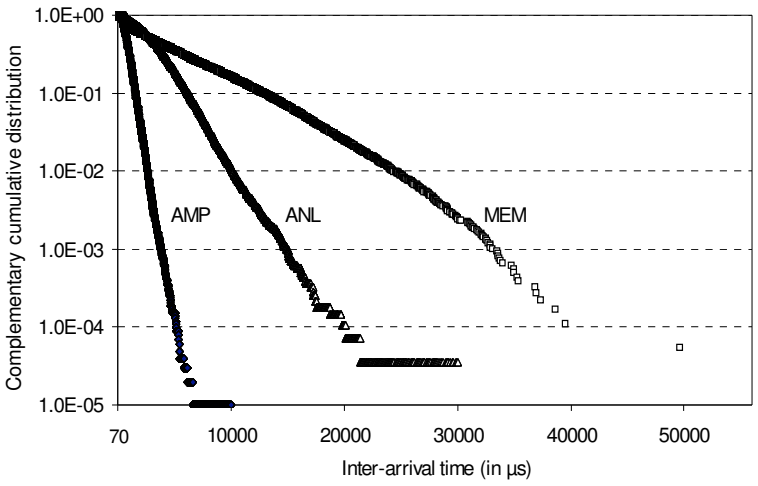


Figure 57 – Complementary Cumulative Distribution function for burst inter-arrival time for MBS threshold = 9 KB.

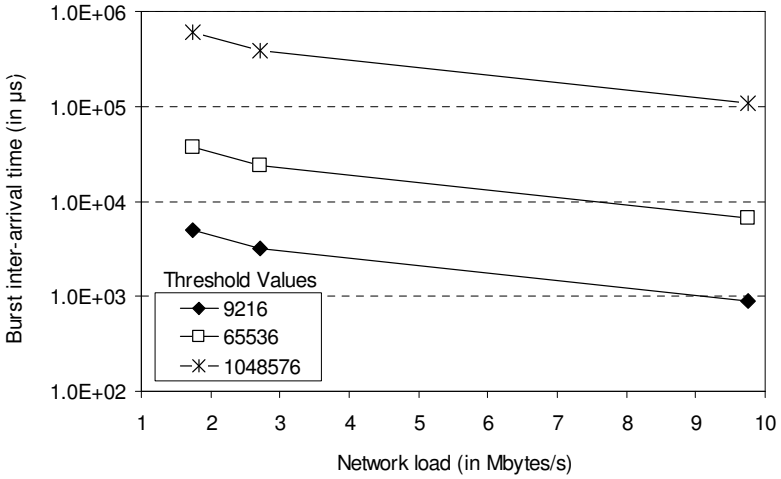


Figure 58 – Weighted average burst inter-arrival time versus network load for MBS.

Figure 59 depicts the average packet delay per burst for the MBS assembly scheme. For the studied data traces, when the size threshold is set to 9 KB, the average packet delay per burst values are almost equal, diverging only when the size threshold increases. As anticipated, the higher load network point shows a lower packet average delay. The increase in the average packet delay when size threshold is increased from 9 KB to 64 KB is different for the data in AMP when compared to the data in ANL and MEM. ANL and MEM also here seem to follow a close behaviour, expectedly

suggesting that when the link is more loaded, the MBS aggregation algorithm performs better in terms of average packet delay per burst.

Accordingly with the definition of the burst assembly algorithm, burst size is very close to burst size threshold and no graph is shown.

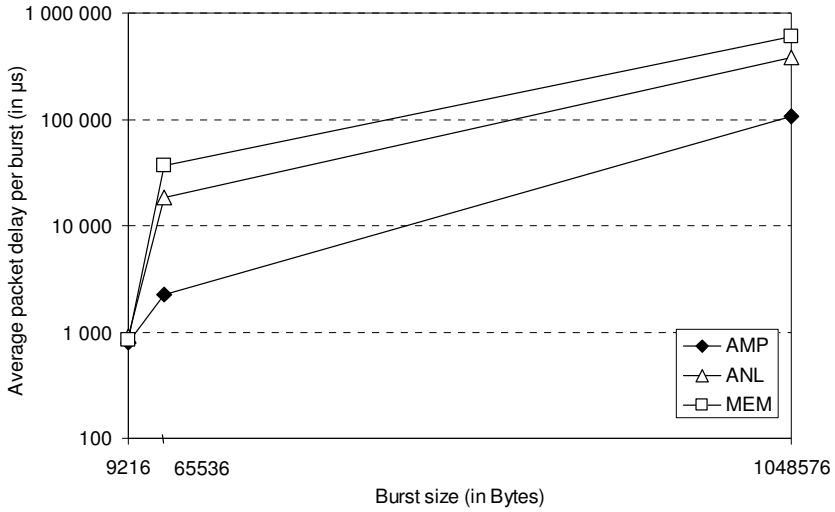


Figure 59 – Average packet delay per burst for MBS.

4.4.3.2. Results for Maximum Time Delay

As expected, the average inter-arrival time between bursts is always higher than the time threshold defined for the burst assembly. Figure 60 shows the relation between the weighted averages for the burst inter-arrival times for MTD plotted against burst assembly time threshold. It is clear that inter-arrival and threshold times converge when the later increase. AMP holds the lowest relation between inter-arrival and threshold times, thus performing better than ANL or MEM, which is a consequence of its higher traffic load.

There is a *quasi* linear relation observed in all data traces between the number of packets contained in the bursts and the utilised time threshold, as shown in Figure 61. As anticipated, the higher load traffic data yields a higher number of packets per burst and the number of packets in bursts per data traces for AMP and ANL / MEM diverge

as the time threshold increases. At a higher aggregation threshold value, there is a difference of almost one order of magnitude in the number of packets per burst between AMP and ANL / MEM.

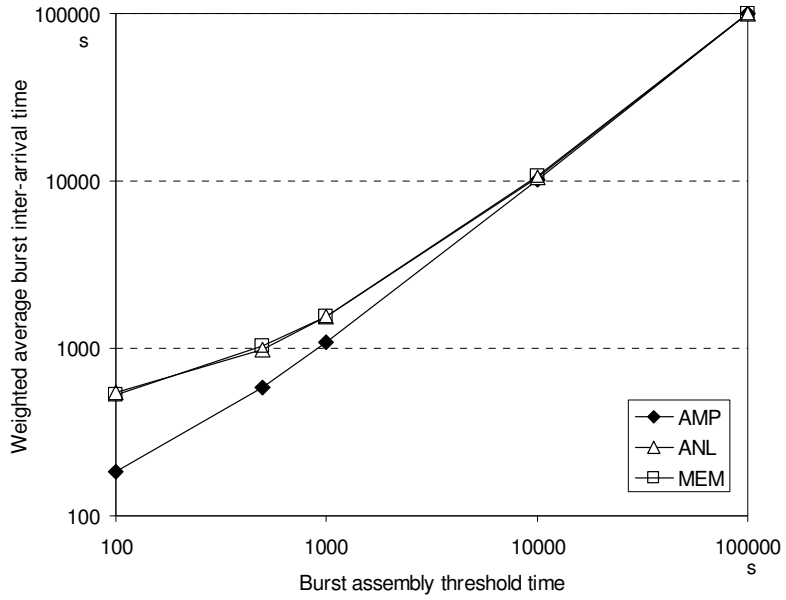


Figure 60 – Burst inter-arrival time versus burst assembly threshold time for MTD.

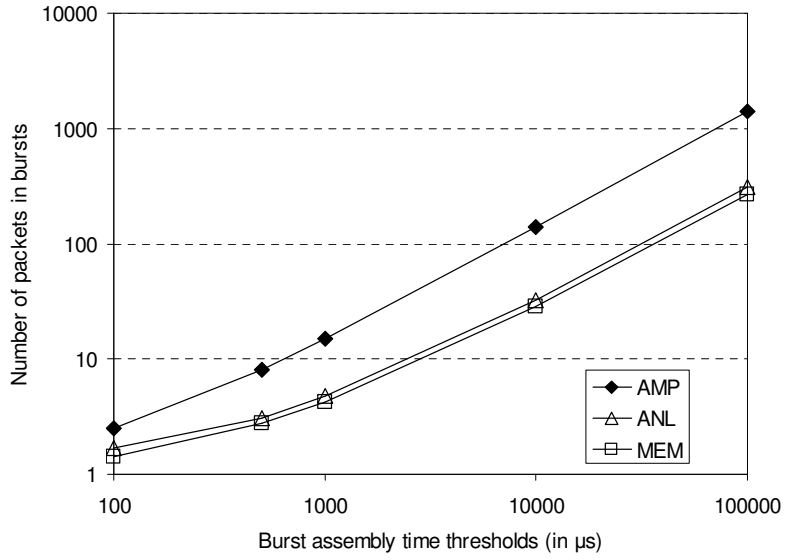


Figure 61 – Number of packets in bursts versus burst assembly (aggregation) thresholds for MTD.

Burst Size depends on the MTD aggregation ratio since the faster the incoming packets arrive, the bigger the bursts are. Figure 62 shows how data is aggregated into bursts of variable sizes when the time threshold is set to 10^4 s. In line with well known results [162, 163], we can see that the burst size distribution resembles a Gaussian shape when the number of generated bursts increases, which happens when we consider the MEM trace, an expected result that follows the Law of Large Numbers.

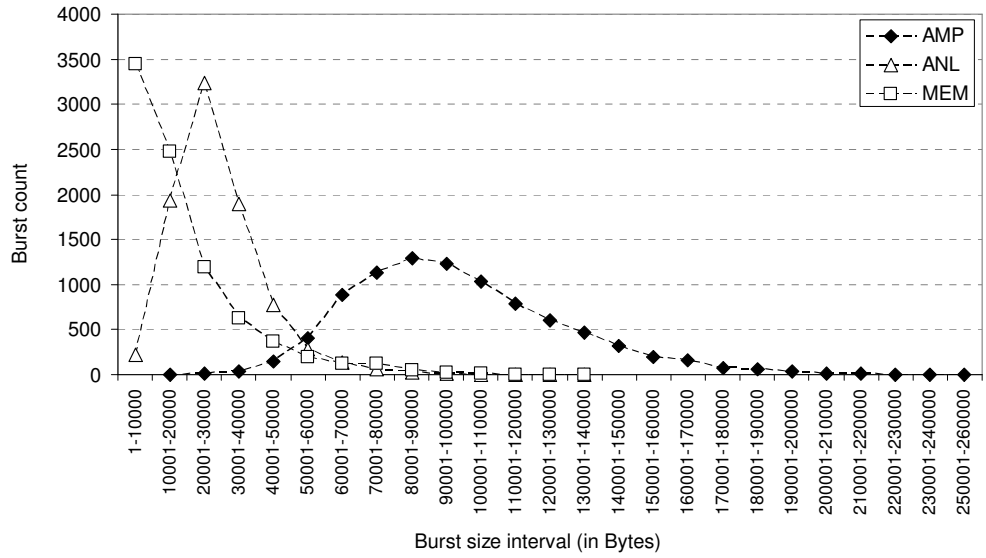


Figure 62 – Burst Size Histogram for MDT when time = 10^4 s.

The plot character for each data trace is different, even for sources with comparable traffic load such as ANL and MEM. This result suggests that the bigger the load of the tributary traffic link, the more the burst size distribution tends to the Normal distribution. The Normal distribution nature of some of the characteristics of the aggregated bursts is also visible when the packet count per burst is plotted. For a burst assembly time of 10^5 s, bursts contain from 88 to 1887 packets and follow a distribution close to the one depicted in Figure 62.

Figure 63 shows the results when time was set to 100 s (the lower end of the chosen burst assembly threshold scale). At this burst assembly threshold, only high intensity traffic sources achieve bursts with more than 16 packets (the maximum was one burst in AMP which contained 21 packets). This figure shows the complementary

cumulative distribution function of packets in bursts for this particular burst assembly scenario. Here, bursts containing up to three packets account for 78.5% of the traffic with AMP, 95.1% with ANL and 98.3% with MEM trace files.

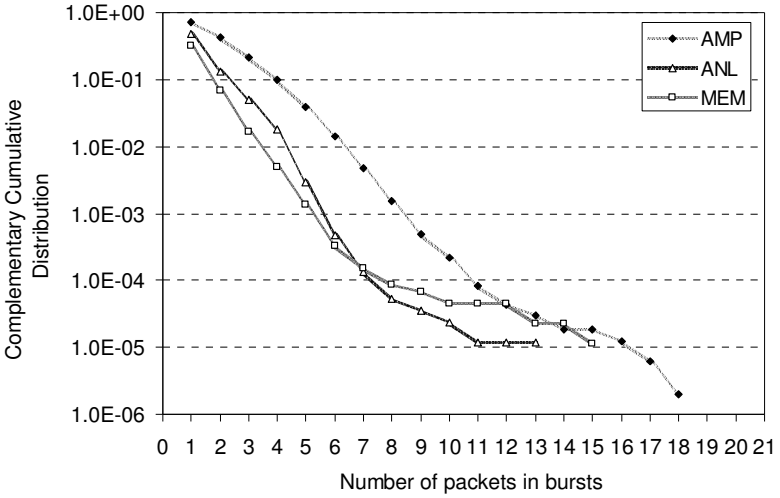


Figure 63 – Number of packets in bursts relative distribution for MTD = 100 s.

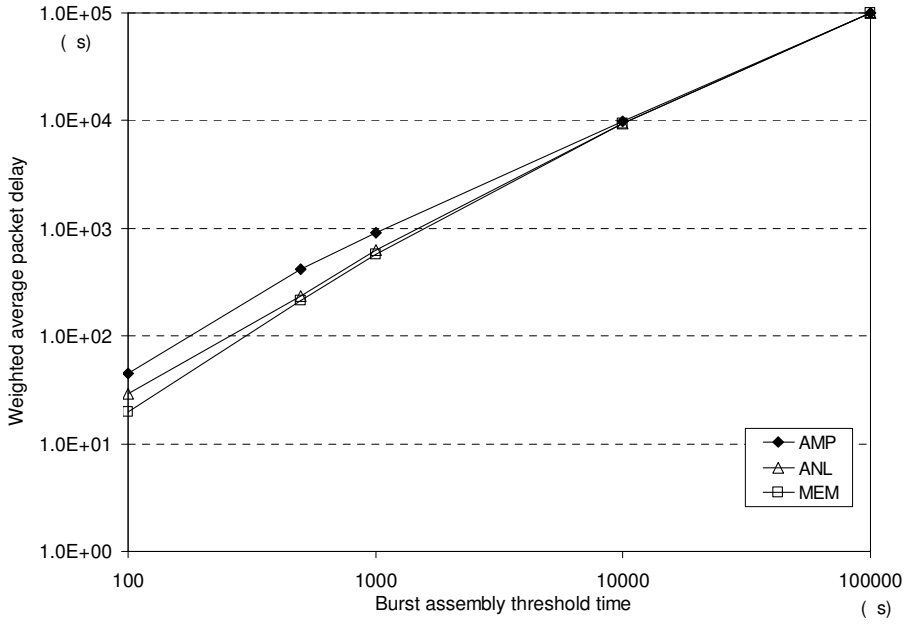


Figure 64 – Average delay of packets in bursts versus burst assembly threshold times.

The average packet delay for MTD is shown in Figure 64. Note that the average packet delay converges to the burst assembly threshold when it is higher than 10^3 s. Additionally, low load traces (ANL and MEM) show better performance for MTD in terms of average packet delay than AMP; in case of low time thresholds, MEM performs almost twice better than AMP, possibly because bursts contain fewer packets.

4.4.3.3. Results for Hybrid Assembly

The Hybrid Assembly (HA) algorithm combines two thresholds: time and size. Figure 65 shows the packet inter-arrival time for three data traces, for each HA scenario. The tendency depicted in Figure 65 remains when regarding packet delay time per burst. It is clear that ANL and MEM behave similarly and for burst assembly times below 10^3 s inclusive and the performance of the HA is almost constant regardless of whether the burst size threshold is set to 9 KB or 64 KB. AMP performs better probably because of its higher packet load and performs better with time threshold of 10^4 s and burst size 9 KB than with time of 10^3 s. This behaviour may be related to the change in the burst assembly threshold (from size to time or vice-versa).

This hypothesis is confirmed by the plot of Figure 66. Here, the nice dragon-like shape of the plot exhibits several interesting features of the HA algorithm. The data plotted for values of bursts bigger than 7500 bytes (approximately) show the effect of the size constrained branch of the algorithm; these were the bursts that contain fast incoming packets, *i.e.*, they were very close together in time so the time threshold was never reached. It is also visible that these bursts are responsible for a major share of the output stream (bursts whose size is bigger than 7500 bytes account for 74.70% of the total bursts count).

The lump shapes on Figure 66 are a well known phenomena related to the size distribution of IPv4 packets [155]. IPv4 packets that are 1500 bytes long are very common in data traces, supposedly because of path and/or link MTU issues and also, as a consequence of the popularity of IP over Ethernet encapsulation scheme. This graph shows local modes near the multiples of 1500 bytes, namely 1500, 3000, 4500, 6000, 7500 and finally 9000 bytes. Note that 75% of the bursts in this scenario have 7500

bytes or more, which is in line with previous results, in particular the ones depicted in Figure 61 and Figure 62 considering MTD. This behaviour is also visible in other data traces and in other burst assembly scenarios.

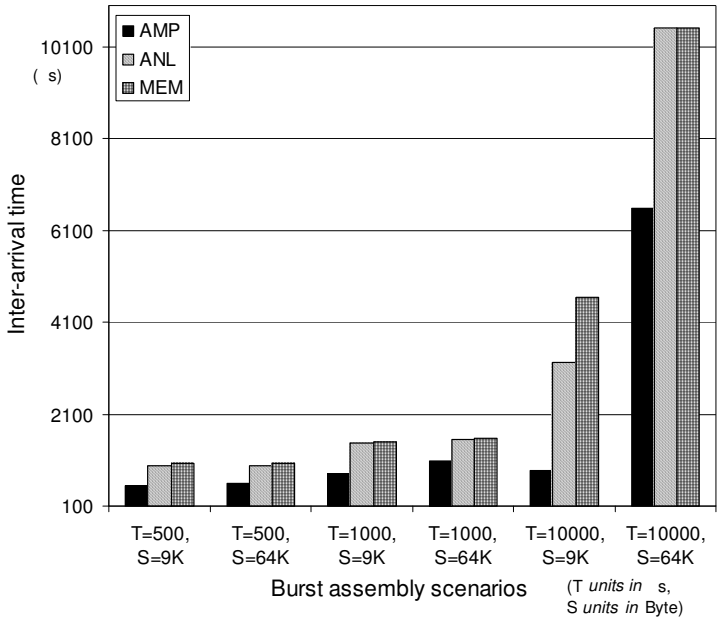


Figure 65 – Burst inter-arrival time versus burst assembly scenarios for HA, considering three network collection points (AMP = AMPATH, Miami, Florida, USA, ANL = Argonne National Laboratory to STARTAP, MEM = University of Memphis).

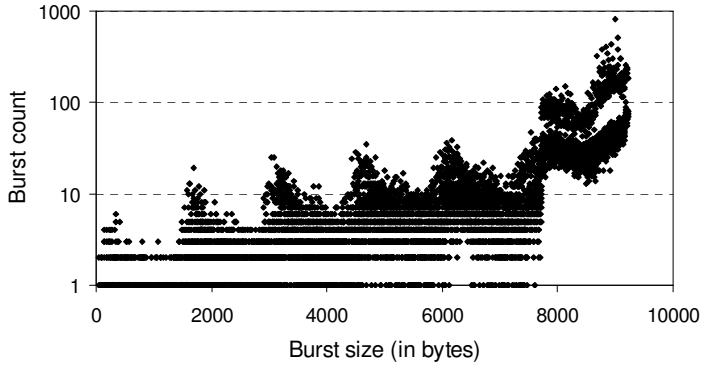


Figure 66 – Burst size histogram for HA ($T=10^3$ s, S=9 KB) on AMP data.

4.4.3.4. Performance assessment of burst assembly algorithms

The assessment of the performance of burst assembly algorithms using real IPv4 data allows for the following conclusions:

1. Globally, from all the studied scenarios for the burst assembly algorithms, both MTD and HA outperform MBS in terms of delay added to the burst and/or to the packet in the burst. Therefore, as the assembly of large size bursts results in high inter-arrival times and packet delay per burst, these large bursts should only be considered for transmitting data that is not time critical, such as news servers synchronization or data backup between servers. The size threshold is dependent of the traffic load on the input link, typically being bigger than 1 MB for these data traces;
2. The performance of the burst assembly algorithms is also a function of the network point where the data is to be assembled, as different network sites show different performances. This feature is visible in all examined burst assembly algorithms and several threshold values may have to be studied to reach a sustained optimum performance for a specific network point. Thus, for an optimum operation regarding the compromise between large bursts and low mean packet delay in a burst, the combinations of thresholds for the burst assembly algorithm have to be dynamically set to adjust to time changing traffic conditions;
3. As IPv6 packet sizes and flow rates are expected to follow current IPv4 models, following the research presented in [20] and the conclusions in section 4.3.3, we can expect similar results for the assembly of IPv6 packets into bursts.

The results presented here and also published in [20] show that, even for such a short time span, there are several candidates to rank as the best burst assembly algorithm. We defined the more effective algorithm as the one that, for a specific traffic scenario, achieves the biggest bursts (to maximize the statistical multiplexing effect) and the lowest mean packet delay (to optimize the data flow in the network). These two criteria served to conjointly rank the performance of the burst assembly algorithms. This

results in that there may not exist a single “Best Burst Assembly Algorithm”, but rather an algorithm that performs best for a given network traffic scenario.

Figure 67 shows the rank of the tested algorithms concerning two parameters: Burst Size and Mean Packet Delay per burst. The data plotted on this figure is arbitrarily sorted on the average Packet Delay for the ANL data trace. It depicts two Y axes, the left one associated with increasing values of burst size and the right one associated with decreasing values of packet delay. Dotted lines show mean packet delays per data trace (right Y axis) and solid lines show burst sizes per data trace (left Y axis). Regarding the rank of the burst assembly algorithms, as may be seen, concerning the burst size (left Y-axis, solid plot lines), the higher its value the better the performance of the burst assembly algorithm and thus the “MTD 100000” and “MBS 1MB” scenarios show the best results. Inversely, these scenarios show the lowest performance regarding mean packet delay. As to the packet delay, the ranking of burst assembly algorithms is viewed against the right Y-axis (dotted plot lines). As this axis depicts values in decreasing order, higher points in the graph (depicting lowest packet delay values) show the best burst assembly algorithms. As expected, “MTD 500” and “HA T=500 S=9K” perform almost as well as the original packet delay because of its very short threshold; the “MTD 100” scenario shows better performance than the original trace – something that is physically impossible, but explained by the method used in the calculation of the average on the packet delay per burst - as some bursts were timed-out with zero packets, their average packet delay per burst was zero, thus decreasing the overall average.

Figure 67 clearly shows that, if the minimum per packet delay is set around 1 ms, the “HA T=1000 S=9K” and “MBS=9K” scenarios show a good compromise between both rank criteria. If the minimum per packet delay is increased a bit more but still around 1 ms, we can see that the scenarios “HA T=10000 S=64K” and “MBS 64K” are still a possible choice, but only for the AMP trace and not for the other two traces, as in these the packet delay rises to 10 ms and more.

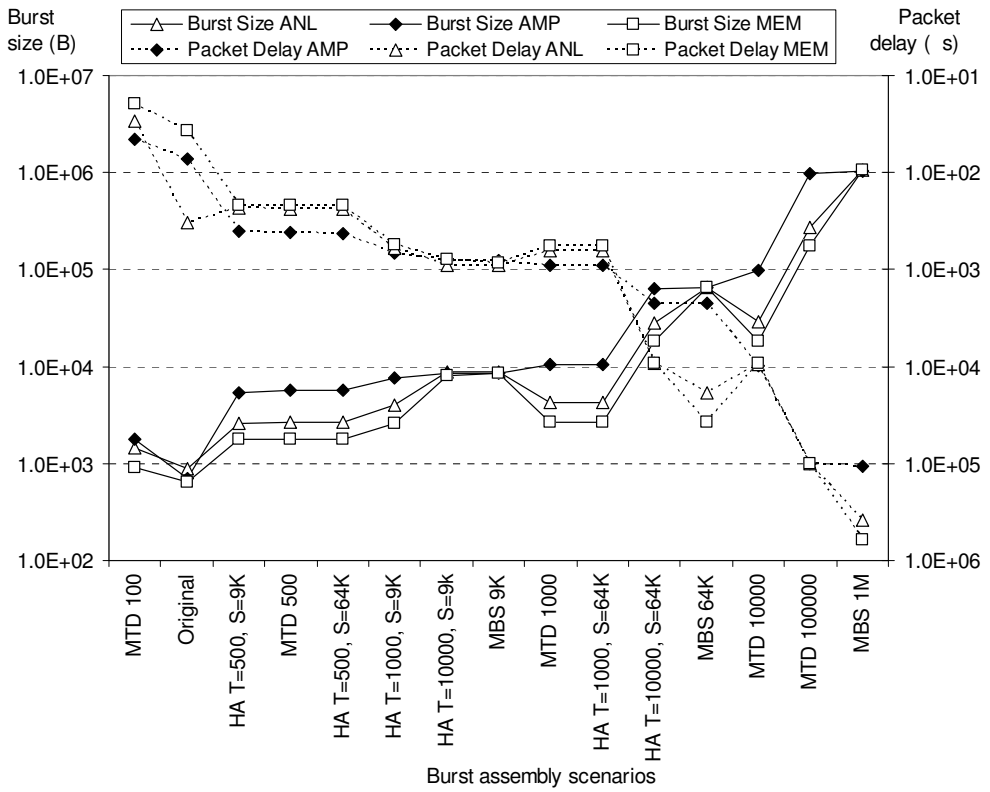


Figure 67 – Performance comparison for the burst assembly algorithms versus burst assembly scenarios, depicting both average burst size (solid line plots) and average packet delay (dotted line plots).

In line with the conclusions published in [20], it may be seen that the burst assembly algorithms and thresholds must be carefully selected in order to improve efficiency and that these may need to be dynamically adjusted to optimize the burst assembly algorithm performance.

4.5. Relevance of OBS network performance metrics

The performance of burst switched networks is often measured in terms of burst loss or burst drop probability. Recently, [146] proposed that burst loss was not equal to packet loss and these values vary within the same range. Research activities described here show different results for several assembly scenarios and particularly that there

exists an equivalence relation between burst loss and packet loss, although the later is of more interest to the end user than the former.

4.5.1. Basic assumptions and burst assembly algorithms

To test the assumption that burst loss and packet loss have different meanings in OBS networks, we used to simulator described in section 6.5, feeding it with real IP for packets. Again, packets used in the simulation are real IPv4 packets, recorded from NLANR and obtained in [146]. Additional information on this data is provided in section 4.3.1.

In order to assure IP address security, the source and destination addresses disclosed in the IP *tsh* packet header section are hashed to preserve the anonymity of the original machines. Issues on addresses are important because burst assembly is primarily performed in a "by destination" basis. The simulation handled the computation of the destination addresses for the bursts based on the destination address in the packet *tsh* data as follows: when an address was extracted from the packet, it was looked up in an address table. This address table contains two entries: the first is the IP address itself and the second is the pseudo-address of the destination machine. If the extracted IP address is not yet present in the address table, then a random pseudo-address is assigned to it as its destination and this pair was added to the table. This way, the full initial address space was homogeneous and randomly distributed over the available pseudo-addresses of the destination machines. This task is repeated in each node, as a way to closely mimic the hash of the initial IP address space. As an example, while hashed address *12345* processed in node *A* refers to destination machine *X*, it may refer to destination machine *Y* when the same file is processed by node *B*.

The network topology simulated was a four-node bidirectional ring. Shortest path routing was used and full wavelength conversion was assumed for the OBS simulation, using JIT and also the JET signalling protocols. JIT is an immediate reservation protocol and does not perform void filling and thus every burst is treated independently of its size. On the other hand, JET is burst size sensitive as it performs

delayed reservation and attempts void filling, so burst size is important to maximize the efficiency of network resource reservation.

The topology and the remaining default simulation parameters are not relevant for the focus of this research, as a change in these would only alter the performance of the network in terms of burst loss ratios. The simulation was performed with a large set of burst assembly thresholds to allow a wide range of loss ratio values and thus test the possible correlations of burst and packet loss over the whole counter-domain.

4.5.2. Burst assembly simulation

The algorithm used for burst assembly in this research was HA, with several different thresholds. Thresholds were varied allowing HA to emulate MBS, with time threshold set too high and MTD, with size threshold set also too high, for a given network load. Thresholds used for burst size were set to 64 KB and 9 KB and assembly time varied from 100 s to 2000 s for 64, 16, 12, 8, 4 and 1 users in each edge node. Time thresholds and user load were combined to assure that burst loss really occurred in the network with burst loss ranging from 1.445% to 98.966%.

Burst assembly algorithms using real IP traffic were studied in [19] and presented previously in this chapter. Figure 67 shows how different sets of thresholds change the performance of burst assembly algorithms and consequently define the optimum zones for operation as function of the tributary traffic characteristics, defined as the scenarios that find a compromise between the minimum inter-arrival time between bursts and the maximum burst size.

Since the efficiency of burst assembly algorithms is a function of the point in the network where the data is assembled into bursts [19], HA was used with a wide set of thresholds as to obtain a large range of burst characteristics. The result was the creation of bursts very differentiated in terms of size in bytes and size in number of packets, being this clearly visible in Figure 68. The values ranging from 0.905% to 85.043% show the ratio of standard deviation calculated over the averaged burst size in bytes and burst size in packets.

The relevant results the provided by the simulator were: number of bursts, size in bytes for each burst, size in packets for each burst, for both created bursts and dropped bursts. The ratio of created over dropped bursts was calculated and averaged for several simulations with different simulation time lengths.

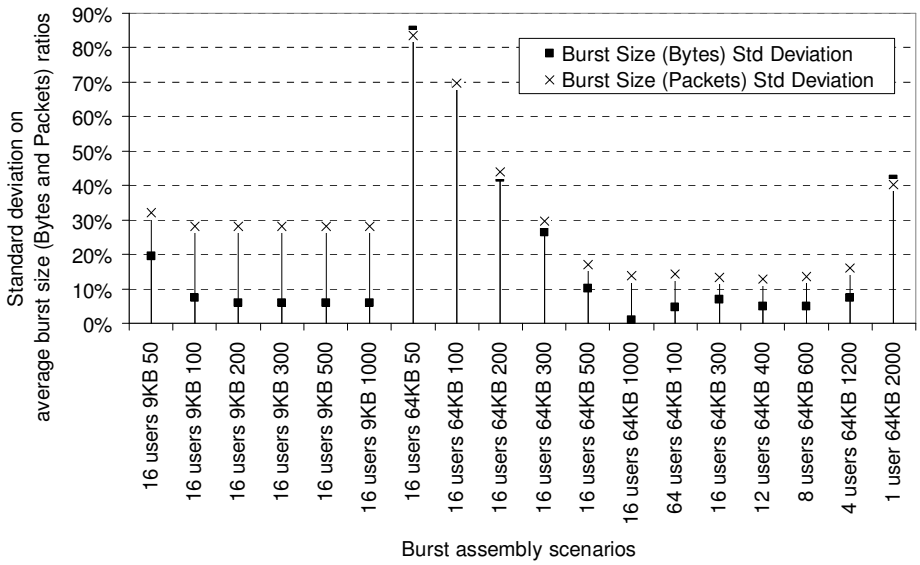


Figure 68 – Standard deviation ratio of average burst size (measured in bytes) and average burst size (measured in number of packets).

4.5.3. Burst loss versus packet loss

The primary metric used for performance assessment of burst switched networks has been burst loss. There are a number of underlying assumptions in this statement that can be expressed in a simplified form, as follows:

1) all bursts are made of independent smaller data entities, which may be called packets (without loss of generalization);
2) all bursts are equally sized;
3) all bursts contain an equal number of packets.

If these three assumptions are hold true, then there is no doubt that burst loss metric is an adequate performance assessment measurement and what is more, Burst

Loss, Packet Loss and Byte Loss ratios are equal. But if bursts are not equally sized, what does it mean that a network lost a burst – exactly how many bytes were in this burst and what is more, how many packets were lost? That is to say, the Burst Loss metric may not be relevant to real networks, who are know to exhibit self-similar bursty traffic [164-167].

Also, to the end users – machines and humans using the network – burst loss may not be meaningful. The expected performance the network is supposed to deliver, is measured in terms of "how long and how well is this content taking to travel from machine *A* to machine *B*" and this often means "how many packets were lost" and "how delayed the packets were". This also points out to conclusions already known from the study of burst assembly algorithms using real IP packets: minimum packet delay and maximum burst size, *i.e.* the optimization of burst assembly process is achieved for the HA algorithm using time and size thresholds that are function of the network load on the burst assembly machine and thus, its efficiency depends on the point in the network where the burst is assembled [19]. As a result of the optimization of the burst assembly process, it has to be assumed that realistic burst switching deals with bursts that are not homogeneously sized and of course do not contain a fixed number of packets [19].

If the three above mentioned assumptions can not be held true, as in the case where very heterogeneous burst traffic is generated which is the case simulated and presented here, only two alternatives remain: i) either burst loss is not adequate as a performance assessment metric because it is not equal neither to byte loss neither to packet loss and the later would be more "user meaningful", or ii) with real traffic the simulation follows the Law of Large Numbers and so, the final results on the network can be assumed as if all the bursts have the same number of packets and these in turn are equally sized, to the average number of packets per burst the first and the average number of bytes per packet (and per burst) the later.

The network was loaded with bursts assembled from real IPv4 packets and the number of data channels were defined in order to allow burst losses. The remainder simulation parameters are presented in section 6.5.

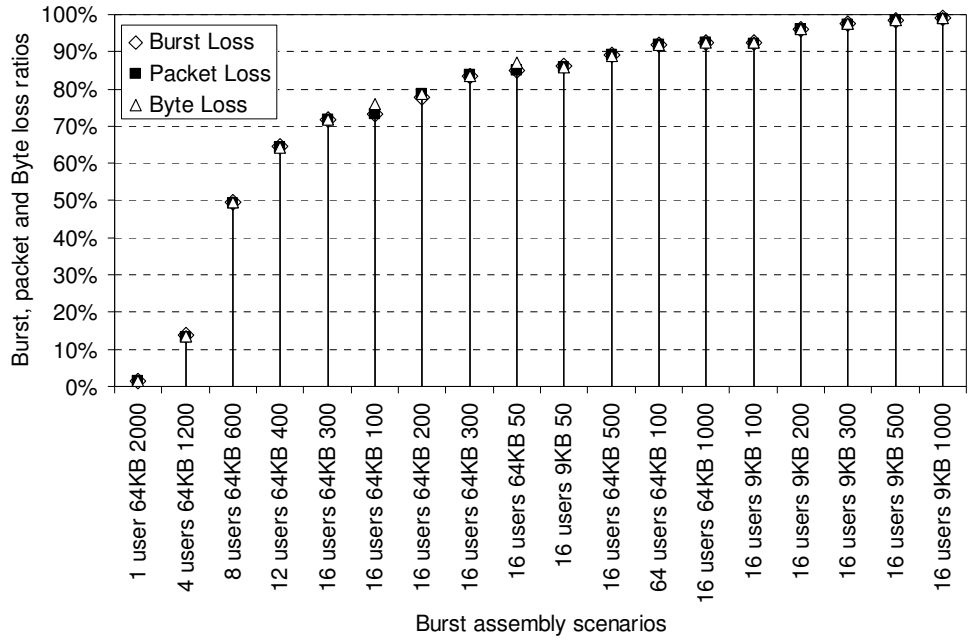


Figure 69 – Burst, packet and byte loss ratios for different burst assembly scenarios in an OBS JIT four nodes ring network (time thresholds in x-axis are s).

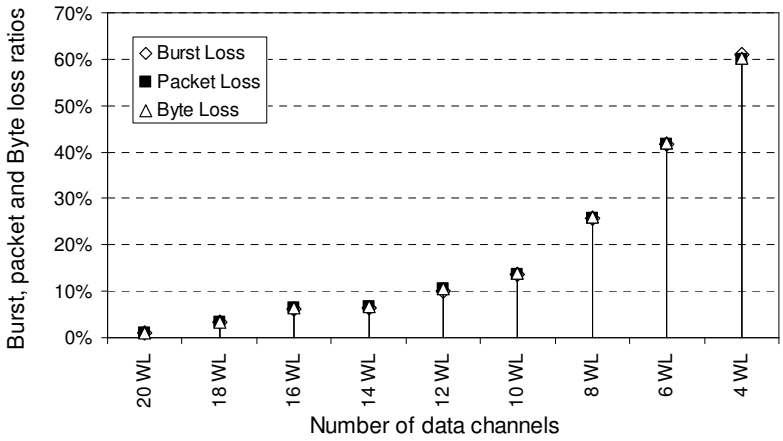


Figure 70 – Burst, packet and byte loss ratios for different burst assembly scenarios in an OBS JET four nodes ring network, 64 KB burst size threshold, 40 s time threshold and variable number of data channels (in x-axis of the graph).

A set of three ratios was devised and implemented in the simulator as follows:

i) Burst Loss Ratio is the number of bursts dropped divided by the number of bursts created at edge nodes;
ii) Packet Loss Ratio is the sum of the packets in the bursts that were dropped divided by the number of packets assembled in bursts at the edge nodes;
iii) Byte Loss Ratio is the sum of sizes (in bytes) of bursts that were dropped divided by the sum of sizes (of bytes) of created bursts at the edge nodes.

The three values were computed for all the simulated scenarios. Figure 69 and Figure 70 show the obtained results for JIT and JET signalling protocols respectively, with several burst assembly scenarios. As expected, despite such a wide range of burst characteristics in terms of size in bytes and number of constituent packets and also, despite of the differences in the way the network signalling protocols accept or drop the bursts, the Burst Loss, Packet Loss and Byte Loss ratios, are almost coincident.

4.5.4. Conclusions

Kantarci, Oktug and Atmaca [36] have evaluated the issue of burst loss versus packet loss using Pareto distributed traffic generation. When they estimated it against Packet Loss for different burst assembly algorithms, their conclusion was that Burst Loss is not a reliable metric for performance assessment of OBS networks, since Packet Loss probability was lower than Burst Loss. On the contrary, results presented here, obtained through simulation using real tributary IP data packets and realistic burst assembly algorithms, show that Burst Loss is a reliable metric for assessment of Burst Switching networks and that Burst Loss ranges very closely to Packet Loss and to Byte Loss, even when bursts are very heterogeneous in size both packet and byte wise. Also, this study proves that Burst Loss, Packet Loss and Byte Loss are equivalent performance assessment metrics for Burst Switched networks even when the signalling and resource reservation protocols are burst size sensitive, *e.g.* when void filling is performed (*e.g.* the JET protocol). Furthermore, it must also be noted that simulation using real tributary data associated with algorithms that are efficiency concerned,

produce results that do not always agree with the ones obtained by statistically generated data.

4.6. Summary

This chapter was devoted to the study of burst assembly algorithms, their definition and performance using real IPv4 traffic. As real meaningful IPv6 data traces were unavailable at the time of this research, we proceeded to generate as realistically as possible IPv6 traffic based on real IPv4 traffic. While doing this, we concluded that there are benefits to be reaped by using larger IP packet sizes, namely in the reduction of the number of packets that are routed and switched in the core networks.

We also assessed the performance of burst assembly algorithms with real IPv4 traffic and defined a set of metrics to rank the efficiency of these algorithms. We conclude that efficient burst assembly algorithms must be adaptable to traffic changing conditions.

The relevance of OBS network performance metrics was also discussed and assessed. We conclude that, opposed to some authors research, when efficiency concerned algorithms are used on real traffic, burst loss, packet loss and byte loss are equivalent measurements, and propose an explanation to this finding.

In face of the results presented in this chapter and following previous conclusions, we concluded that IPv6 will behave similarly to IPv4, mostly because of the widespread presence of Ethernet transport in access networks.

Chapter 5.

IP Packet Aggregator and Converter Machine Concept

5.1. Introduction

Complementing the previously presented approaches to OBS and to packet aggregation, we now present the IP Packet Aggregator and Converter (IP-PAC) Machine Concept [18, 22], currently in filing process as an European Patent by Nokia Siemens Networks AG. The IP-PAC machine concept may be viewed independently of the OBS network paradigm. Its purpose is to implement burst assembly and disassembly algorithms, in a manner that while allowing backward compatibility with IPv4, also aims to allow forward compatibility with IPvFuture.

In this chapter, we will present the IP-PAC machine concept and motivation, its proposed working procedure and the IP-PAC simulation results. The chapter ends with a summary of the presented topics.

This chapter is partially based on the international patent [18] and on papers [19, 22].

5.2. IPv6 burst transmission motivation

Currently the Internet Protocol (IP), largely used in network environments, is undergoing a shift from version 4 to version 6. This results in a situation in which some sub-networks operate only with IPv4, others with IPv6, while others have the capability of processing both above-mentioned IP packets formats. Naturally, the communication of IPv4 with IPv6 native networks implies application of a device transforming the

sender native format into the receiver native format, in such a way that the resulting data stream can be reinterpreted correctly.

As the share of IPv6 native systems increases in the market [168, 169], there is an increased need to improve inter-communication of these systems with legacy IPv4 sub-networks (which still constitute the majority of deployed systems); a number of solutions have been proposed, which go from tunnelling to re-encapsulation and / or from conversion to translation [169-173].

Yet, to the best of our knowledge, none of the existing solutions addresses the issue of effective utilization of the increased packet capacity of the IPv6 Jumbogram format and in this sense they are limitative and do not take full advantage of the potential embedded in the IPv6 protocol.

On the other hand, the IPv6 protocol is increasing its presence in backbone systems [169, 170, 173-175]. This means that IPv4 traffic still needs to undergo some type of transformation to cross these IPv6 networks.

The machine concept presented in this chapter, named IP-PAC for Internet Protocol Packet Aggregator and Converter, aims at improving the encapsulation and transmission efficiency for IPv4 and IPv6 packets, conveyed over the IPv6 or IPvFuture[2] routing structure. In a general case, this machine concept may be considered a generic IPv4/6 to IPv6/Future aggregation machine, *i.e.*, the machine concept describes a mean to perform burst assembly of IPv4 and IPv6 packets into an IPv6 packet, or from a wider perspective, into a future version of the IP protocol.

The IP-PAC algorithm proposes a combination of the following actions:

1. Aggregation of IPv4/6 packets;
2. Re-encapsulation of the previously aggregated packet conglomerates to IPv6/Future.

[2] From this point onward, we shall refer only to IPv4 to IPv6 burst assembly meaning the burst assembly of IPv4 or IPv6 into IPv6 or IPvFuture.

Using the combination of both above mentioned processes, a series of incoming IPv4/6 packets may be combined into a single IPv6/Future packet, independently of the flow[3] they belong to.

Following the HA algorithm described in section 4.2.3, the applied aggregation method is twofold:

1. A time span is dynamically set between the first and the last packets, which are aggregated into a single transport packet; this time span defines an average packet delay, which all incoming packets are subject to. Moreover, the value of the time span set for the aggregation period will depend on the varying network load of the incoming data flow.
2. A maximum size is dynamically set for the transport packet, depending on the current network conditions, link load of the data flows – in the case of IPv6 packets, the maximum acceptable size is 4 GB (Jumbogram option [12]).

5.3. Machine concept

5.3.1. Placement of an IP-PAC in an IP network

As IP-PAC is a burst assembly machine, it can be used in networks where data transmission may benefit from the statistical multiplexing gains brought by the burst assembly process. In this sense, the working environment for the IP-PAC may be heterogeneous, as discussed in Chapter 4. Figure 71 shows a core network to which several sub-networks connect, some having a native IPv4 format, some having routers and or gateways and a server that connects directly to the core network. IP flow aggregation was addressed by Schlüter in 2000 [176], who concluded that the factor of reduction from host-to-host to source-to-destination flows was primarily dependent on

[3] Data "flow" is a series of strictly time and space localized data packets, sharing the same flow identifier (often referred to as flow label), originating from a certain IP source cluster (group of source IP addresses) and targeted at a particular IP destination address.

the mean number of host-to-host flows that start in a region ***s*** (source) and are destined to a region ***d*** (destination). Schlüter proposes that the best environment to achieve high reduction flow ratios is a backbone routing domain.

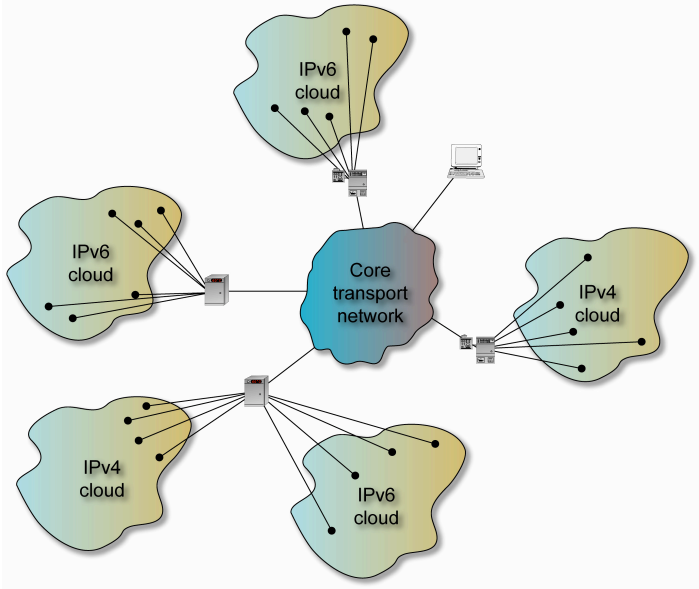


Figure 71 – Schematic representation of a utilization of IP-PAC machines at the edges of a core network, with non-IP-PAC machines.

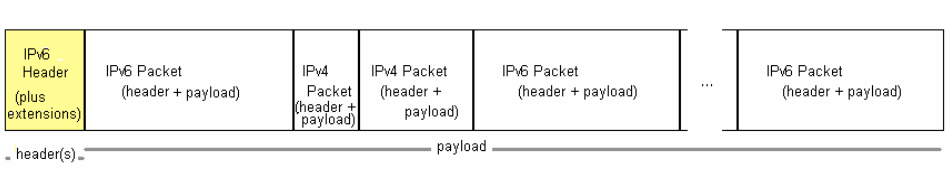


Figure 72 – Scheme of the format of an IP-PAC generated packet.

Packets that are generated by the source IP-PAC machines have a format similar to the sample packet depicted in Figure 72 except in the following situations, for which aggregation should not be performed:

a. if the payload consists of only one packet and this packet is already in the converted format (IPv6 or vFuture), *i.e.*, is already in the format used in the core network;

b. if the destination does not have packet disaggregation capabilities, *e.g.* when the destination is known to be a non-IP-PAC machine.

In this last case, IP-PAC should still queue the outgoing packets as to send them sequentially and thus still profit from the minimization of the core network switching effort brought by the path accommodation phenomenon [107].

In order to enable the IP-PAC machines to recognize an IP packet as an aggregated packet, the header of the aggregated packet (if existent) should have the three higher order bits of the Traffic Class field set (see [10]). We refer to this procedure as IP-PAC flagging and a packet is said to be "IP-PAC flagged" if its three higher order bits in the Traffic Class field of the header are set.

A brief explanation on how communication between IP-PAC and non-IP-PAC machines may be established is described in the following sections.

5.3.2. Communication between an IP-PAC machine and a non-IP-PAC machine

If the source is an IP-PAC machine and the destination is a non-IP-PAC machine, such as a server, a router or a gateway, that will most likely have no disaggregation capabilities, the source packets (non-aggregated but still queued) are sent from the IP-PAC machine sequentially, with a minimum delay.

If the source of the packets is a non-IP-PAC machine but the destination is an IP-PAC machine, then, upon receiving a packet, the IP-PAC will try to find the IP-PAC flag. In this case, as the packet was sent from a machine without aggregation capabilities, it is not IP-PAC flagged and thus the packet is directly interpreted and forwarded to the corresponding exit interface (after query to an internal routing address table).

Figure 73 shows how an IP-PAC machine performs the burst assembly process and forwards the resulting bursts (in this case without encapsulation in an IPv6 / vFuture envelope) to a destination network.

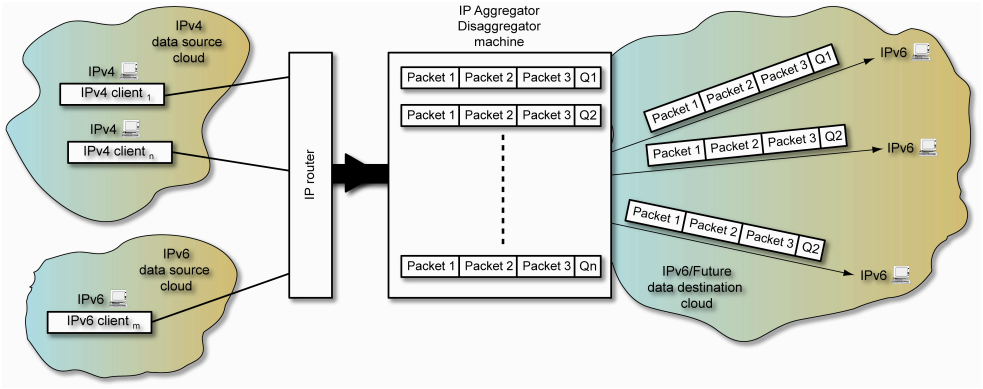


Figure 73 – IP-PAC as gateway depicting several burst assembly queues (Q1 to Qn).

5.3.3. Communication between IP-PAC machines

If an aggregated packet is destined to another IP-PAC machine (see Figure 74), the source machine will transmit a packet similar to the one depicted in Figure 72. The packets transmitted are IP-PAC flagged, unless they fall in one of the conditions mentioned before, *i.e.*, they have not been subject to the addition of an extra header by the IP-PAC sender machine. In this case, the destination address of the packet will be the address of the destination IP-PAC machine.

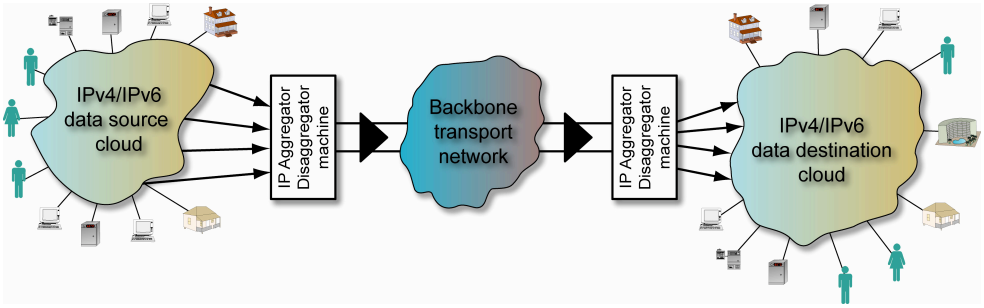


Figure 74 – IP-PAC machines as ingress and egress nodes in a backbone.

5.4. IP-PAC burst assembly algorithm

IP-PAC implements the HA burst assembly algorithm, dynamically adjusting the burst assembly thresholds following the conclusions in [19]. Providing that there is also network feedback as, for example, the predicted network load on the links, the size of the aggregated packet may be set to meet the current conditions of the network status. The size of the packets is set based on the constraints of the network, such as link MTU and path MTU [154], as specified in the IPv6 protocol definition [10]. If possible, the maximum size of a packet may reach the size of a Jumbogram, up to 4 GB. The format and procedure of network feedback, as well as of IP-PAC status communication, can be organized using ICMPv6 [177].

Defining the burst assembly algorithm time threshold as the maximum time a packet is allowed to wait before is sent into the network and assuming that longer bursts increase the statistical multiplexing effect, the dynamic threshold definition algorithm for the IP-PAC hybrid burst assembly algorithm is described as follows:

1) If the threshold limiting the burst assembly process is size threshold and this size threshold is still smaller than the Link/Path MTU for that network, increase size threshold by some ratio;
2) If the threshold limiting the burst assembly process is time threshold and burst size is almost (to some ratio) the burst size threshold, decrease the burst assembly time threshold by some ratio;
3) If the threshold limiting the burst assembly process is the time threshold and burst size is smaller (to some ratio) than the burst size threshold, increase the burst assembly time threshold by some ratio, not exceeding the initially defined time threshold.

This algorithm is depicted in the flowchart in Figure 75. The additional control here proposed provides adjustment for changing traffic conditions. Note that time threshold may increase or decrease, but size threshold only increases up until the limit of the network stated Link or Path Maximum Transmission Unit (MTU). As Path / Link MTU limitations are usually related to limitations in buffering and switching and as in

OBS networks the switching of the burst is done purely in the optical domain and no buffering is foreseen (unless for the case of architectures that use FDLs), it may not make sense to define a Path / Link MTU in OBS. Nevertheless, it can still make sense to define a boundary to the burst size, in particular if we understand that the burst has to be O/E converted and disassembled, thus, electronically buffered. In Figure 75, a set of ratios, *R1*, *R2* and *R3* are defined. These ratios are used to increase the Size Threshold (*R1*), to define the vicinity of the burst size in relation to the Burst Size Threshold (*R2*) and to define the increase and decrease ratio of the Time Threshold (*R3*). Also, in Figure 75, when either Burst Size Threshold or Time Threshold are increased, the ***(A)*** note means that these will augment up until their initially defined maximum values.

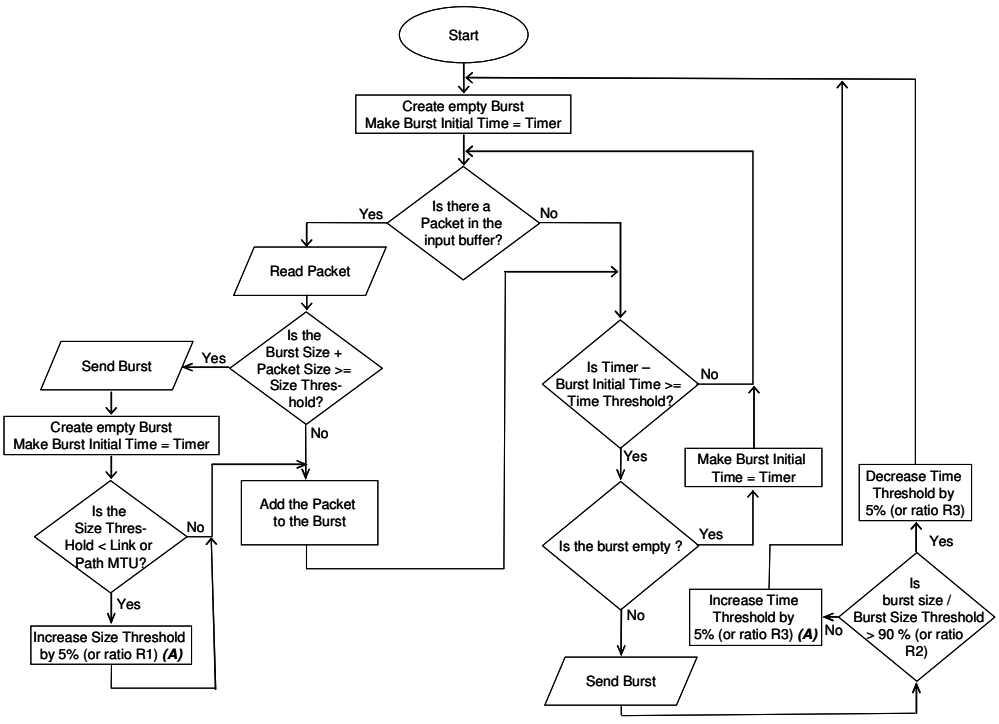


Figure 75 – Generic Hybrid Burst Assembly algorithm with dynamic threshold control.

Figure 72 shows a sample packet generated by an IP-PAC machine. Please note that the initial IP header (shown in yellow or in gray shade) has information about the total length of the packet. The IP-PAC knows that the data that follows the first header

is also an IP header and can thus interpret the following packet, removing it from the payload and forwarding it to its destination. This procedure is done until the payload is empty.

The scenario depicted in Figure 73, showing the existence of one queue per destination is a simplification as more accurately we foresee that IP-PAC may have several queues per destination, in order to cope with Quality of Service issues.

5.5. Simulation and results

In order to assess the performance of the IP-PAC machine concept, a simulator was built. This is a deterministic simulator since it does not perform traffic generation and the bursts are generated according to non-random algorithms. Instead, the input of the simulator was performed using data files that contained real IPv4 packet traces previously recorded by NLANR (see section 4.3). Moreover, taking in considerations the conclusions in Chapter 4, the results are extensible to IPv6.

The basic topology was set as follows: the simulator mimics a burst assembly IP-PAC machine, which receives a number of input streams each corresponding to a data trace file. The number of stream clients, *i.e.*, the number of trace files that feed the simulator is part of the input data provided to the simulator. The burst assembly thresholds and its corresponding adjustment ratios, previously presented in section 5.4, are also defined at the beginning of the simulation. For the performed simulations, including the results here presented, the thresholds were defined as follows: the burst maximum size was set to 9 KB and to 64 KB, the adjustment ratios *R1*, *R2* and *R3* were defined as 5% (see Figure 75) and the time thresholds were defined as 100 μs, 1 ms, 10 ms, 100 ms and 500 ms. The threshold values were defined following the conclusions from Chapter 4 on the performance of burst assembly algorithms.

To further assess the performance of the IP-PAC machine concept, the simulator was tested with the burst assembly thresholds presented in section 4.4. Figure 76 depicts the variation in the packet count introduced by the Internet Protocol Packet Aggregator and Converter (IP-PAC) machine, for these burst assembly scenarios. Considering the

best burst assembly threshold rank discussed in section 4.4, for the "HA T=1000, S=9K" burst assembly scenario, bursts show an average decrease between 75.1% and 90.9% in the packet count, for the studied data traces.

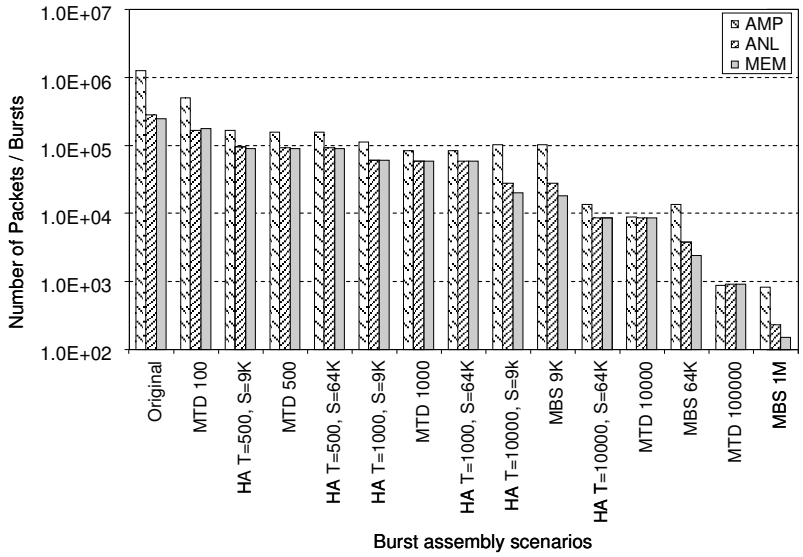


Figure 76 - Packet or burst count per data trace versus various aggregation scenarios (times T are in s and sizes S are in bytes).

Figure 77 to Figure 80 show, respectively, the results for the IP-PAC simulator when fed with a series of seven files containing real IPv4 data traces captured at the MEM site (see section 4.3).

A decrease in the number of transmitted packets leads to significant benefits in terms of statistical multiplexing. Figure 77 shows how the number of data packets traversing the core network (where the edge nodes are IP-PAC machines) decreases dramatically, leading also to a decrease in the number of signalling transmissions and in the number of *acknowledgement*s, with the consequent increase in the overall network structure efficiency. Again, it must be noted here that transmission of a single data packet is always accompanied by a certain amount of transmission overhead, related to negotiation / channel set-up times, during which other transmissions on the same link are deterred. Decreasing the number of data packets traversing the core network structure leads therefore to better resource utilization, since the channel set-up and

negotiation process is executed less frequently. Moreover, the probability of channel occupation also decreases along with the decrease in the number of traversing packets, while the greater packet size (resulting from the aggregation process) only marginally increases the link occupation probability.

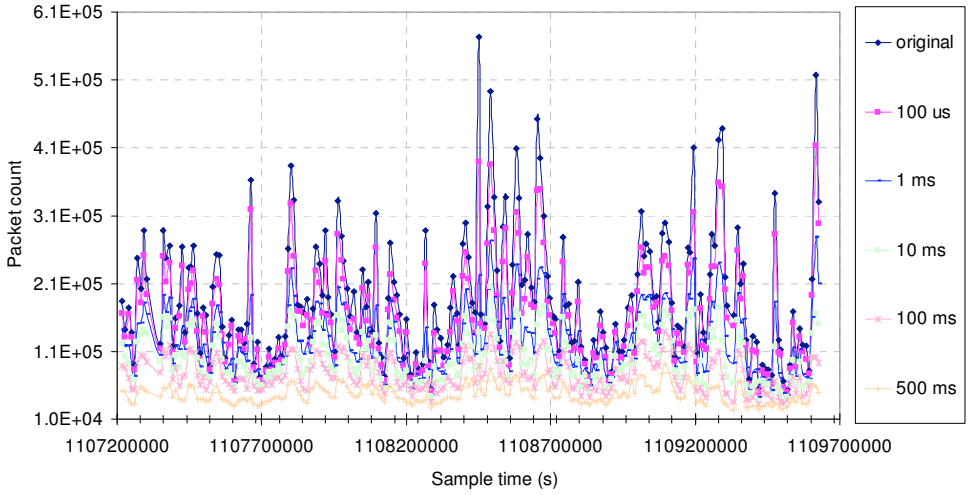


Figure 77 – IP-PAC packet count per data trace for various burst assembly time thresholds (MEM series of data traces).

It is also necessary to mention that depending on the type of traffic (video/voice delay sensitive data flow, general file transfers, newsgroups or email services), the aggregation time threshold should be differentiated. Delay-insensitive services, *e.g.* newsgroups or email servers, might benefit from longer aggregation periods in excess of 100 ms, leading to over 50% decrease in the number of transmitted data packets (see Figure 76) and 4-5% increase in the number of transmitted bytes (on average). Slight increase in the number of transmitted bytes results from the addition of the IPv6 packet header (Jumbogram header in a general case) to the payload consisting of a series of aggregated IPv4/v6 packets – thus the greater the number of average size of the aggregated packets, the smaller the relative overhead. However, the 50% decrease in the number of transmitted packets for 100 ms aggregation scale indicates that the sizes of the aggregated packets are relatively low and their temporal separation is short, resulting in a significant number of original IPv4/v6 packets being aggregated into a single transmission packet (see Figure 78).

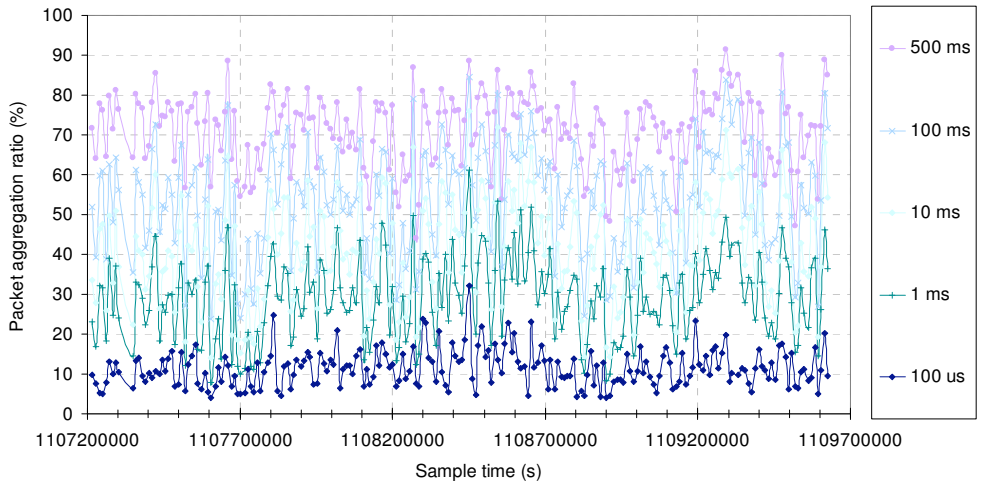


Figure 78 – IP-PAC packet compression ratio for various burst assembly time thresholds (MEM series of data traces).

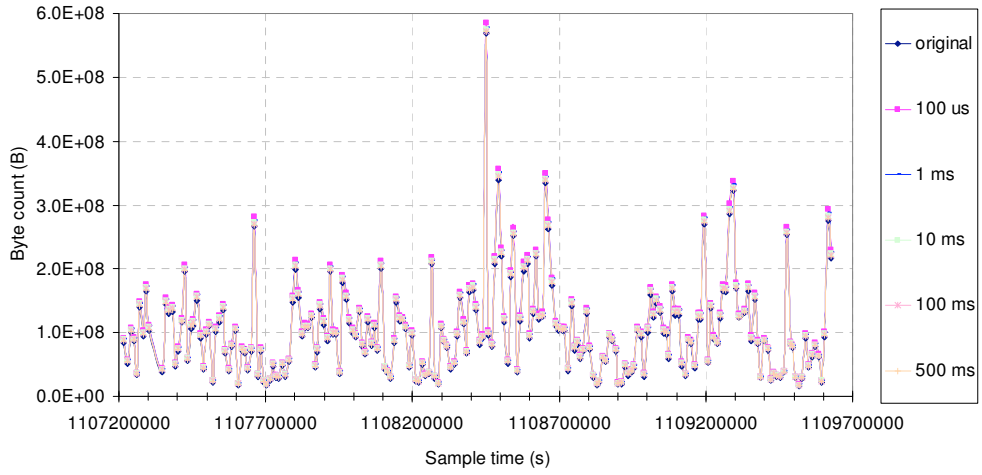


Figure 79 – IP-PAC number of bytes in burst transmissions for various burst assembly time thresholds (MEM series of data traces).

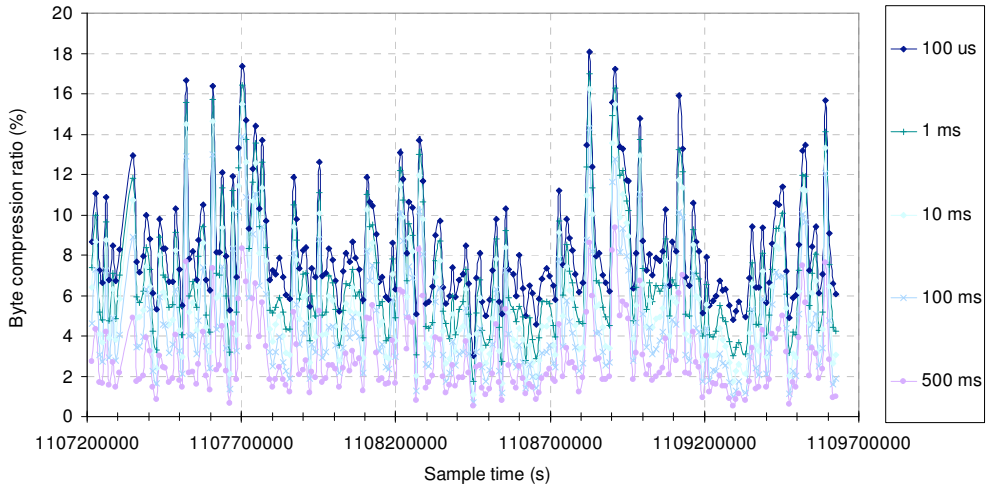


Figure 80 – IP-PAC byte compression ratio for various burst assembly time thresholds (MEM series of data traces).

As anticipated, the application of the IP-PAC transmission scheme leads to an increase in the number of transmitted bytes as Figure 79 shows, because of the added IPv6 headers and extensions. The overhead imposed by the re-encapsulation may vary between 2-3%, for 500 ms threshold in MTD and 10 – 12 %, for 500 s threshold in MTD. However, there is still place for a reduction on the occupancy time of the link, since the assembled packets (bursts) require only one framing space, as opposed to a framing space required for each of the constituent packets, thus resulting in the ability to transmit more user data over the same time period, as seen in Figure 80.

Further simulation was performed to assess IP-PAC ability to address bottleneck problems. All tests and simulations were carried out using native IPv4 data packets as input data flows. The conducted tests included varying (static) aggregation time and size thresholds as well as dynamic allocation of the threshold values.

To address the issue of proving bottleneck resolution, a new simulator was built. Channel bottleneck is resolved by definition, as the aggregated packets are transmitted only when the transmission channel is free.

Using simulation, we tried to assess the probability of occurrence of such bottlenecks – or better saying – increased delay in transmission of the aggregated

packets. Thus, whenever a bottleneck occurs, we can expect an aggregated packet transmission time longer than the pre-defined aggregation time.

Figure 81 shows the simulator window after a simulation was completed. The left section of the windows shows the values loaded for the simulation, while the right section shows two tabs, each containing the graphs depicting the behaviour of the traffic simulated.

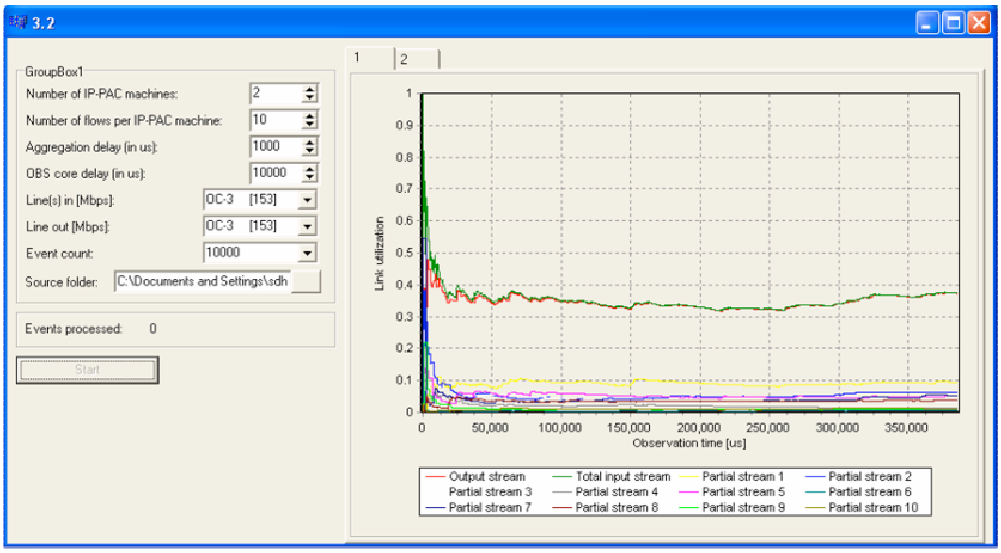


Figure 81 – Simulator version 3.2 after running a simulation.

The tested transmission protocols and physical media are:

1. IP over a 5 hop linear OBS network, using the JIT protocol, at 1 Gb/s
2. IP over a 5 hop linear OBS network, using the JIT protocol, at 10 Gb/s

Other media were testes, namely IP over Ethernet at 1 Gb/s and IP over SONET OC 48 (2.448 Gb/s), but these results are not shown here as they fall out of the scope of this thesis. Each of these networks was simulated using packet traces from different sources, namely the ones obtained in NLANR (see section 4.3.1).

Table 10 – Simulation results for an OBS network with Aggregated Packets Statistics (Transmission).

Simulation Number	Packet Size Minimum	Packet Size Average	Packet Size Maximum	TX Burst Count	Burst Packet Count Minimum	Burst Packet Count Average	Burst Packet Count Maximum	Burst Size Minimum [B]	Burst Size Average [B]	Burst Size Maximum [B]	Burst Packet Delay Minimum [s]	Burst Packet Delay Average [s]	Burst Packet Delay Maximum [s]
1	75	770.30	19 660	381	3	11.7471	32	329	9 046.5376	58 049	0	543.1475	1 850
2	69	708.85	19 424	389	3	10.1576	21	461	7 213.7066	25 958	0	549.8282	1 000
3	69	815.70	1 966	481	2	9.6682	23	321	7 880.2069	38 926	0	552.9473	1 000
4	76	420.69	14 116	469	2	7.3580	16	252	3 090.0782	16 086	1	517.5579	1 000
5	69	619.27	19 424	344	2	10.7671	21	211	6 654.8884	31 693	1	555.5086	1 000
6	31	849.93	19 660	325	5	15.1176	46	655	12 825.709	66 126	0	548.3873	2 446
7	69	728.75	19 423	345	3	10.9276	31	516	7 977.0984	41 415	1	548.7667	1 061
8	36	864.05	19 660	327	5	14.7080	54	547	12 726.2092	68 992	1	558.4284	2 567
9	69	773.88	19 660	383	4	12.6769	33	613	9 811.7679	56 854	0	544.6679	1 824
10	76	618.04	16 700	388	2	9.1971	21	332	5 688.2986	32 214	0	554.1382	1 000
11	71	703.10	19 424	370	3	9.3773	22	366	6 597.6565	38 365	0	546.8784	1 000
12	36	593.19	15 460	386	4	12.9884	24	621	7 716.3679	26 210	0	544.4179	1 000
13	75	847.11	19 660	373	3	11.1665	31	401	9 470.2568	56 496	1	549.1982	1 743
14	69	595.65	19 424	388	1	11.0278	22	88	6 574.7579	31 693	0	543.1482	1 000
15	69	762.81	19 660	430	2	9.3069	19	264	1 073.6982	32 688	0	554.1769	1 000
16	75	691.51	17 852	431	3	9.8865	19	300	6 847.8879	30 777	0	555.6683	1 000
17	70	1 000.21	19 660	99	1	5.4965	14	92	5 573.783	22 213	7	629.6077	1 000

Table 11 – Simulation results for an OBS network with Aggregated Packets Statistics (Reception)

Simulation Number	Packet Size Minimum	Packet Size Average	Packet Size Maximum	RX Burst Count	Burst Packet Count Minimum	Burst Packet Count Average	Burst Packet Count Maximum	Burst Size Minimum [B]	Burst Size Average [B]	Burst Size Maximum [B]	Burst Packet Delay Minimum [µs]	Burst Packet Delay Average [s]	Burst Packet Delay Maximum [s]
1	75	770,30	19.66	380	3	10,1865	21	485	716,5570	25.958	1.925	5.593,6069	13.304
2	69	708,85	19.424	372	3	11,7673	32	329	774,5379	58.049	1.749	6.383,2690	31.871
3	69	815,70	1.966	460	2	7,3479	16	525	422,0282	16.086	1.060	4.191,3665	9.773
4	76	420,69	14.116	471	2	9,6754	23	321	812,7567	38.926	1.569	5.350,7066	14.260
5	69	619,27	19.424	316	5	15,1048	46	655	846,5665	66.126	2.535	8.292,1079	44.849
6	31	849,93	19.66	335	2	10,7348	21	211	623,7483	31.693	1.576	5.945,8865	15.416
7	69	728,75	19.423	319	5	14,7053	51	547	862,4765	68.992	2.913	8.218,6790	44.813
8	36	864,05	19.66	336	3	10,9879	31	516	731,4766	41.415	1.868	6.021,4589	17.535
9	69	773,88	19.66	380	2	9,1686	21	332	622,9789	32.214	1.063	5.090,4065	12.715
10	76	618,04	16.7	374	4	12,7073	33	613	718,5068	56.854	2.223	6.918,4167	23.730
11	71	703,10	19.424	377	4	12,9765	24	621	598,8979	26.21	2.272	7.064,6572	14.640
12	36	593,19	15.46	361	3	9,3883	22	366	704,6382	38.365	1.521	5.112,8673	11.484
13	75	847,11	19.66	378	1	11,0467	22	379	603,5173	31.693	1	6.002,1577	15.297
14	69	595,65	19.424	364	3	11,2179	31	401	855,2565	56.496	1.378	6.156,4265	24.255
15	69	762,81	19.66	422	3	9,8883	19	300	702,7466	3.077	1.347	5.492,6677	10.927
16	75	691,51	17.852	421	2	9,3184	19	264	763,6869	32.688	1.219	5.161,2479	12.014
17	70	1.000,21	19.66	92	1	5,6285	14	92	1.031,8076	22.213	1	3.542,6382	9.546

For each simulation, we defined a network with two edge nodes with a particular link capacity. The first scenario consists of an Ethernet network with two nodes, and scenarios two and three consist of an OBS network with two nodes edge nodes and three core nodes. We then fed each of the considered nodes with 10 data flows obtained from real traffic samples, *i.e.*, data packets were not generated randomly. Simulator times are expressed in microseconds. The simulator also builds several log files, tracing the flow of the bursts. One of these files is used to store overall simulation statistics, depicted in Table 10 and Table 11. As we see in the chart in Figure 81, we have an overload of the link in machine 1 at the time near 0, although this does not happen in all performed simulations.

As expected, when a bottleneck occurs, as seen in the chart in Figure 81, the total delay of the bottlenecked burst (or aggregated packet) is bigger than the defined aggregation time. A bottlenecked burst is delayed enough time to allow the channel to support its transmission. We can see in Table 10 and Table 11 that, although we collected 10 SONET OC3 channels into a single OC3 channel, with real traffic conditions, transmission bottlenecking is observable in only about 36% of the simulations.

5.6. Discussion and Conclusion

The results obtained by simulation allow us to number the advantages of the IP-PAC machine concept, as follows:

1. There are statistical advantages due to packet aggregation and an overall decrease in the amount of packets traversing the core network structure at a given moment of time [75, 178-183]. Such a decrease in the number of packets traversing the core network results in:
    a. Decrease in the transmission overhead, since the transmission of a single data packet (regardless of its size) is always related with transmission of the signalling messages (channel set-up, resource reservation) as well as possible re-transmission events in the case of the concurrent attempts to access the transmission media. Decreasing

the number of traversing packets results in fewer signalling messages as well as decrease in the channel reservation overhead. Also, the number of framing and preamble spaces is highly reduced because the packets are now part of a single mega-packet payload.

b. Decrease in the channel blocking probability: smaller packets traversing the transmission channel result in increased channel blocking probability, since their temporal distribution is highly chaotic. A new packet to be transmitted has therefore a limited possibility of gaining access to the transmission medium. Therefore, decreasing the number of packets attempting channel access at a given moment of time increases the probability of the given packet actually gaining access to the link.

2. Although there is an increase in the byte count of the assembled packets that are encapsulated inside an IPv6/Future envelope with the IP-PAC to IP-PAC transmission scenario, there are routing benefits that can be reaped inside the core network, because the tributary packets are viewed as a single IPv6/Future packet. In order to diminish this overhead, it is expected that the IP-PAC machine must have a feedback loop, monitoring the aggregation process efficiency and allowing for simple packet forwarding in case of low (for example below 5%) aggregation ratio. This optimization strategy falls in the scope of future work.

3. All IPv4 packets are inherently transported inside IPv6 packets and thus the core network structure does not have to maintain compatibility with the older IPv4 standard (*i.e.* Dual Stack implementations), thereby lowering the equipment costs and allowing for application of more optimized and thus efficient IPv6/Future compatible hardware, or, from a more comprehensive point of view, all IPv4 and IPv6 packets are converted into IPv6 and/or vFuture packets, being this issue dependent only on the nature of the IP-PAC installed interfaces.

4. Since both time and size thresholds employed in the aggregation and re-encapsulation mechanism are adjustable, it is possible to introduce dynamic link management in the aggregation/re-encapsulation machine in such a way that the aggregation parameters depend on the current state of the data link (number of data flows, self-similarity level, link load, bandwidth utilization and request, etc.),

5. The proposed IP-PAC machine is targeted at the core network structures, where the significant advantages from application of statistical multiplexing produce measurable improvements in terms of decrease in packet loss and path blocking probabilities. One of the switching paradigms that benefits from the proposed concept is Optical Burst Switching. When the number of traversing packets decreases, so does the transmission overhead on relaying a certain amount of bytes from source to destination, since the aggregated packet needs only one path set-up event after which a continuous data stream is transmitted from the source to the destination node.

6. Attenuation of bottleneck problems, since these occur only when the sum of the loads of input channels is bigger than the capacity of the output link, for a duration time approximately close to the aggregation time, or when buffering inside the IP-PAC machine is not able to store the received data. Performed simulations show that ten real traffic OC3 links where successfully aggregated into a single OC3 output link, with a reasonably low burst delay in some simulations.

There are also a few possible disadvantages, namely:

1. Increased average packet delay, when considering the end-to-end transmission time, since packets are subject to additional processing when entering and leaving the aggregation/re-encapsulation nodes; in this case, a criteriously chosen aggregation time, taking into account the type of traffic received, guarantees that this delay does not introduce any significant degradation of the network performance.

2. Since packet loss occurs at each network level, loss of a single aggregated packet means loss of a few aggregated and encapsulated packets in the best case, or hundreds in the case of a Jumbogram and thus packet loss issues in the core transport fabrics become crucial. Our initial results indicate that packet loss problems should be mitigated by application of buffer space within the IP-PAC machines, capable of storing at least the last data transport packet directed at the given target IP-PAC machine (destination cluster). Upon proper packet reception at the destination node, the *acknowledgement* message transmitted towards the destination allows for verification of transmission success and for freeing the utilized buffer resources. It is expected though that packet loss issues become critical only above a certain aggregation threshold, where the number of aggregated original IP data packets becomes significant (packet aggregation ratio of 25% and above, for example). Therefore, the buffering and signalling service should be dynamically available only for certain aggregation thresholds, in order to decrease the transmission overhead for short aggregation times below 10 ms, where the amount of required signalling messages would be significant.

5.7. Summary

This chapter presented the machine concept for the IP Packet Aggregator and Converter – IP-PAC. The machine working environment was defined in this chapter, as well as the working protocols for the communication of an IP-PAC machine with non-IP-PAC machines. A new burst assembly algorithm was presented and discussed. The performance of the new machine concept and its algorithms was assessed through simulation and the results were presented, including the impact of the use of such machine in the resolution of bottlenecking problems.

Finally this chapter presents the advantages and drawbacks of this new concept and the most relevant conclusions.

Chapter 6.

Architecture and Performance Evaluation of Common Control Channel Optical Burst Switched Networks

6.1. Introduction

In OBS networks the traffic management decisions are mostly performed in the ingress nodes, as the core nodes are expected to be simple. When an ingress edge node sends a burst into the network, the corresponding control packet (CP) that precedes the burst already includes information on the proposed network path that the burst and its CP must travel. This restriction on the network design causes the information in the nodes to be used only locally, meaning that the network as a whole system does not benefit from the information and experiences of individual core and edge nodes. Other proposed architectures, such as the one reported in [62], use a centralized management model to optimize the utilization of the network information. The goal of this chapter is to present a new OBS network architecture, named Common Control Channel Optical Burst Switched Networks – C^3-OBS, registered as International Patent by Siemens AG. This architecture allows the network to use and share the information carried in each CP in each node to outperform both the TAG OBS, by diminishing the burst loss probability and the TAW OBS, by decreasing the time needed to get an acknowledgement of resource reservation. The architecture presented hereby solves both aforementioned problems by implementing the concept of "network-state-awareness" using a common control channel.

This chapter is devoted to the discussion, evaluation and comparison of the performance of this new architecture with OBS networks. The architecture for a C^3-OBS network and its node functional scheme are presented and discussed, as well as the algorithms implemented in the Signalling Engine. New problems related to this architecture are identified and solutions are proposed. The assessment of the performance of C^3-OBS networks for several regular and real topologies, as well as the comparison with the assessment on the performance of identical OBS networks is also presented. The estimated delay for a burst travelling network topologies in C^3-OBS and in OBS is assessed, and a summary of the discussed topics concludes this chapter.

This chapter is partially based on the international patent [23] and on papers [14, 15, 24-27, 140].

6.2. Architecture of a C^3-OBS network

6.2.1. Architecture of core nodes

In TAW or TAG OBS, the control channel is used much in the same way as an electronic data channel, in the sense that a given setup message, encapsulated in a CP, must visit and be interpreted by all the nodes in its path. For example, if a CP has its source in node 1 and is destined to node 3, it will not reach node 3 until node 2 has received, interpreted and decided to forward it (see Figure 82).

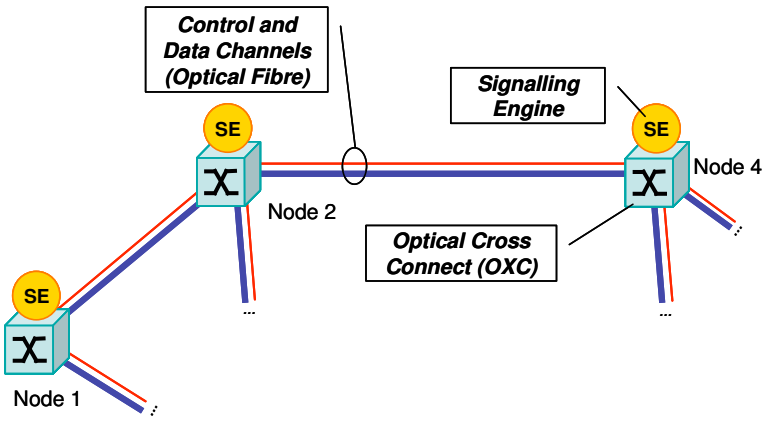


Figure 82 – Schematic representation of an OBS architecture.

This means that if a burst has to travel through *n* nodes, the setup message is to be sent, received and converted in the O/E/O manner and finally resent, *n* times. The Signalling Engine (SE) will process each setup message and try to reserve network resources such as a given output channel at the output fibre, wavelength conversion facilities and OXC switching capability for the each corresponding burst, in view of its estimated arrival and route time for the node in question.

Figure 10 shows the case for the JIT [90] protocol (TAG OBS). It depicts how the time that a setup message takes to travel from the ingress to the egress node is calculated as shown in equation (1).

There is no assumption made as to the time the control packet takes to cross each link, as it is assumed that a node only issues a CP at a time on a particular control channel and also that the data burst travels the same link at the same speed as the CP. The time needed to make the Initial Offset, as shown in Figure 10 is established by equation (1).

Currently, T_{OXC} is around some few milliseconds and T_{Setup} is near some tens of microseconds [55] and although the trend in the industry is to achieve shorter setup and configuration times for both the signalling engine and the OXC, the transmission of bursts end-to-end may still suffer significantly in a network with a high burst loss ratio, thus degrading the end user experienced quality of service of the network.

The proposed new architecture features a control channel that is shared by all network nodes. By using a common control channel, each of the nodes in the network receives the setup information with the delay inherent to the propagation of the signal in the fibre, thus allowing them to locally keep a near synchronized network model of the network. The common control channel is deployed throughout the network topology as a loopless tree, detailed in this chapter.

In OBS, each node keeps a simple data structure to record the resource occupancy status. This structure is updated each time a CP arrives and the maintenance of this structure is performed accordingly with the resource reservation protocol specifications, namely, in terms of void-filling or wavelength conversion policies. In

C^3-OBS the concept of this structure is extended, as now each node keeps a structure that not only reflects the local reservation requests of the node, but also the requests for the other nodes in the network. We call this extended structure the Local Network Model (LNM), which is updated with the information received in all the CPs, even those CPs which do not directly concern that node. Through the LNM, each node can also keep track of the network status and performance because it can monitor the whole resource reservation of the network. Thus, by querying its LNM, it is easy for each node to plan the path and or departure time of a burst, or to apply advanced contention resolution techniques. As each of the nodes keeps track of the resource reservation requests in the network through the maintenance of the information in the LNM, each node is said to be aware of the state of the network. The LNM is detailed forward in section 6.2.3.

Figure 83 shows the operating principle for the Common Control Channel OBS (C^3-OBS) using an immediate reservation protocol. Other protocols, such as delayed reservation protocols, may also be implemented.

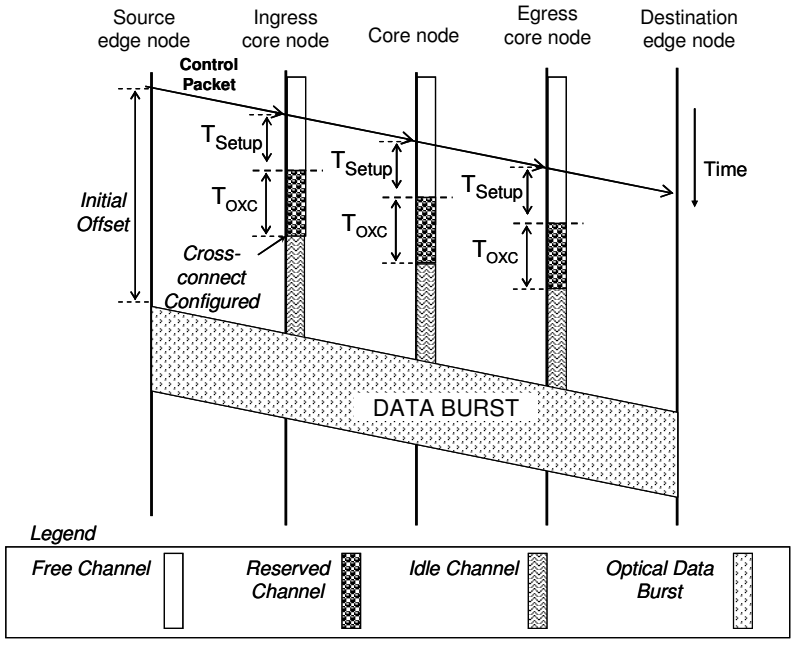


Figure 83 – Example of a signalling and burst transmission in a C^3-OBS network.

It can be seen from Figure 83 that:

$$T_{Offset} = T_{Setup} + T_{OXC} \tag{24}$$

independently of the number of nodes the burst must traverse, and T_{Offset} is $(n-1)$. T_{Setup} smaller than the one given by equation (1). This decrease in T_{Offset} is calculated for ideal conditions, that is, when the control channel is not overloaded and the node sending the CP finds the control channel free; otherwise, the CP has to be delayed using a collision detection / avoidance mechanism enough time as to be relayed safely into the control channel. The gain obtained by comparing equations (1) and (24) is ideal and in fact the average time a burst takes to cross a C^3-OBS network is slightly larger than its equivalent for OBS as we will see in section 6.5.

To allow the implementation of the propagation of the control channel, a new architecture for the core node is presented. Figure 84 shows the C^3-OBS OXC node block architecture. The array of optical switches associated to each incoming fibre link implements the common control channel topology. Also visible is the connection exiting the E/O conversion module that splits the signal and allows for the broadcast of new control packets to all output fibres. If the array of switches does not block the control channel, it is also visible that incoming CPs are replicated (the optical signal is split), being one copy delivered to the O/E module for interpretation and update of the LNM in the signalling engine and other copy forwarded directly to the output fibres. Also noteworthy is the fact that, considering that no blocking is applied, the CPs entering from Input Fibre 1 are split and merged with all other CPs entering from other input fibres and forwarded to all output fibres. Additionally, it is easy to detect time concurrent CPs in the input fibres, thus allowing them to be rescheduled (in the electronic plane, where they are interpreted and may be easily buffered) into the outgoing control channel without collision.

The Signalling Engine (SE) has several functions: it converts Control Packets (CPs) from optical to electronic form, interpreters it and composes the new CPs that may be necessary, managing possible overlaps from the various input control channels, doing its posterior electronic to optical form conversion to the selected output fibres.

Complementarily to this, it also controls the array of blockers, whose function is to implement the tree in the control channel. The SE also runs the scheduling and resource reservation algorithms (see section 6.2.4) and interfaces with the burst assembly machines, although this interface is not depicted here (see section 6.3).

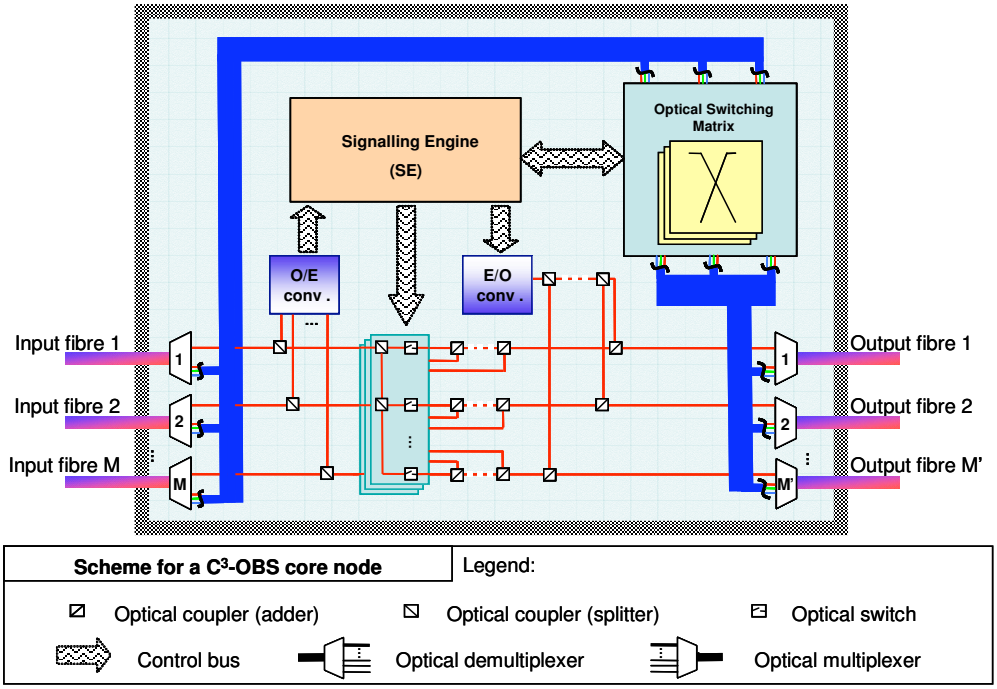


Figure 84 – Schematic illustration of the block architecture for the C^3-OBS core node.

6.2.2. Common control channel

The control channel conveys all the messages of the OBS network, including burst resource reservation, inter-node messages (for example discovery and status messages) and so forth. Although the initial physical topology of both the control and the data channels is the same because the control channel is a particular lambda channel in the fibre that connects two nodes, the logical topology for the common control channel does not necessarily coincide with the logical topology of the data channels. Considering a simple bus topology, the implementation of the Common Control

Channel means the direct connection of the control channel in each input fibre to the control channel output fibre. In a tree topology, the implementation of the Common Control Channel implies the merge of the control channels in the input fibres and also the splitting the internal control channel of the node to each of the control channels of the output fibres. In more complex topologies such as the ones that include rings, the broadcast of the CPs and the need to avoid endless CP feedback through the Common Control Channel forces the control channel to be shaped like a broadcast tree. This common control channel topology shaping procedure is implemented by a new OXC node architecture that allows for a configurable "hard-wired" setup of the merge and split of the control channel entries and exits in the OXC machine, as static as possible. The initial configuration of the control channel is achieved through the operation of a special spanning tree algorithm upon network start-up or reconfiguration phases, either by causing loops in the control channel to be opened or allowing non loop links to be merged. This special spanning tree algorithm also tries to minimize the maximum the number of nodes a CP has to cross to reach all then nodes in the network, named network control diameter (NCD) and is subject of future research. All the control channel topologies depicted here were defined individually.

Figure 85 shows an illustration of a network topology, depicting data channel and control channel topologies separately. Note that the special spanning tree algorithm disconnects the control channel at its entry inside the node, not at its exit from the node. This allows CPs to reach all the nodes in a shorter time span, and is implemented in the node architecture by sets of optical switches shown in the diagram in Figure 84. For a given topology, there are generally several possible control channel topologies. In particular, for the data channel topology shown in Figure 85, a number of different control topologies can be defined. Note that the special spanning tree algorithm disconnects the control channel at its entry inside the node, not at its exit from the node.

Since the control channel is shared, the network control functions may be distributed in the network structure, allowing core and edge nodes to decide on the traffic flow by querying a constantly updated database that holds the reservation schedules for all nodes (see section 6.2.3).

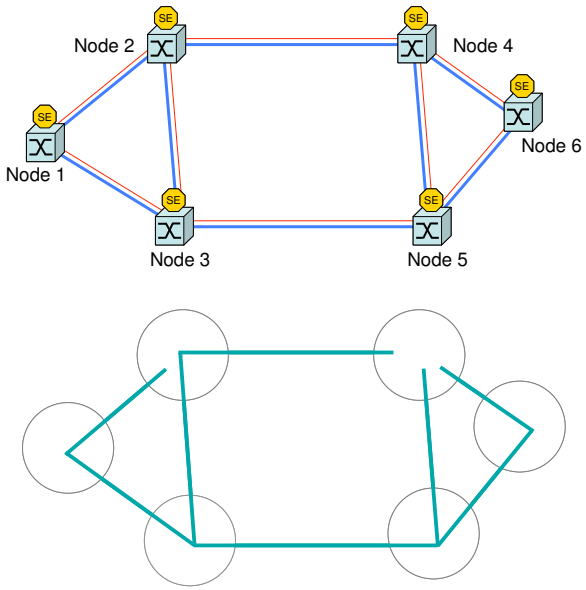


Figure 85 – Schematic illustration of a mesh network with 6 nodes and 8 links (upper scheme) and one possible control channel topology (lower scheme).

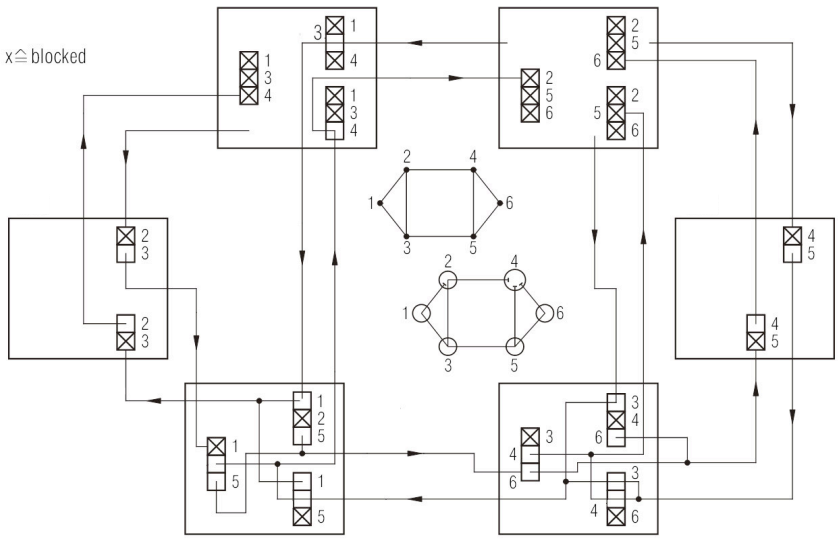


Figure 86 – Schematic illustration of C^3-OBS OXC in a topology showing open and closed switches for the control channel.

Figure 86 shows how the arrays of switches of six C^3-OBS OXCs are configured to implement a common control channel topology for the sample topology depicted in Figure 85. For example, it can be seen how the input fibre in node 2 from node 1 finds all its switches closed, to prevent the control channel to connect further on, although the control channel of the fibre incoming from node 3 is allowed to connect to the control channel of the fibre exiting to node 4, because its corresponding switch is open (see Figure 85 for a scheme of the control channel topology).

From the analysis of Figure 85 and Figure 84 follows that a node may receive duplicate CPs: for example, assuming the delay of the links in the network is not negligible, when node 1 issues a CP node 2 will receive two copies of this CP and node 4 will receive three copies of this CP. Thus, a node must discard the CPs that are known to have already been processed and used to update the LNM. As the control channel of each fibre is O/E converted and interpreted individually (see Figure 84), if each CP is uniquely referenced, *e.g.* using a unique ID, CPs duplication detection and discard is immediate. This ID can be a simple counter field, that when used with the source address allows a node to compare it with its own *source node address* + *counter tag* record and decide whether this message is relevant or not.

Using the example depicted in Figure 85 and also observing Figure 86, if node 1 sends a CP into the network, it can be seen that node 2 and node 3 will receive it at the same time (assuming equally long links), despite the fact that the control channel is interrupted inside node 2.

The propagation of the CPs is depicted in Figure 87 for the sample network presented in Figure 85 and considering a CP issued by node 1; in this figure, active links are depicted as dashed lines. From Figure 87 it is clear that after time t_3 all the nodes in the network have received the information in the CP issued by node 1, although the propagation is finished only at time t_4. It can also be seen that, as mentioned before, several nodes receive duplicate CPs – these are discarded at reception if their ID tag field is outdated. We define the Network Control Diameter (NCD) as the largest number of hops a control packet may experience until it is replicated throughout the network.

For this topology the NCD is 3, so at time t_3 all the nodes have received all CP, independently of its source node.

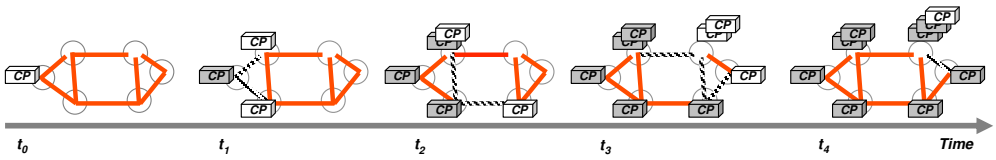


Figure 87 – Sequence of propagation of a control packet in a sample network for C^3-OBS.

In Figure 85 we can see how the control channel fully overlays the link topology, *i.e.*, the control channel spans over all the links. There is a different approach to this setup, considering Hamiltonian or semi-Hamiltonian circuits or paths [184] for the control channel. This approach considers that on a specific network topology, not all of the links contain a control channel, since the CPs that travel in that channel will still be able to reach all the nodes. From a purely static point of view there is an advantage to the use of Hamiltonian paths – the duplication of the CPs in the nodes may be greatly reduced (Hamiltonian circuit or semi-Hamiltonian topologies) or eliminated (Hamiltonian path). There is also a disadvantage, as with a Hamiltonian topology the Well Informed Time t_{WI} for the network will increase (presented in section 6.2.5), thus augmenting the number of Concurrent Control Packet events (Concurrent Control Packet events are discussed in section 6.2.5 - Well informed nodes and degree of network awareness of nodes).

Using a Hamiltonian path or circuit to the control channel topology means that the control topologies links do not necessarily overlay all the physical data links of the network. As a sample of this, Figure 88 shows the control topology for the sample network in Figure 85 (upper scheme) as a Hamiltonian path. Figure 89 shows how these simpler control topologies do not generate redundant CPs, although they affect the overall CP propagation time for the sample network topology in Figure 85, as in this case, the maximum number of hops between any two nodes raises from 3 to 5, *i.e.*, the NCD is 5, being this the number of hops for the CP to travel from node 6 to node 4 and *vice-versa*. The study of the performance of Hamiltonian or semi-Hamiltonian Control Channel topologies (paths or circuits) for more complex networks, such as the European

Optical Network [130] (EON), will be addressed in future work. The results presented in section 6.6 consider full-layout control channel topologies.

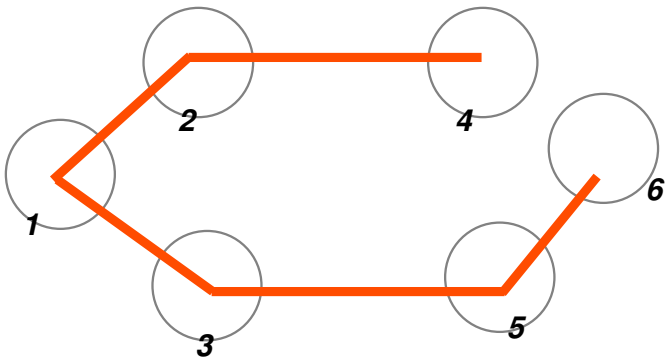


Figure 88 – Sample Hamiltonian control channel topology for the sample network in Figure 85 for C^3-OBS.

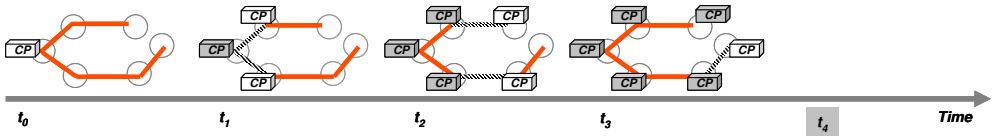


Figure 89 – Sequence of propagation of a control packet issued by node 1 for the sample network in Figure 85 for C^3-OBS using the Hamiltonian control channel topology in Figure 88.

6.2.3. The Local Network Model

The C^3-OBS new architecture features a control channel that is common and shared by all network nodes. By using a common control channel, each node in the network receives all the signalling information coded in the CPs, with the delay inherent to the propagation of the signal in the fibre, thus allowing each one of the nodes to keep a closely synchronized Local Network Model (LNM).

The Local Network Model may be view as an extension of existing OBS local resource reservation status database [44, 45]. The LNM data model, whose complexity depends on the complexity of the signalling protocol, is extended to allow the recording of the reservation status for all the nodes in the network. The update of the LNM is done

with the information received in the CPs, propagated automatically by the common control channel as described in section 6.2.2. Through the LNM, each node can also keep track of the network condition and performance as well as of the resources reservation status throughout the network, thus allowing each node to assess the predicable availability of the resources of its neighbours and decide for a particular burst, for instance, what will be its path or its next hop on the way to the egress node. Hence, each node is said to be aware of the state of the network through the LNM.

The complexity of the local database is directly related to the complexity of the signalling protocol, *i.e.*, for JIT, the local database can be reduced to an array of time values for each channel (wavelength) per output fibre, while the implementation of void filling signalling protocols implies the operation of a list with reservation times for each wavelength in the output fibres.

At an ingress node, each burst ingress request will trigger a query on the LNM. Upon this request, the algorithm managing the LNM will try to find the least cost available path, being cost a synonym of path length or path with least configuration effort or whichever combination of criteria is used to define the most suitable path for a burst.

The degree of complexity of the operation of the LNM may be minimized using two rules – the first one is, upon request for a route for a new burst, to limit the depth of searches of available paths between two nodes to a limited small number. Although this rule may be seen as over restrictive, the higher the number of searchable paths, the longer the paths will be, thus bursts will end up crossing more nodes, demanding more routing operations and occupying more network resources. More important than the number of possible paths is their separation degree, *i.e.*, the least two paths have in common, the more homogeneously the traffic will be distributed on the network links. For the results presented in this chapter we used from 1 to 4 possible paths, following an algorithm that returned the *k-shortest paths* [185, 186]. The second rule is to limit the extent of the analysis of availability of the resources by limiting the possible departure time of the burst to a minimum time; this time is termed “Departure Horizon” and was kept to zero in most of the simulations, *i.e.*, either the resources were available at the

time the burst was ready and thus reservation was successful, or if the resources were not available at that instant, the burst was dropped, although the resources might be available the next instant. With a larger Departure Horizon, the algorithm managing the LNM, termed "Travel Agency" and presented forward in this chapter, tries to schedule the burst as soon as any route allows it, up until the Departure Horizon time value. The setting of both the number of paths for search to a small value and the departure horizon to zero (or close to zero) limits the depth of the search on control data structures in the LNM. In section 6.6 we present simulation results with different Departure Horizon times and with different number of search paths.

Unlike the DWR-OBS approach, C^3-OBS may implement a variety of resource reservation protocols, either using void filling or not, either using immediate or delayed reservation. As all the decisions are made locally, the unique restriction in choosing a reservation protocol is the direct relation between its complexity and the size and complexity of operation of the LNM. In our simulations, a JIT-like reservation scheme was used, *i.e.* immediate resource reservation and no attempted void-filling.

6.2.4. The Travel Agency Algorithm

Travel Agency is the name we chose for the algorithm designed to manage the Signalling Engine (SE) in the Optical Cross Connect (OXC), briefly presented in Chapter 3. As opposed to the foreseeable functioning of an OBS network, where the ingress node keeps a routing table with one or more (few) routes for each ingress-egress pair of addresses and performs static source routing, or in some cases more complex schemes, *e.g.* as Fixed Alternate Routing [128], the Travel Agency Algorithm (TAA) is an adaptation of the fixed source routing algorithms, with the difference that in this case, each burst ingress request receives a response with a route that is calculated for its particular characteristics, or if no route is available, it receives the information of no availability of resources. If the burst cannot ingress the network, this means that the x least cost paths the TAA is authorized to scan are not available for the burst transit at the time intended for its departure, added with the margin of the defined Departure Horizon. In this case, the burst may be kept in the client node, which in turn may decide to try its re-ingress or its loss.

On the other hand, if a burst is allowed to enter the network, its travel path may be different from the one that was assigned to the previous burst with the same priority and ingress-egress addresses and it will only drop if there is a Concurrent Control Packet scenario, created either naturally of forced by the ingress of some higher priority burst.

In the TAA no assumption is made about the resource reservation protocol utilized in the network – TAA will handle the LNM accordingly to the chosen resource reservation protocol. Its main feature is that the routing table is assessed for network resource availability each time a burst ingress request is received.

For each burst ingress request, the TAA assesses the availability of the network resources for the desired departure time, for the k shortest paths [185, 186]. The first path whose resources are available for the ingress request is then reserved for that burst, a process that results in the creation the correspondent CP of the burst. If a non-zero Departure Horizon is allowed, the k shortest paths are ranked in order of its earlier availability and the first one is tagged for reservation and used in the CP creation.

For the results presented in this thesis, the TAA does not run best-fit algorithms *e.g.* as proposed by [112], although its use is not waived.

6.2.5. Well informed nodes and degree of network awareness of nodes

There is a minimum time to which the LNM is said to be "aware" of the planned burst transits in the network. Since one of the major goals of C^3-OBS is to achieve very low burst loss due to the implementation of the "network awareness" at each node, which allows for a more efficient decision making process at the edge and core nodes, it is also important to define this time horizon to which the nodes are known to be "well informed" (Figure 90). For any given instant t_k, the time the node is well informed is

$$t_{WI} > t_k + n\delta, \tag{25}$$

where t_{WI} is the Well Informed time, n is the largest number of hops a control packet

may experience until it is replicated throughout the network, termed Network Control Diameter (NCD) and δ is the delay of the link (considered as the average delay of all links). As the control topology is a tree shaped graph, NCD may also be viewed as the maximum depth of this tree.

In a distant future ($t_x >> t_{WI}$) the node is therefore considered to be "fully informed" (see Figure 90) because it can schedule bursts with full knowledge of the network reservation status.

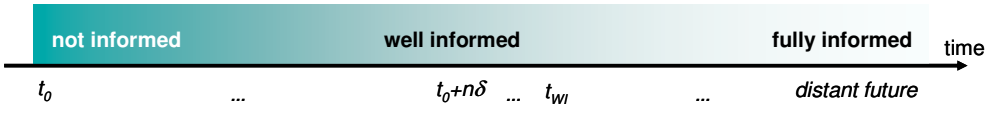


Figure 90 – Evolution of the degree of network-awareness for a core node in a C^3-OBS network.

It is clear that the better the given node is informed about the current network status, the better its routing decisions will be. Nevertheless, it is important to keep the data structures and decision algorithms to a minimum, as to lower the effort imposed on the network equipment logic.

As an example, consider the C^3-OBS network shown in Figure 85 and assume that at the given time instant t_0, node 1 and node 6 decide to schedule bursts that have to travel between them and so they do not collide with each other since they travel in opposite directions (assuming shortest path is chosen). Additionally, assume that burst 1 to 6 departs at time t_1 and burst 6 to 1 departs at time t_2. If these times are to be optimal, they must follow the aforementioned considerations and, for instance, in a TAG scenario, t_1 and t_2 would be

$$t_1, t_2 > t_0 + T_{Setup} + T_{OXC} \quad (26)$$

The degree of network status awareness each node has at each of these times, is expressed as follows: at time t_0, only node 1 knows that it will send a burst with destination to node 6 (likewise, node 6 in regard of the burst destined to node 1); but at

time $t_3=t_0+n\delta$ (n is the NCD and δ is the time the CP takes to travel the longest link, see Figure 87), all the nodes in the network know of these two transmissions, since this is the maximum time the CP takes to reach all nodes. So any decisions taken by the nodes up until time t_0 are known by all the nodes at time t_3. As an immediate consequence, the further the bursts are scheduled, the better probability they have to traverse the network without finding already reserved the necessary network resources. This is a known OBS result [72, 187-189]. Thus, one may conclude that for any given instant t_k, if a node tries to schedule a burst to a time after $t_k+n\delta$, it is likely to have all the information needed to decide whether the operation is feasible at that time. At this time the given node is therefore considered to be "well informed" and this delay is named as t_{WI} (see Figure 90).

If the network is well informed, *i.e.*, edge nodes are aware of the state of network resources, then they are able to send bursts and its control packets into the network without the T_{Offset} delay, provided the resources in the path nodes are already configured and available, profiting from the path accommodation *phenomena* [107, 190]. In his case, the burst will be sent not using the "shortest path" *criteria*, but following the route with the "no configuration needed on nodes". Other strategies that may decrease standard T_{Offset} time delay may take into account that some nodes along the path are already configured and so the path chosen can be the "lesser occupied node route", "fewer wavelength conversion route" and so on.

The risk of burst loss when a burst has already departed from its ingress edge node still arises when at least two nodes try to simultaneously schedule bursts that will try to use the same network resources in the same time period. In this case there are two possible causes for burst loss: the first is due to the fact that the network is large and the LNMs cannot be updated in reasonable time, or, the second, the offset time for the burst is smaller than the Well Informed Time. We call this a Concurrent Control Packet Scenario. The Concurrent Control Packet Scenario happens whenever two (or more) simultaneous reservation requests for the same resources are attempted. If the nodes are not well informed, the resource reservation is not simultaneous and only one of the bursts has departed by the time the LNMs are updated, then the nodes which bursts are still in queue will reschedule them to a later time, or a node may decide that a core node

in the burst path will apply other routing or buffering techniques to solve this contention. If the nodes attempt the resource reservation with apparent success and the bursts leave the nodes before the LNMs are updated, *i.e.*, when a set of resources *R* is registered as free for a given time interval *t'* at each of the LNMs, then burst collision will occur somewhere along the path.

Considering the network in Figure 85, an example might be one where that at time t_0 both node 1 and node 3 decide to send a burst through the path {node 1 → node 2 → node 4} and using the same wavelength in the link connecting node 2 to node 4 at an overlapping time interval (see Figure 91). The bursts are predicted to depart from node 1 and node 3 the same time t_1. Current time is t_0 and t_1 does not follow (25), *i.e.* $t_1 \leq t_0 + 3\,\delta$, being 3 the calculated largest number of hops the CP has to travel to inform all the nodes in this network (NCD) (see the proposed control topology in Figure 85). In this example, both the burst ingressed at node 1 and the burst ingressed at node 3 will try to concurrently use the only available data channel in the link between node 2 and node 4. At time $t_0+\delta$ (t_2 in Figure 91) node 2 will decide to drop one of the bursts, since it has only one data channel to node 4. Also, at time $t_0+\delta$ node 1 and node 3 have received the CPs issued by node 3 and node 1, respectively. So at this time, one of the nodes (the one that has the lower priority burst, or the one who has the lower burst loss ratio, or the one that follows some performance metric embedded in the common set of rules that operate the LNM) will reschedule its burst (time t_3 in Figure 91) as not to coincide with the occupancy of the link between node 2 and node 4. Node 2 does not need to send a negative acknowledge CP because the burst drop was inferred by the other nodes through the operation of their LNMs.

The Concurrent Control Packet event may still occur if more than one wavelength is available, but both concurrent nodes have scheduled the burst route to the same wavelength. In such a situation, node 2, which receives both the CP from node 1 and from node 3 will attempt wavelength conversion for one of the bursts (Figure 92). If there are no resources left to further route the burst, then it is dropped (Figure 91 at time t_3).

In a Concurrent Control Packet situation, by means of a common set of simple rules, *e.g.* node hierarchy in terms of burst loss ratio or in terms of burst priority, all the nodes in the network will identify the contention situation while updating their LNMs with the data from the CPs from node 1 and node 3 and will assume which burst was successfully transmitted and which burst was dropped or wavelength converted. In the case of wavelength conversion, the node where the contention occurs will try to convert one of the incoming bursts (possibly the last one to arrive to the node) to the next immediately free data channel. Having detected the Concurrent Control Packet scenario and computed the new path or new port for the burst, the next node in the second path of the burst will expect it to arrive not at the initial data channel, but at the next immediately free from the previous node point of view.

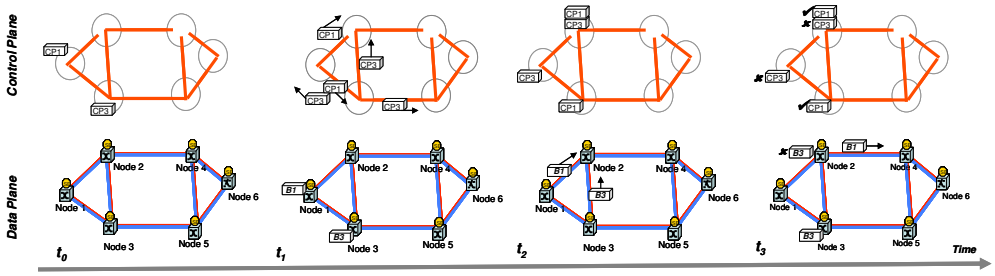


Figure 91 – Sample concurrent control packet event.

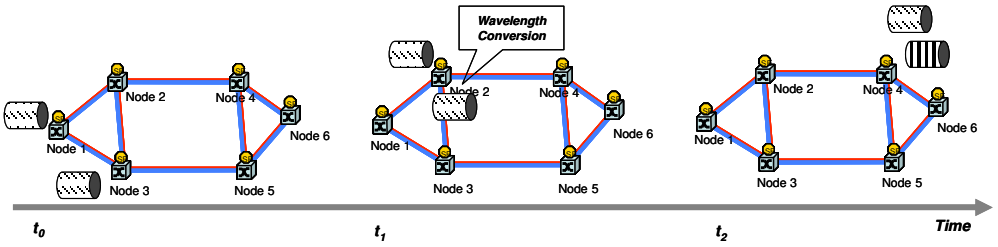


Figure 92 – Bursts in a concurrent control packet scenario at Node 2.

Burst transmission is therefore more likely to succeed when it is scheduled to a time near of after this "well informed" threshold t_{WI}. Note that t_{WI} is a function of the NCD of the network and thus, the primary goal of the aforementioned spanning tree

algorithm is to minimize this value as to lower the t_{WI} as to obtain a shorter "fully informed" status for all nodes in the network.

For networks with long links, one of the proposals to overcome the high CP propagation time is to perform this broadcast concurrently with the burst assembly process as proposed in section 6.5.

6.2.6. Scope, limitations and proposed solutions

Since the control channel is planned to be used in a broadcast manner, its scope of application needs to be defined: the expected size for an OBS network or, in this matter, for a C^3-OBS network is a sensible topic. Additionally, another two major aspects need focus – the node count and the link length, since these influence directly the number of messages the control channel can contain.

One of the expected problems to overcome in C^3-OBS is the collision of messages in the control channel. Figure 93 shows the impact of the size of CP messages in the occupancy of the Control Channel, considering several network control topology diameters and possible message sizes. With a link length of 1 km, a maximum of 62 messages (estimated each to be 100 bytes long) are allowed to simultaneously traverse the channel at the given moment of time. Each of these 100 byte long messages traverses 1 km control channel in 5.08 s, considering a transmission rate of 40 Gb/s. Figure 93 shows the maximum number of messages per second expected to populate the network control channel.

The calculated number of messages that can be conveyed in the control channel was assessed in [24] thus allowing the network to manage an aggregated bandwidth of $1.4x10^{14}$ b/s (Figure 94), taking into an account the average burst size of 2,620 KB (value obtained from [105]) and estimating that only one third of the messages are used to allocate network resources for bursts.

Figure 93 shows that the number of manageable messages in the control channel is inversely proportional to the message size and directly proportional to the control channel link length. Keeping that observation in mind, application of the C^3-OBS

networks is preferred at the metro level, where the network diameter is relatively small, allowing for a fast propagation of the CP to all nodes: Additionally, the number of nodes should be moderate as to allow the maintenance and effective operation of its LNM databases.

The increase on the node count may augment proportionally the network control topology diameter and expectedly the number of the control messages, thus resulting in an increase of the number of CP collisions, which demands for the application of faster logic at the nodes. Additionally, increasing the number of nodes will augment the size of the LNM and raise both the complexity and the operation time of its algorithms. In the same manner, higher complexity of the signalling algorithms, namely by the operations performed by complex reservation or void filling mechanisms, may render ineffective the maintenance of the LNM at all. The performance issue for complex versus simple signalling protocols is well studied and lead to the proposal of JIT$^+$ [55].

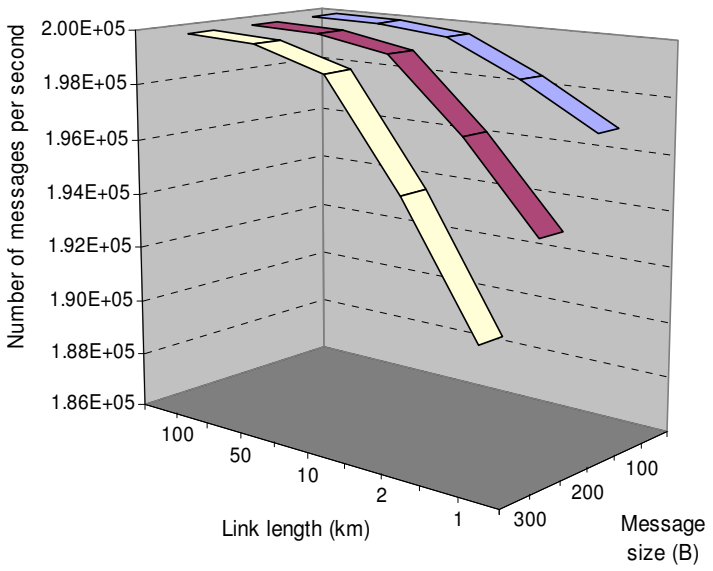


Figure 93 –Number of messages per second in the control channel versus link length versus message size.

On the other hand, increasing the link length increases the delay for the reception of the CPs. The aforementioned analysis brings out a number of questions related to the possible shape and applicability of the C^3-OBS networks. One possible scenario to overcome the aforementioned limitations is to limit the C^3-OBS dimension. The second approach entails applying burst assembly and scheduling algorithms that are sensitive of the network awareness degree supported by the nodes, as described in section 6.2.5. A third approach is the application of Domains (see section 6.4). Still, in section 6.6 we assess the performance of long real C^3-OBS topologies, with very interesting results.

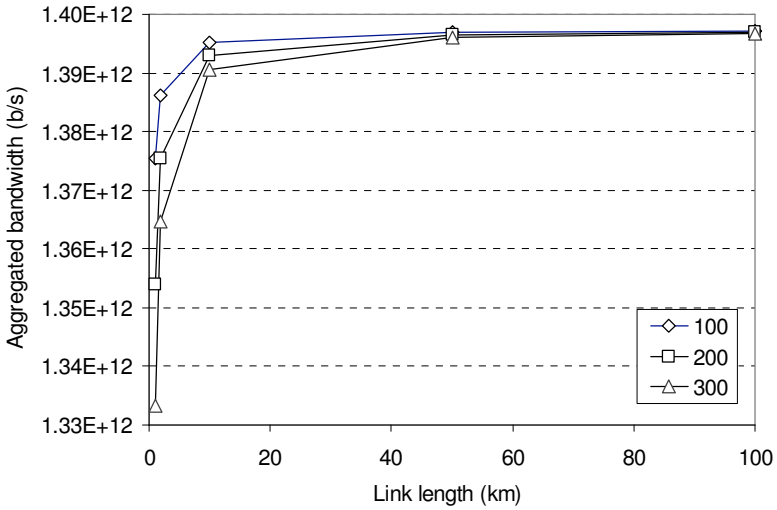


Figure 94 – Manageable aggregated bandwidth versus CP size for three control message sizes (100, 200 and 300 bytes).

6.3. IP-PAC and burst assembly in C^3-OBS

The C^3-OBS architecture and the IP-PAC machine concept presented in Chapter 5 may be used conjointly to more efficiently handle and manage an OBS network. Following this approach, the burst assembly and disassembly tasks are no longer to be handled by the OBS edge nodes, but by the IP-PAC machines instead. As IP-PAC machines are generically designed to connect IP networks and do not foresee a close

connection to OBS networks, some changes are suggested in this section, to allow higher efficiency.

At the edge of a C^3-OBS network, we can assume that the query to the LNM of an ingress node may be triggered by the burst assembly machine using a control message (still in the electronic plane) as soon as the first packet enters the burst creation queue in the client node (the burst assembly machine), because most of the characteristics of the burst are well defined by that time, namely, the burst destination, the burst priority and to a reasonable extent, the burst estimated departure time and / or the burst estimated size. Viewed from this perspective, the burst assembly process may run concurrently in time with the query in the LNM and possibly with the CP propagation. A different approach was suggested in [67], where the authors state (quote) "(…) As soon as a data burst (DB) is ready, its corresponding Burst Header Packets (BHP) will be generated and sent into a separate control wavelength (…)" (end quote). This approach imposes an additional delay to the packets inside the burst, as they will have to wait for the CP manufacture in addition to the queuing time in the burst assembly process. The maximum time a packet has to wait, starting from the instant it ingresses in the burst assembly queue until it exits the OBS network, *i.e.*, after the burst is disassembled, is:

$$T_{MaxPacketDelay} = T_{Assembly} + T_{CPCreation} + T_{Offset} + T_{Travel} + T_{Disassembly} \quad (27)$$

where $T_{Assembly}$ is the burst assembly time, $T_{CPCreation}$ is the time the ingress node takes to create the CP, T_{Offset} is the offset time as defined by equations (1) and (2), T_{Travel} is the time the data burst takes to travel the total path length and $T_{Disassembly}$ the time the egress node takes to disassemble the burst. This equation states the maximum time and thus is applicable if the packet in question is the first packet of the burst. As the packets arrive at the burst assembly queue and are placed at the tail of the burst, $T_{Assembly}$ tends to zero. Please note that some burst assembly algorithms do not follow a First In First Out (FIFO) ordering of the packets in the burst, *e.g.* in [74] a bust includes packets with different priority constraints, sorted in a special order, *i.e.* higher priority packets are placed in the burst head, to allow possible burst segmentation and burst tail loss.

If the ingress node is notified of the burst creation by the arrival of the packet that creates the burst assembly queue, equation (3) becomes

$$T_{MaxPacketDelay} = \max\{T_{Assembly}, T_{CPCreation} + T_{Offset}\} + T_{Travel} + T_{Disassembly}, \quad (28)$$

as the burst assembly process taking place in the ingress node runs concurrently with the CP creation and possibly with the network configuration phase measured by the offset time.

Following this approach, the IP-PAC is required to do two additional things:

1. To be able to predict the estimated characteristics of a burst as soon as the burst assembly queue is created; and
2. To be able to send a CP request to the ingress core node on the C^3-OBS network.

Additionally, the IP-PAC machine may need to perform the following actions:

3. Keep the burst in the assembly queue or in a buffer in case of negative availability of resources (NAK message from the C^3-OBS ingress core node),
4. Send the adequate NAK messages to the client nodes whose packets were in the dropped bursts.

The burst characteristics that require prediction are the burst size and the burst estimated departure time. These two properties may be easily defined using the IP-PAC knowledge of the input traffic conjointly with the burst assembly thresholds.

Using the IP-PAC machine and C^3-OBS architecture, no longer makes sense to differentiate between edge node and core node [44, 45], since it becomes clear that an edge node is an IP-PAC machine, and a core node is one that connects only to other C^3-OBS nodes and possibly to IP-PAC machines (see Figure 95). While in [191, 192] it was proposed that core nodes do not have the ability to produce CPs, in C^3-OBS and

due to its improved functionalities, *i.e.*, the LNM and the TAA, each node may create the necessary CPs.

Within this approach and provided that the CP has a compatible format for OBS and for C^3-OBS, it is possible to consider a mixed network where some nodes are C^3-OBS (forming an island of well informed nodes) and other are OBS nodes. The analysis of these mixed topologies will be subject of future work.

By using the IP-PAC machine as the ingress node to a C^3-OBS network, we achieve two simplifications: firstly, the OBS edge node is replaced with a more versatile machine, capable of performing buffering, assembling, disassembling and the management of tributary IP traffic for several network scenarios; secondly, since in a C^3-OBS network there are only core nodes, the investment becomes more flexible: in such a scenario, all it takes to transform a core node into an edge node is the connection of an IP-PAC machine.

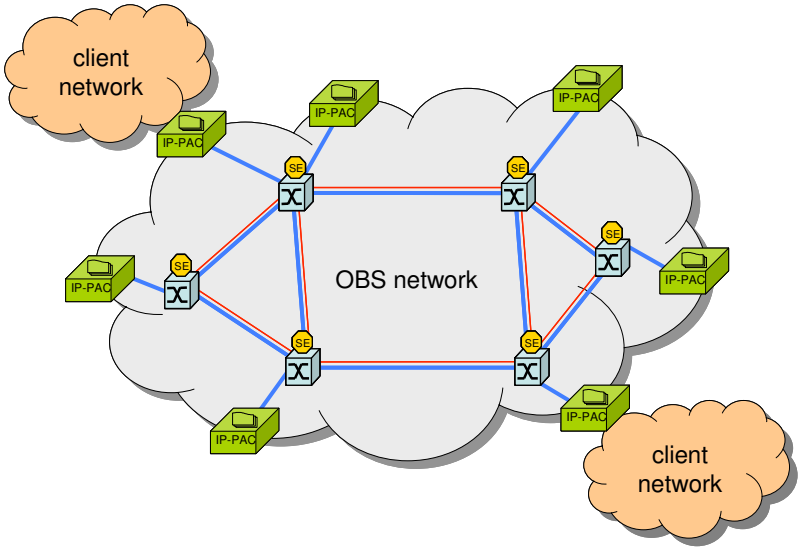


Figure 95 – Sample C^3-OBS network with IP-PAC machines as edge nodes.

6.4. C^3-OBS domains

Among the limitations that suggested the decrease the complexity of the architecture and the operation of a C^3-OBS network (see section 6.2.6), we count the

limitation of the number of nodes in a well informed network and the limitation of the length of the control channel, *i.e.*:

i) To allow only a restricted number of nodes as to maintain a LNM within the limits of acceptable computability; and
ii) To limit the network spans of metropolitan or regional haul, as to allow the fast propagation of network control packets.

In electronic networks, one of the proposed strategies to simplify routing tables is the use of Domains. Each routing equipment maintains a routing table regarding only its domain and there are border or frontier routers that interface two or more domains.

This problem has not been studied in OBS, since each OBS node only sees its adjacent neighbours. But the "network awareness" resulting from C^3-OBS imposes an additional effort to the logic of the node and this effort increases with the size of the network. It is possible to use the domain concept in C^3-OBS, as a way to limit the network awareness effort to an adequate number of nodes.

Having in mind what we have presented so far, we propose the following:

1) A C^3-OBS domain is a set of nodes that have common control channel architecture. Domains interconnect via a frontier node that listens to more than one different common control channel,
2) Nodes in a given C^3-OBS domain will be aware only of the status of nodes of that domain,
3) Nodes in a given C^3-OBS domain will know the maximum setup times (number of hops or NCD as the worst case value) for every other domain they may connect to,
4) C^3-OBS frontier nodes will be aware of the nodes from all the domains they belong to and
5) C^3-OBS frontier nodes may interface with several domains and have a separate control channel topology for each domain they interface with.

The use of domains allows to overcome the scalability problem helping to enlarge the proposed scope of C^3-OBS network from metropolitan or regional to national or continental.

In Figure 96 we show 8 domains; domains 1 to 7 have the same topology (data and control channel) and domain 8 has an unknown topology. For this example, we have kept the numeration on the nodes of networks in domains 1 to 7. We will refer these addresses as *Dx.y*, where *x* refers to the domain number and *y* refers to the number of the node in this domain. Of course, some nodes have more than one address as they belong to more than one domain and for example, D6.6 refers to the same node as D7.1.

With the name of each domain, Figure 96 shows a value that represents the maximum number of hops from any of the frontier nodes to any node in that domain. For example, the D3 domain has a value of 3, because from node D3.1 to any other node in D3, it takes at the most, 3 hops.

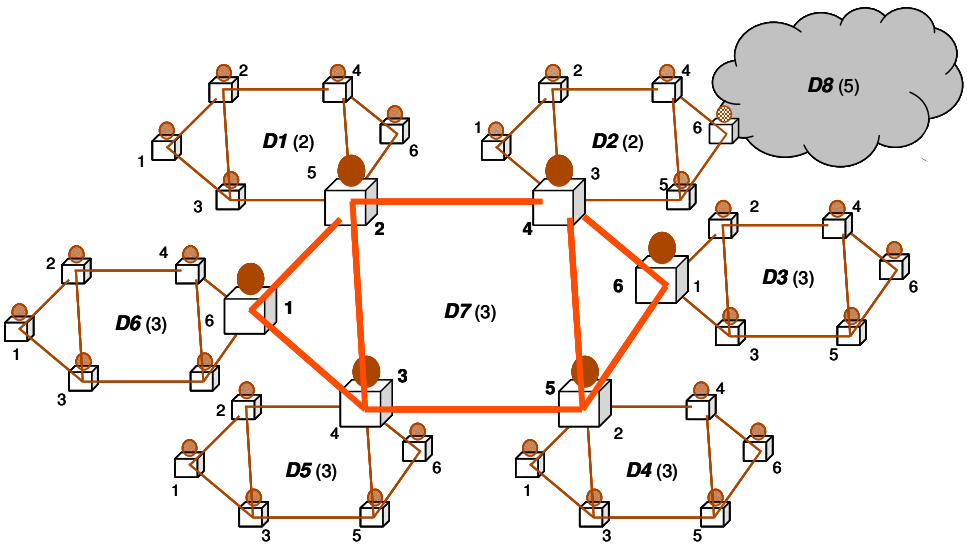


Figure 96 – Example of domains in C^3-OBS

Let us now suppose that node D4.5 wants to send a burst to node D1.1. The calculation of setup time considering a TAG protocol (for example) will be as follows. We know that inside each domain, nodes can manage internal traffic. In this case, each of the satellite networks (D1 to D6 and D8) know their gateway address (or gateway

addresses, in the case of D2). So nodes in domain D4 know that the path for D4.5→D1.1 is locally D4.5→D4.2. Nodes D4.2 and forward, route the burst accordingly to their routing tables. In domain D7, node D7.5 knows that D4.5→D1.1 is locally D7.5→D7.2. And finally D4.5→D1.1 to node D1.5 in domain 1 is D1.5→D1.1.

Node D4.5 calculates the offset time adding its local offset (calculated as show previously), to a value that belongs to its Domain Routing Table. For our example and relating to Domain 4, the Domain Routing Table (present in each node in this domain) would have the following content: D1=5; D2=5; D3=6; D5=6; D7=3; D8=10. These values reflect the maximum number of hops a Data burst has to travel to get from the Frontier node in Domain 4 to any node in Domains 1, 2, 3 and so on.

As we limit the global network awareness by the introduction of domains, node D4.5 is ignorant as to the status of Domain 1. Nevertheless, node D7.2 knows the state of both domains 1 and 7 and may choose not to forward (overload) the burst into domain 1. So for the burst transmission intended before, the total offset time would be:

$$T_{TotalOffset}=T_{OffsetLocal}+5.T_{Setup}, \tag{29}$$

where:

- $T_{TotalOffset}$ is the total offset time for the burst and
- $T_{OffsetLocal}$ is the local offset time as defined in (24).

or in a more general form,

$$T_{OffsetLocal}=T_{OffsetLocal}+DND(Dd).T_{Setup}, \tag{30}$$

where:

- $DND(i)$ is the function that returns the value of Domain Network Diameter table associated with the key i
- Dd is the key referring to the destination domain

As the network awareness is limited to the restricted number of nodes belonging to the same domain, node D4.5 is ignorant as to the status of Domain 1. Nevertheless, node D7.2 knows the state of both domains 1 and 7 and may choose not to forward (overload) the burst into domain 1, or may choose to route the burst alternatively, respecting the Domain Network Diameter previously announced to the other domains.

A hybrid network where domains implements TAG signalling for internal traffic and implements TAW for extra-domain traffic may be adopted, although this would add a few extra hops to the communication. In this last example, the closest well-informed node of Domain 1 is node D7.2, which is 3 hops away from any source domain and so the round trip for the CP from the entry gateway from Domain 4 into Domain 7 to the closest well-informed node about Domain 1 would be at the most 6 hops. In this case, (30) would be

$$T_{TotalOffset}=T_{OffsetLocal}+9.T_{Setup.} \quad (31)$$

The use of network domains in OBS allow the coverage of a much wider geographical area, being for instance, the domains 1 to 6 several metro ring and the domain 7 a national ring interconnecting each of the other domains (see. Figure 96).

Figure 97 shows a functional scheme for an Optical Cross Connect Frontier Node for a C^3-OBS. Note that in this node, the internally generated CPs may not be replicated to all output fibres – instead, the Signalling Engine (SE) which follows the operational algorithms associated with the C^3-OBS network, decides to which fibre or fibres a given CP must be send to. This allows for the separation of the control topologies of the several network domains this node may interface with.

The introduction of C^3-OBS Network Domains allows the following:

a. As the C^3-OBS nodes are only aware of the status of the nodes in its own domain, the LNMs in each node is kept to an acceptable size in terms of computing effort.

b. As the length of the paths within a Domain is minimized, the propagation of the network CPs is done faster and thus, the Well Informed Time for the nodes in the network is also minimized, allowing for a shorter offset time between the CP and the burst

c. The utilization of Domains allows for better network planning.

The scheme in Figure 97 does not include neither does not limit any add-drop burst capability that may be integrated. It also does not show and does not limit the use of optical signal regeneration or FDLs.

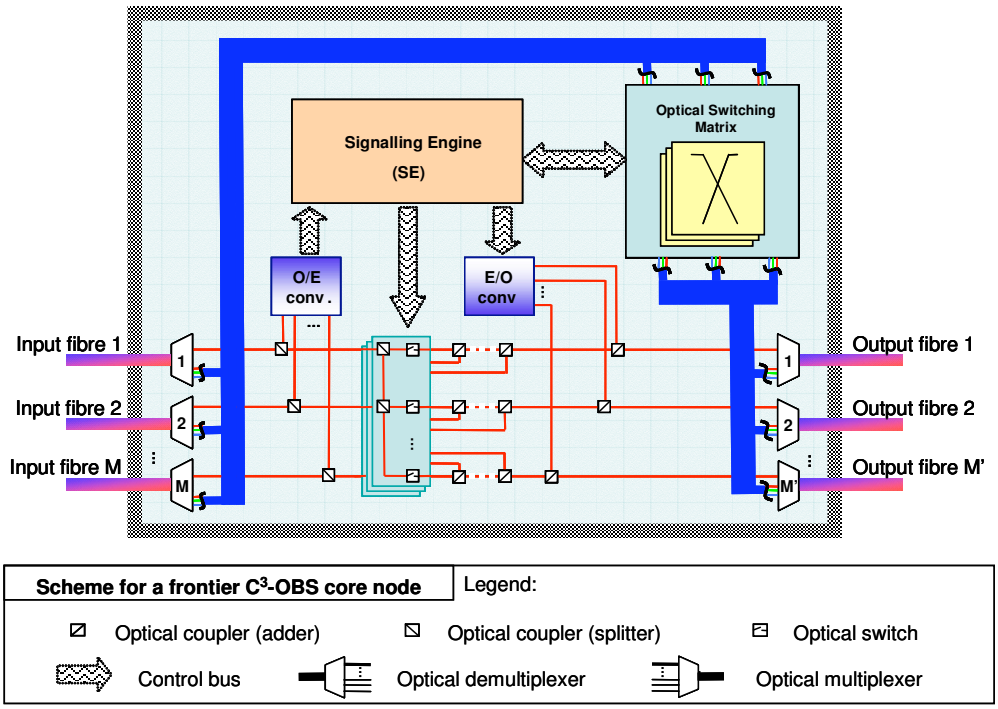


Figure 97 – Schematic illustration of the block architecture for the frontier C^3-OBS node.

6.5. Modelling and simulation of OBS and C^3-OBS networks

6.5.1. Performance assessment through simulation

Network performance evaluation is achieved using several approaches: analytical tools that model the network behaviour, *in situ* measurements and simulators [191-193]. However, when the networking paradigms that are to be evaluated have not

yet reached the status of prototype or are too expensive, such as the C^3-OBS, the only performance assessment tools that are left are the analytical tools and the simulators. The performance assessment of an OBS network node has been published in the literature, *e.g.* Teng and Rouskas in [55, 84, 194] for the JIT, JET, Horizon and Jumpstart protocols.

Analytical models for JIT, JET, Horizon and JIT^+ nodes are presented in [55] and the analytical results for a single OBS node are successfully compared with simulator results. However the analytical models presented are only exact for the case of the JIT node [55] and its use is limited to low connectivity topologies [11, 55], thus rendering the simulation of complex network topologies such as the EON or the NSFnet viable only with the use of network simulators. A very thorough and detailed list of available network simulators in general and OBS simulators in particular, describing its advantages and drawbacks, is available in [11].

Following the analysis in [105], a previously built and validated simulator (OBSim) [105] was rebuilt and improved, to allow the simulation of the new C^3-OBS architecture. Java was kept as the programming language, due to its object manipulation capabilities. This is both a stochastic and deterministic, event driven, symbolic simulator [193]. Its double stochastic and deterministic nature derives from its ability to generate simulation events (burst creation, burst transmission requests, resource reservation requests and so on) that are either generated though a adequately modelled random function, or generated by the actions that result from the burst assembly algorithms applied to the previously recorded data trace files. In this last case, each simulation run returns exactly the same results because of the deterministic nature of its input and of the implemented algorithms (NAN routing was simulated only with stochastic traffic generation). The models that were used to generate random burst traffic are detailed in section 6.5.2. Simulators are event or time driven [193]. In the latter case, the simulator has a clock that commands the data flow inside the software. The OBS and C^3-OBS simulator is event driven as the data flow inside the simulator is triggered by events that are stored in an event queue. These events are burst transmission requests, burst creation actions, or any other types of messages added by edge or core nodes in response to the burst assembly, resource reservation and

management algorithms. Finally the simulator is symbolic because each of the simulated real objects, *e.g.* burst assembly machines, switching nodes, the signalling engine and so on, are symbols inside the software, *i.e.*, in this specific case these objects are instances of Java classes.

6.5.2. Burst traffic model

The burst traffic characteristics depend on the simulation scenario as the simulator admits two different scenarios: the first one is completely deterministic allowing the creation of bursts using a burst assembly machine that get its input from *tsh* files containing traces of real IP packets; the second scenario generates random bursts at random time and with random length.

The file feed scenario assumes that each user issues IP packets. These packets are aggregated into a burst in the edge node the user is connected to. When burst assembly conditions are met (time base, size based, or both) a burst is created and injected into the core network. This scenario assumes one or more *tsh* files for each user, each file containing several real IP packet headers as presented in section 4.3.1. Based on this information we can reconstruct a packet (since the payload of the packet is not really relevant in terms of simulation) and simulate its transmission in the network structure. When the whole file is parsed, a new file is assigned to the user. Since *tsh* files are traces of independent periods of time, they where synchronized by assuming that the timestamp of the first packet for each user is its zero time simulation point. We then subtract that timestamp from all following timestamps, which guarantees concurrent traffic generation for all users and also for the users that may end up using more than one *tsh* file in the simulation run.

Based on the conclusions previously presented in Chapter 4, we can assume that the traces are version agnostic, and hence we view the results as if IPv4 or IPv6 traffic was used.

The random burst creation scenario focuses on the bursts, not packets. Each user generates traffic with intensity described with two parameters: average delay between issuing bursts and average length of burst. Given these values, the user generates bursts

to the edge node that is attached to and the burst is then normally injected into the network. Each simulation scenario consisted of bursts being generated by users attached at each edge node, with an overall rate of 1 burst per millisecond per edge node, exponentially distributed, while the number of data channels was increased from 4 up to 72 in a 4 channel addition step. The burst size is also exponentially distributed with an average size of 10 milliseconds.

The destination for each burst was assigned randomly, following a uniform distribution. Clearly this scenario assumes some generalization (supported by previous research in [193, 195]), but offers much shorter simulation times. In the stochastic simulation the simulation followed the simulation conditions presented in [55, 62].

6.5.3. Simulator architecture

The OBS simulator is an object oriented, event driven simulator either stochastic or deterministic, since its input data can be read from *tsh* captured data files (see section 4.3.1) or randomly generated. The simulator models all the physical components of network with objects and processes, simulating the way the objects behave.

Classes in the simulator are divided by two packages: the Network package and the Event List package. The Network package contains the classes that allow the simulator to instantiate a virtual model of the network topology that is being simulated. Figure 98 shows the Unified Modeling Language (UML) class diagram for this package. The Event List package contains the classes that are used to control the data flow in the software during simulation time, including the classes that mimic the possible simulator events. Figure 99 shows the UML class diagram for this package.

A network topology consists in a list of nodes that connect with each other through links. Nodes have traffic generation characteristics and traffic switching resources, and links have traffic transport characteristics and inflict delay in that transmission in the sense that data has to travel the length of the link.

A network topology is virtualized in the simulator by the definition of relations between instances of classes belonging to the class Network. The Network class

instantiates the necessary classes Node, User and Link (see Figure 98), according to the script defined by the user and shown in Annex A. The links are considered unidirectional and thus if a bidirectional link exits between node 1 and node 2, there will be two link connecting the two nodes.

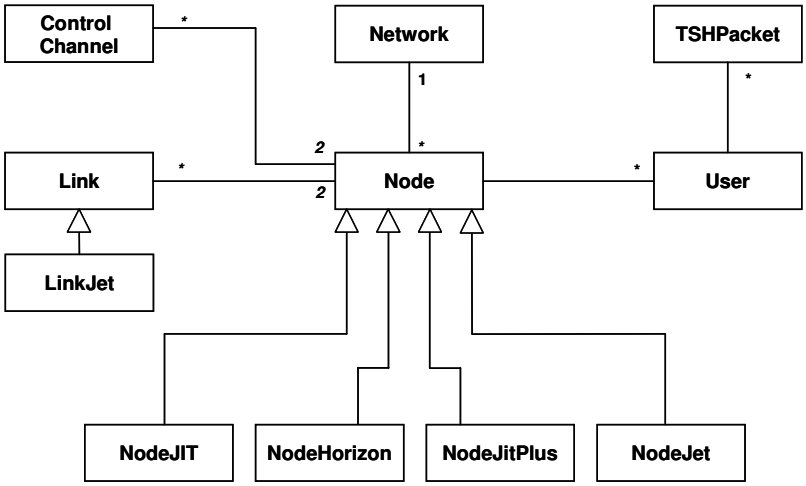


Figure 98 – UML Class Diagram for the Network package.

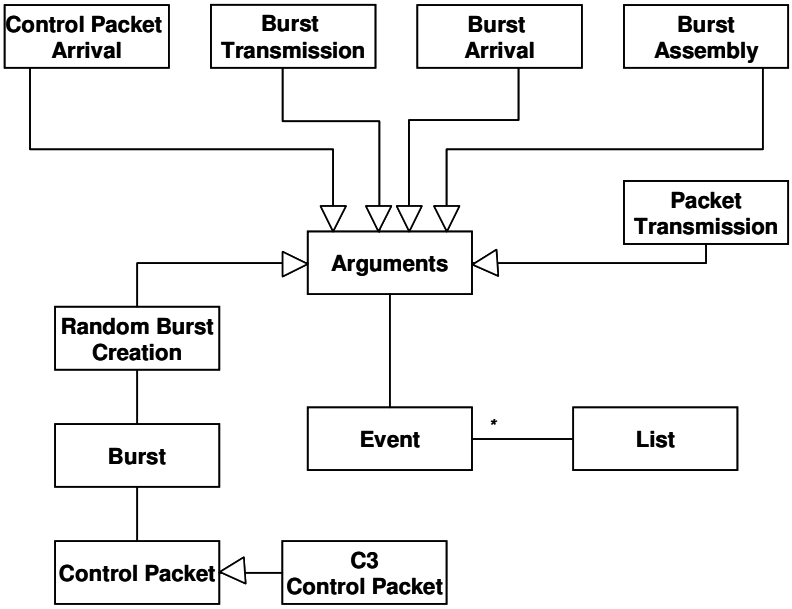


Figure 99 – UML Class Diagram for the Event List package.

The main features in the simulator that allow the simulation of the new architecture are:

1. In C^3-OBS a CP creates not only one event at each node it passes, but each time a CP is issued, it creates a several events at all nodes, scheduled in view of the time delay the CPs take to propagate through the defined control channel topology,
2. The control topology is read from a configuration file;
3. Finally, each node implements the TAA and LNM and thus, each burst ingress request has a TAA response resulting from querying its LNM.

There is a complete description of the simulator architecture, its parameters and sample input and output files in Annex A.

6.5.4. Simulator validation

The validation of the results of the simulator was done following [193]: firstly, the simulator logic and its model were thoroughly checked each time a new version was implemented, secondly the results were compared with the results published in [84] and thirdly the simulator log files were analysed for protocol behaviour congruence and simulation model correctness. These two tasks were performed to comply to the three last points of the checklist presented in [193]. The first two points of the checklist, the homogeneity and independence of the random number generator [196] and the validity of the stochastic number generators were tested in initial studies and proved to be adequate, *i.e.*, as a previous step of the validation process, and following [193], the validity of the Java random number generator and of the generator of the stochastic variables inside the simulator were verified. The Java Random class uses a 48-bit seed, which is modified using a linear congruential formula to return a random number. Throughout the simulator the objects from the class were instantiated with an empty constructor, which means that then the seed was initialized to a value based on the current system time. The validity of the Java Random class is guaranteed by the Java definition documentation [196, 197]. The validity of the stochastic variables derives

from the validity of the Java Random class as the stochastic variables in the simulator only use this random number generator.

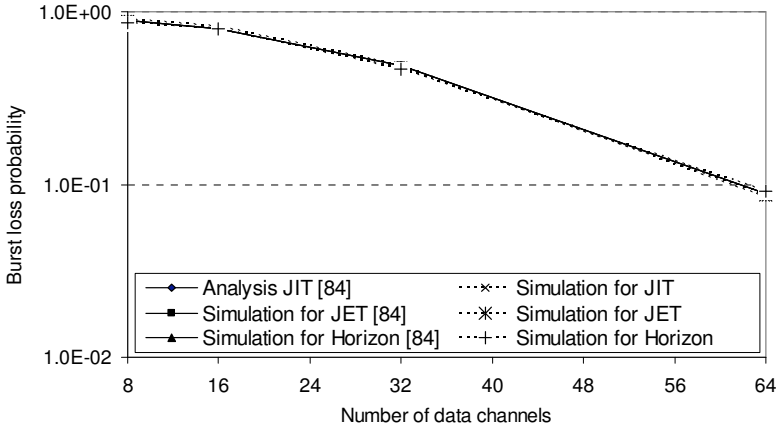


Figure 100 – Burst loss probabilities as a function of the number of available data channels, comparing the results published in [84] and obtained by simulation when the mean burst size was defined equal to T_{OXC}, for current technology values.

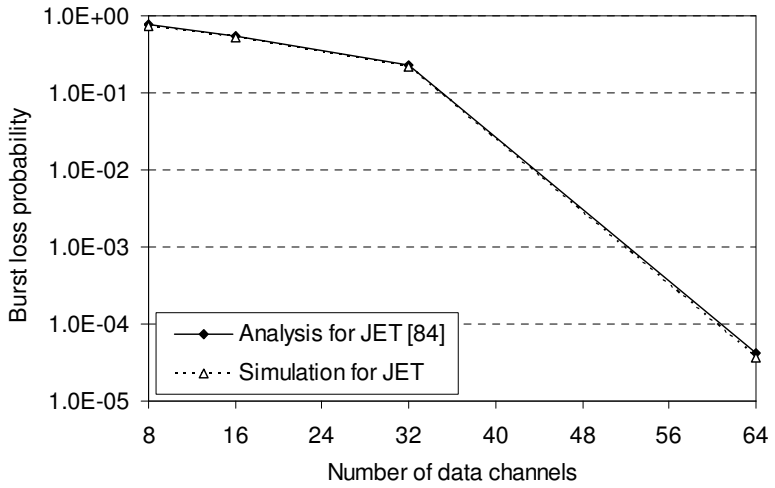


Figure 101 – Burst loss probabilities as a function of the number of available data channels, comparing the results published in [84] and obtained by simulation when the mean burst size was defined equal to $5.T_{OXC}$, for current technology values.

In [84] Teng *el al.* published a detailed performance comparison for the JIT, JET, Horizon and JIT$^+$ OBS protocols. These authors extended this analysis for several

burst sizes and several scenarios for different T_{OXC} and T_{Setup} times. Figure 11 in [84] shows burst loss probability when the mean burst size was set to 10 ms for the analytical model for JIT, JET and Horizon and for simulation for JET, Horizon and JIT$^+$ of a single OBS node with cross traffic. Figure 100 shows the comparison between the results published in [84] and the results obtained by our simulator using the same scenario and traffic generation parameters, *i.e.*, assuming values for T_{OXC} and T_{Setup} that are expected for current technology and assuming the mean burst size equal to T_{OXC}. In order to further validate the results of the simulator for lower burst loss probability values, a simulation setup was defined also following [84] but for mean burst size values equal to $5T_{OXC}$. The remaining simulation parameters were kept unchanged. Figure 101 shows the comparison between simulation values for the burst loss probability for the JET protocol published in [84] and the values obtained by our simulator. Expectedly, due to the stochastic nature of the simulators, the results shown in both figures are not equal, but in a very close range.

6.5.5. Simulation results

Simulation variables for OBS and C^3-OBS networks, namely, the suggested times for T_{Setup} and T_{OXC} were adopted following [55] and are presented in Table 12.

Table 12 – Values for simulation parameters for OXC configuration and node setup times.

T_{OXC}	T_{Setup}		
	JIT, JIT$^+$	Horizon	JET
10 ms	12.5 s	25 s	50 s

A typical simulation run is divided into following steps:

1. Creation of the network: in this step, the network description, read from the configuration files, starts the instantiation of the objects that will model the network. *Network* object is created, with number of parameters that are applicable to entire network. After this the Simulator creates the topology representation by creating instances of the *Node*, *Link*, *User* and eventually *ControlChannel* classes. According

to the description file, all nodes are placed in the network, their parameters are set and they are connected through *Link* class instances, for which parameters are also set up. After this step the Simulator has knowledge about structure of the network and also about traffic generation.

2. Creation of event list: in this step, objects representing simulation events and the event list are created. The time unit is defined as picoseconds to allow keeping very precise track of event order. When the objects of the simulator are instantiated, the event list is void. The list starts to be filled with events as follows: if the simulation scenario is random burst creation, the simulator will generate a single burst from each user (to a random user destination) with random delay from the initial time of the simulation and with random size (following the defined time and size models); otherwise, if simulation scenario assumes the IP packets are assembled into bursts from the trace files, one burst from each user at the zero time of the simulation with a random user destination will be generated. In either simulation scenarios, this starts the event list and its processing; other events will populate the list, such as new burst generation and the network response to the initial requests.
3. Event list processing: the simulator reads the first event from the event list and processes it; the first *Event* object is removed from event list, its contents read and messages are sent to the appropriate objects, which in turn may add new events to the list. Initially the events on the list are burst transmissions, but they are followed by creation of other events *e.g.* burst assembly or events corresponding to burst traversing the network. Statistics are gathered while the simulation runs. This continues until the simulation time is larger or equal to the target time of the simulation or the number of generated bursts per node has been exceeded (as described in the *Parameters* class). After that moment, the processing of a transmission event does not originate the creation of new transmission requests and thus the event list is not increased with new events. When the list becomes void, the simulation stops.
4. Diagram creation: at the end the Simulator may display the statistics in the console output, into a text file or a diagram representing topology of the network. The diagram allows displaying statistics for each node and for each link, placed according to the coordinates defined in the configuration files. The text

representation allows a much bigger configurability, including the choice of the form and the content we would like to see, by formatting the output in the *start* class.

To obtain the burs loss probability results, an interval of 95% for the degree of confidence was estimated using the method of batch means [193, 195, 198]. The number of batches for each result was set to 30, the minimum value to obtain the aimed confidence interval and each batch run lasting until at least 120.000 bursts are generated per node. As the obtained confidence intervals are very narrow they are not shown in the figures to allow an improved readability. Nevertheless, Figure 102 attempts to show the simulation for the JET protocol shown in Figure 101, with the obtained confidence intervals. A sample result of the output of the simulator on a short run is shown in Table 14 in Annex A.

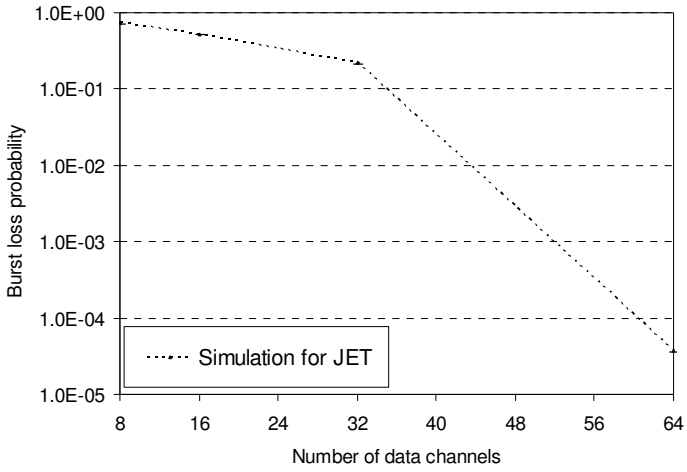


Figure 102 - Burst loss probabilities as a function of the number of available data channels, obtained by simulation when the mean burst size was defined equal to $5.T_{OXC}$, for current technology values, showing confidence intervals.

6.6. Performance assessment of OBS networks

The performance assessment of the discussed OBS architectures is presented in this section. Here we compare the performance of the C^3-OBS architecture with the OBS architecture for several network topologies and parameters.

Our initial hypothesis is that C^3-OBS distributed control of the network has better performance that individual control as implemented by OBS. The possible reasons for the increased performance, *i.e.,* the reasons for the decrease in the burst loss probability, are twofold:

A) Firstly, because of the increased routing capabilities of the nodes, made available by the operation of the LNM. The path of the burst is always chosen burst by burst, in view of the resource reservation status of the network for the estimated departure time of the burst. The path may not be the shortest one, but rather the shortest available path for the expected time of departure of the burst. Also, if no path is available, a node may still decide to transmit the given burst and rely on core nodes to reroute it or drop the burst that had previously made the reservation, *e.g.* in a case where priority criteria are applied. Thus, as fewer bursts are lost, performance is increased because of the more efficient manner the initial route is planned and route management is performed by the nodes. Furthermore, as edge nodes are aware of the state of network resources, they are able to send bursts and their CPs simultaneously, *i.e.* with an offset time of zero, providing the resources in the path nodes are already configured and available, profiting from the path accommodation *phenomena* [108]. This would configure a different void filling mechanism without the actual computational effort that void filling algorithms impose. In this case, bursts may be sent according to other routing algorithms such as ones that choose the route with the "fewer configuration on nodes" or "lesser occupied node route" or even "fewer wavelength conversion route", instead of using the more usual "shortest path" criteria. The routing decision using the above or any other criteria is to be executed by the edge node responsible for the transmission of the given burst and included in its corresponding CP, in regard of the known state of the network (through query to its LNM).

B) Secondly, if the resources needed to transmit a burst are not available from entry to exit in the network, the burst is not admitted at all (and thus remain and can be buffered in the client or edge node). As a consequence

of this, an amount of valuable bandwidth can be used by other bursts to successfully transmit, *i.e.*, this previously wasted bandwidth may now be reused by other bursts to successfully make their way in the network.

To assess the relative weight of each of the expected causes for C^3-OBS performance improvement, it was decided to test the increase on the efficiency of the second of the two expected causes, the re-usage of otherwise wasted bandwidth (occupied by bursts which where to drop somewhere along its path).

To test this last hypothesis we eliminated possible gains that could result from routing and rerouting, *i.e.* the C^3-OBS architecture was simulated for the simplest class of topologies that admit only one possible simple path between any two nodes – bus topologies. Two topologies were tested – three node and five nodes bus networks. As may be seen in Figure 103, C^3-OBS shows a lower burst loss probability for all scenarios, although this decrease in the burst loss is not significant – in average, only 7% less in the three node bus and 10% less in the five nodes bus with 1 km link length. The gain observed for C^3-OBS is accounted because more bursts where able to transit the network using resources that would have been otherwise occupied by bursts that would drop. The 7% to 10% increase observed is a consequence of the increase in the number of nodes and users of the topology and is also expectable, mainly because of the direct relation between burst loss and network load – the bigger load the network experiences, the higher its burst loss probability and thus, in OBS, and more wasted resources the network experiences.

Having accounted for the effect of the reused bandwidth gained from the shift in the architecture, we set to measure the increased efficiency of OBS because of the two conjoint effects – reused bandwidth and enhanced routing capabilities.

Figure 104 shows the increased efficiency of C^3-OBS when compared to OBS for ring networks of four, six and eight nodes with 1 km link length. Its visible how for the eight nodes ring, there is an increase of an order of magnitude for the 32 data channel scenario, while for other networks the performance increase is higher and occurs in fewer data channels scenarios. This increase of performance for C^3-OBS compared to OBS is higher when the network node count decreases – for the four nodes

ring, at the 28 data channel scenario, the C^3-OBS burst drop probability drops to zero while its equivalent for OBS remains around 0.961%. As mentioned before, this is also a function of the offered load each simulation scenario poses – the higher the number of nodes, the higher the load posed on the network.

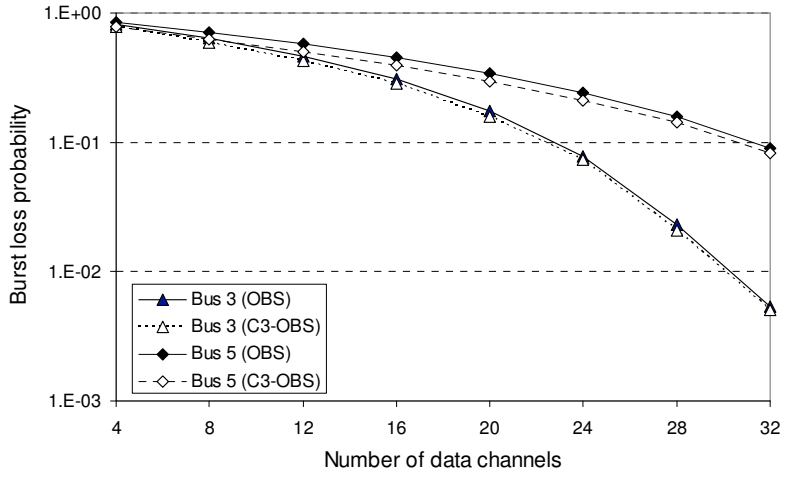


Figure 103 – Burst loss probability versus number of data channels for three and five nodes bus topologies for OBS and C^3-OBS.

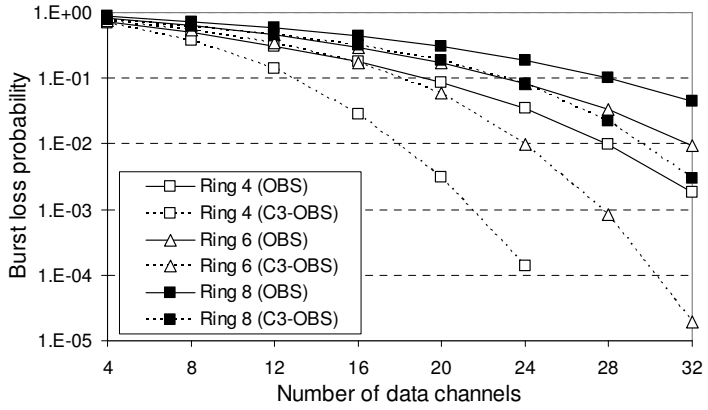


Figure 104 – Burst loss probability versus number of data channels for four, six and eight nodes ring topologies for OBS and C^3-OBS.

As the number of possible routes for ring networks between any two points is still limited to 2, it was decided to increase the connectivity of the 8 node ring topology with two chords, connecting nodes 2 to 6 and 4 to 8. These two chords increase the

number of possible paths between any two nodes, which in turn allows a better planning of burst routes by the C^3-OBS routing algorithms. Figure 105 and Figure 106 show the 8 node ring and 8 node crossed ring topology and control channel topology used in the simulations.

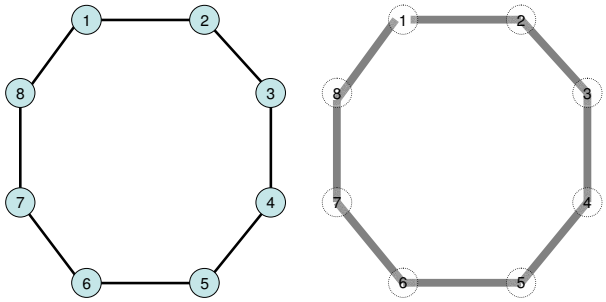


Figure 105 – Topology for the eight nodes ring network (left scheme) and the implemented control channel topology (right scheme).

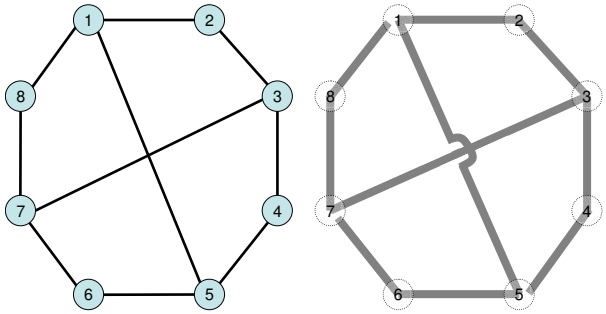


Figure 106 – Topology for the eight nodes ring network with two chords (left scheme) and the implemented control channel topology (right scheme).

Figure 107 shows the burst loss probability for the eight nodes ring and eight nodes crossed ring for OBS and C^3-OBS. It is visible that the increase of the nodal degree of the network (from 2 in the 8-node ring to 2.5 in the 8-node crossed ring) exhibits a decrease in the burst loss probability, even for OBS, which is an expected result [133]. Also expectedly, we can see that eight nodes crossed ring OBS network performs better than eight nodes ring C^3-OBS network. Comparing the OBS and C^3-OBS performances for the eight nodes crossed ring, Figure 107 shows a near 2.5 order of magnitude performance increase for 24 data channels. Although the number of searchable paths for the Travel Agency Algorithm was kept to 2, the efficiency increase

for the eight nodes crossed ring is a result of the increase of the number of available paths for the C^3-OBS routing algorithms.

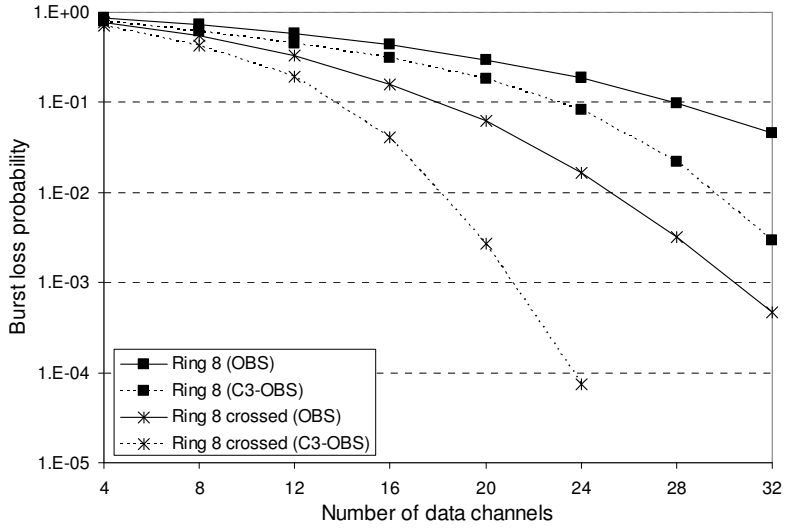


Figure 107 – Burst loss probability versus number of data channels for eight nodes ring topology and eight nodes with cross-connections (second with sixth node, fourth with eighth node) for OBS and C^3-OBS.

6.6.1. Performance assessment of EON and NSFnet with OBS and C^3-OBS

Simulations with real topologies were done with the 19-node European Optical Network EON [130] (see Figure 108) and with the 14-node NSFnet topology (see Figure 27). The simulated EON is a network with 19-node and its largest links (Lisbon-London and Berlin-Moscow) are 1600 km long. Its minimum Well-Informed Time is thus 8 ms, as this is the propagation time for the largest link in the network. We also performed simulations to the EON using a homographic topology whose links are equal in size, being this size defined as 1 km. Also, all links were considered having equal transport capacity (while in the real EON links have different bandwidths) and to each of the 19-node was assigned an equal traffic generation probability. EON is a well connected network, having a mean nodal degree of 3.579. NSFnet is a network covering the USA territory, consisting in 14-node connected by 21 links, the longest of which measures around 3000 km; its nodal degree is 3.143.

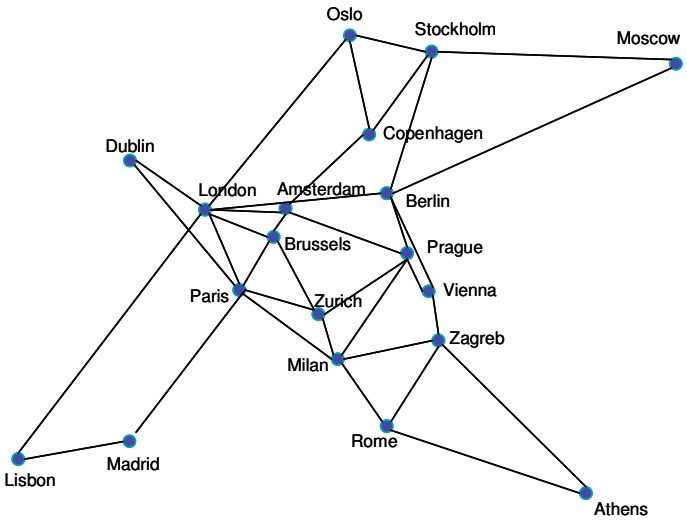


Figure 108 – The 19-node European Optical Network (EON) topology.

Figure 109 shows the results for the OBS and C^3-OBS burst loss probabilities for EON. It is visible that for 32 data channel scenario, OBS returns a zero burst loss probability when *k* is 4 paths for the Travel Agency Algorithm (TAA), result of the high connectivity of this network. Starting at the 12 data channel scenario, C^3-OBS shows already an order of magnitude increase compared to OBS. This efficiency increase augments slowly with the growth of the number of available data channels.

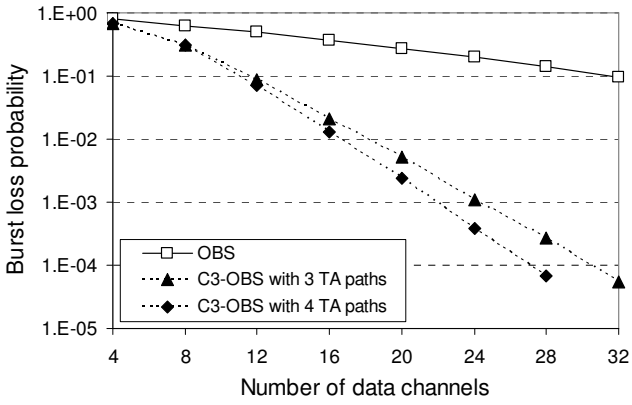


Figure 109 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (full scaled links).

Also visible is the gain when the number of possible paths increases. This is an expected result and although it needs further research, it suggests that the number of maximum possible paths may be defined by the nodal degree of the node and the networks mean nodal degree. The increase from 3 to 4 possible paths for the TAA is not very significant - for 32 data channels the 4 path simulation value drops bellow 10^{-5}, showing an increase of about one order of magnitude when compared to the 3 path simulation. After analysis of the routing tables built by the simulator, we found that the deeper we searched for paths on the topology, these tended to involve more nodes (more hops) and be bigger, thus using more resources on the network and thus the positive effect of routing yet another burst through the fourth possible path had the disadvantage of using more nodes and links for that burst, resulting in other burst losses.

A homographic EON network with all links set to 1 km in length was simulated and its results are shown in Figure 110. We can see an interesting effect when we compare the results from this simulation scenario with the results shown in Figure 109. Here the effect of the number of searchable paths is stronger than in the case of the full length link scenario. Also, comparing the plots from Figure 109 and Figure 110, we can see that OBS performs best in the short fixed link scenario, while C^3-OBS has an opposite behaviour – it performs best in the full length scenario. These opposite trends sent us to the analysis of the log files again. There are two main reasons to the opposite trends: firstly, part of the improved results for the EON with long full scale length links are caused by the buffer effect of long links [57, 58, 199]. Secondly, the opposite trends on OBS and C^3-OBS for full-length and short-equal length topologies is explained by the routing algorithms implemented in the simulator: in OBS, routing is done following a path length criterion. As the links are all equally lengthy, the load of the links is spread in a more homogeneous way, *i.e.*, the links are occupied with a more homogeneous load. Moreover, as links are equal, the choice of paths is enhanced, because to qualify to a given destination, there are more equal cost candidates, which justifies the improved result for the 4 paths plot on the equal-short-length links simulation.

In the full length simulation, the links are taken in scale resulting that some of the shortest links will be occupied more than the others, because they are part of a larger

number of shortest paths. Thus the implementation of blind routing decisions (*e.g.* OBS routing) brings an increased load on these links, causing contention and bursts drop, while well informed routing decisions (*e.g.* C^3-OBS routing) acknowledges overloaded links and nodes and thus routes bursts through alternative paths, resulting in increased performance.

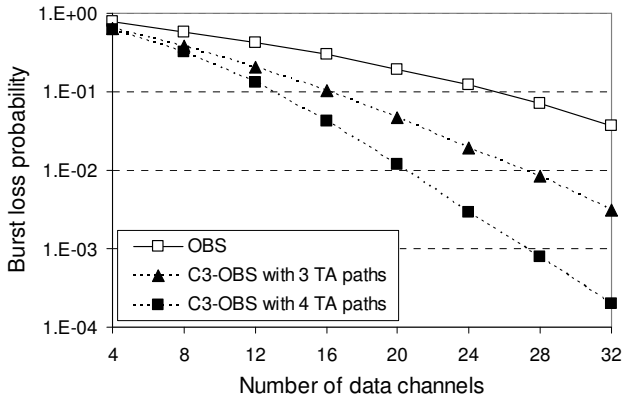


Figure 110 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (1 km links).

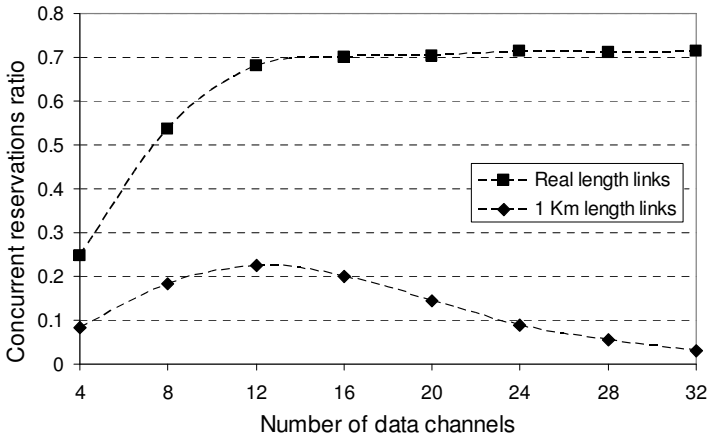


Figure 111 – Concurrent Reservation occurrence probability for 19-node EON with full-scale links and with short equal-length links of 1 km.

The long link effect also has its impact on the number of Concurrent Control Packet events (see Figure 111). With long networks, the t_{WI} is only reached for transmissions which have short path length and thus in many cases the network will

make poorly informed routing decisions. An additional proof to the memory effect mentioned previously was done by changing the links lengths in the full link length simulation. For this scenario, the EON was redefined by making all the links shorter 100 times, *i.e.*, the larger link was defined as being 16 km in length. Figure 112 shows the simulations results for this scenario, including the results previously presented in Figure 109 for comparison purposes. We can see that the performance of the shorter scale EON network decreases by about two orders of magnitude for the 32 data channels simulation scenario, being the sole cause of this decrease the shortening of the link lengths.

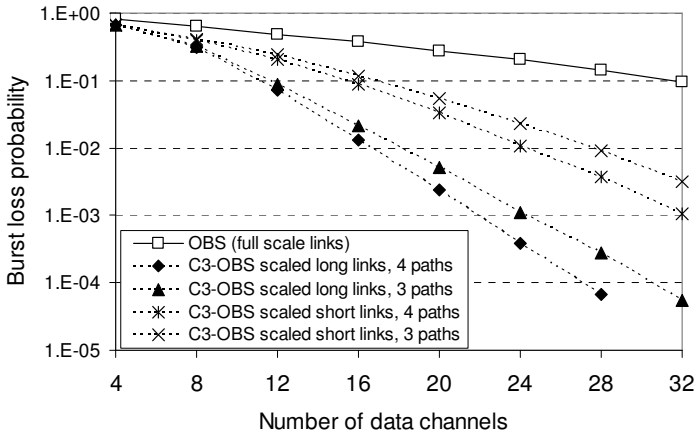


Figure 112 – Burst loss probability versus number of data channels for 19-node EON topology for OBS and C^3-OBS (scaled reduced links and scaled full length links).

The 14-node NSFnet topology was simulated for 4 paths considering full scale links. Results are plotted in Figure 113. We can see that for 32 data channels, C^3-OBS shows a performance increase of more than 2 orders of magnitude compared to OBS. NSFnet was not simulated with 3 paths for the TAA, neither with reduced or fixed link lengths.

The results for the simulations regarding different thresholds for the allowed Departure Horizon (DH) are shown in Figure 114. The topology used was the EON network with the number of searchable paths for the TAA set to 4 and all links equal to 1 km. Four different thresholds were used – 0, 5, 10 and 20 ms, meaning that, counting from the time of the requested burst departure, the TAA searched for resource

availability up to the delay defined as Departure Horizon, *i.e.*, bursts were allowed to depart with an additional small delay. This delay greatly enhances the performance of the TAA, since the longer it takes the burst to depart, the better informed the network is and thus the better the routing decisions are made.

We can see from the plot trend in Figure 114 that burst loss drops to values very close to zero for scenarios where the number of data channels is bigger than 16 (zero values are not shown in the graph).

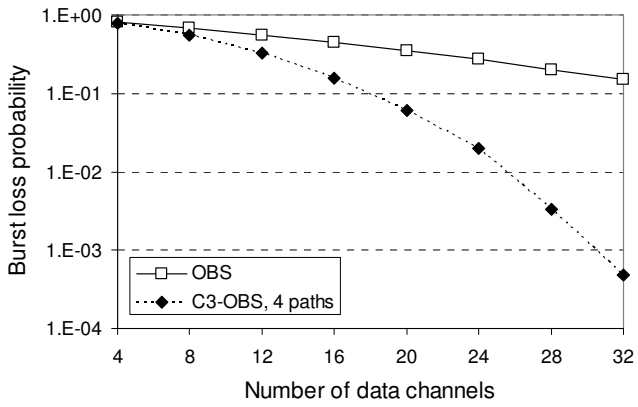


Figure 113 – Burst loss probability versus number of data channels for 14-node NSFnet topology for OBS and C^3-OBS (scaled full length links).

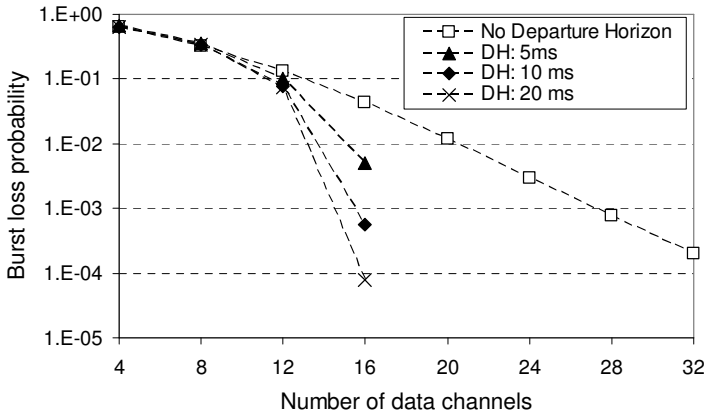


Figure 114 – Burst loss probability versus number of data channels for EON topology for C^3-OBS for different Departure Horizon delays (1 km length links, 4 searchable paths).

6.6.2. Burst delay in OBS and C^3-OBS real topologies

Also important is to assess how delayed the bursts are both in long and short topologies. We simulated burst delay for the EON network (see Figure 108), the NSFnet (see Figure 27), both with full long links. As a mean of comparison, we add delay values for a 4x4 Mesh Torus network with 1 km long links. Burst delay is counted from the moment the CP of the burst enters the network, until the time the burst exits, so according to (28) burst assembly time is partially accounted in these results (at the least).

The assessment of these values for OBS networks was initially published in [140]. Figure 115 shows that for the scarce resource simulation scenarios (up to 32 data channels for NSFnet, 24 data channels for EON and 20 data channels for Mesh Torus 4x4), the mean transit time per burst stabilizes, independently of the burst drop probability (see Figure 109 and Figure 113 for the burst loss rate for the EON and NSFnet topologies), *i.e.*, between the minimum and the maximum value in the 32 wavelength scenario and the 72 wavelength scenario range, the transit time changes about 0.79% while in terms of performance for that range, the burst drop probability decreases over 3.5 orders of magnitude. The shorter transit times for bursts when the networks are heavily loaded are explained as we know that in these scenarios only bursts with very short paths are transmitted [55] and short paths imply short transit times.

The mean link length for NSFnet is considerably larger than the mean link length for EON or 4x4 Mesh Torus– approximately 12.45 ms for NSFnet, 3.06 ms for EON and only 0.05 ms for Mesh Torus. The mean burst transmission times depicted in Figure 115 show almost the same offset between burst transmission times: there is (approximately) a 20 ms offset time for NSFnet, a 24 ms offset for EON and a 20 ms offset for 4x4 Mesh Torus. This offset is related to the burst assembly period and the CP offset time, that is considered here as part of this measurement. The longer offset for EON is a result of the topology: EON is more connected than NSFnet, and there are many comparatively short links, thus causing for a higher usage of short links in the

selected paths, with the drawback that the usage of several shorter links also means the use of an increased number of hops.

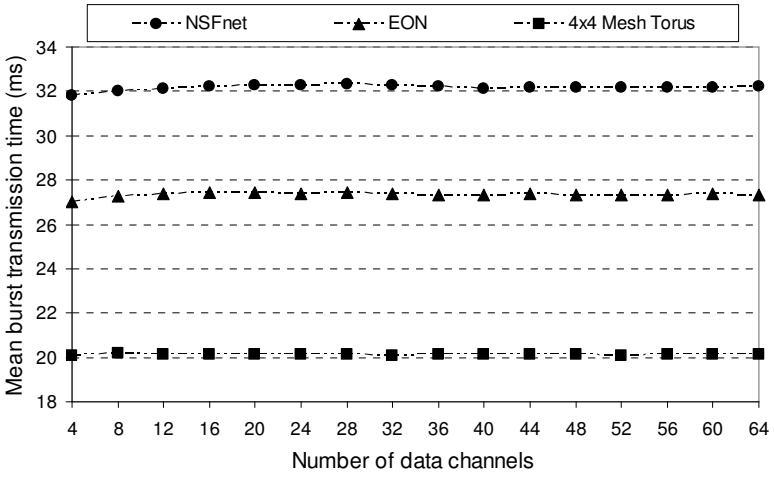


Figure 115 – OBS network mean transmission time for a burst versus number of available data channels in 4x4 Mesh Torus, 19-node EON and 14-node NSFnet topologies.

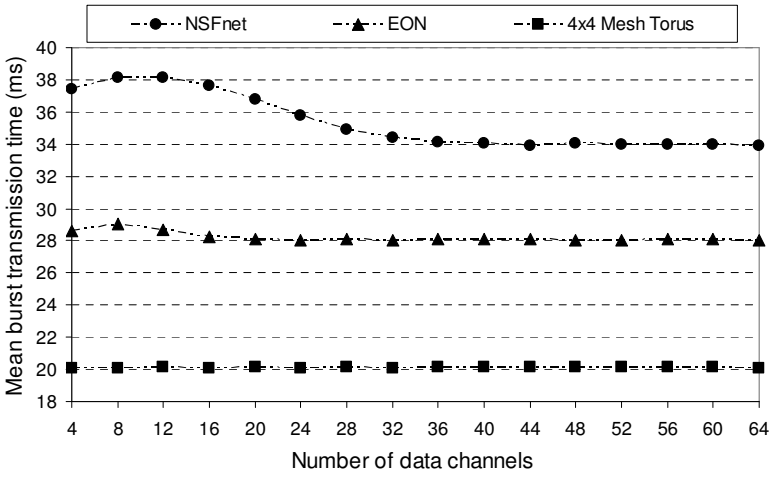


Figure 116 – C^3-OBS architecture mean burst transmission time versus number of available data channels for a burst in 4x4 Mesh Torus, 19-node EON and 14-node NSFnet topologies.

Figure 116 shows the average delay experienced in C^3-OBS by bursts traversing the EON, NSFnet and the 4x4 Mesh Torus networks, in a similar scenario depicted in Figure 115 for OBS, *i.e.*, the burst delay time also includes burst assembly time. The

C^3-OBS burst delay chart has three very interesting differences when compared to OBS burst delay.

The first one is related to the increased delay the burst experience when network resources are scarce, *i.e.*, for a low number of available wavelengths, clearly visible for the NSFnet plot. The explanation to this behaviour is immediate from the simulator log observation: C^3-OBS selects burst routes not according to shortest path only, as OBS does, but selecting the first available path from the set of the *k* shortest paths. This results that initially, while the network is overloaded, more bursts are routed through longer paths and this accounts for longer transit times. Moreover, these bursts would be otherwise dropped in OBS and thus are not accounted in mean burst transit time. This explains the increase for the transit time start for NSFnet and much more lightly for EON, as this is more connected network and its mean path length is shorter than for NSFnet. For 4x4 Mesh Torus it would be strange to see the plot following NSFnet and EON behaviour as this is a highly connected network with very short paths and thus the *k*-shortest paths will all have a very similar length, a conclusion that is sustained by the fact that for this network all links are defined as being 1 km long, *i.e.*, short and homogeneous. Thus, for 4x4 Mesh Torus, the effect caused by a longer path selection when network is overloaded is not visible.

The second characteristic is the longer transit time of the bursts when the network is working at a stable condition, comparing C^3-OBS with OBS, stable condition being, *e.g.*, at least around 36 available wavelengths in the simulations for Figure 115 and Figure 116 for all topologies. As may be seen in these figures, from OBS to C^3-OBS the burst transit times increases: for the 4x4 Mesh Torus topology there is no visible increase, but for NSFnet there is a 5.3% increase in average. Again, this is expected as a consequence of the routing algorithm implemented by the Travel Agency Algorithm in C^3-OBS, *i.e.*, bursts are routed without concerns of shortest path selection, but rather with concerns of more efficient path selection.

The third one is observable in Figure 115 and Figure 116 when the networks are overloaded (with few available wavelengths). In Figure 115 we see that the burst delay increases until it reaches a stable value, a characteristic that is well observable for the

NSFnet and EON plots. This is a result of the OBS architecture: when the network is overloaded, only the bursts that have shorter paths will succeed and thus this results in a lower mean burst transmission time. For C^3-OBS there is an opposite trend: when the network is overloaded, the burst delay times are higher than when the network is working at a more loss free scenario. This is a consequence of the routing strategy for C^3-OBS, as the bursts that would be dropped now have the opportunity to succeed at the cost of using a longer path, something that is clear from the observation of the simulator log file. The need to use longer (detoured) paths decreases as the networks has more free resources, thus causing the burst transmission times to also decrease.

6.6.3. NAN routing for OBS and C^3-OBS

Next Available Neighbour (NAN) routing algorithm [14] was described in section 3.4.2. Following the conclusions of previous sections, when the availability of resources in the network is scarce, the effort of pushing a burst a node further by the application of the NAN algorithm, will cause other bursts to drop, due to the imposed utilisation of the links. But when the availability of network resources increases, NAN routing will be able to use some of the free resources to recover bursts that otherwise would be lost.

As an example, we have simulated NAN routing for C^3-OBS for the 14 nodes NSFnet topology considering fixed link lengths of 1 km and considering full scale links.

Figure 117 shows the results for C^3-OBS with NAN and for C^3-OBS without NAN routing. The results for NAN have a very interesting and expected characteristic: NAN routing is detrimental for the performance of the network when the network is saturated with traffic, a feature that is visible when the network has from 4 up to 20 data channels. This behaviour was also observable for NAN in OBS networks, whose results are shown in Figure 43. When the network is not saturated (at burst loss probabilities below 1%), NAN increases the performance of the network, being the trade-off the increased burst delay (already presented in section 3.4.2).

Figure 118 shows the simulation results for NAN routing for C^3-OBS for the 14-node NSFnet topology using links with sizes in real scale. We can see that the

behaviour for the performance of NAN routing agrees with previous results. Here, the complementary routing algorithm is detrimental when network resources are scarce (in this scenario for up until 24 data channels) and for a larger number of data channels, NAN routing improves the performance of the network, reaching one order of magnitude for 32 data channels. For 36 data channels, the burst loss probability for NAN routing drops to zero (not shown in the figure).

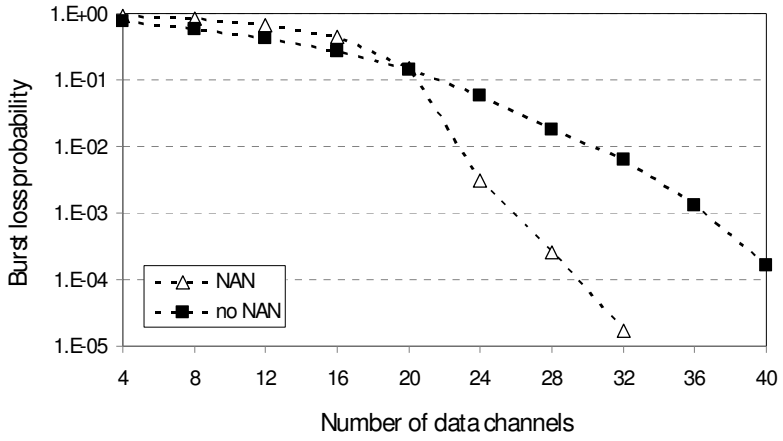


Figure 117 – Burst loss probability versus number of data channels showing C^3-OBS and C^3-OBS with NAN routing for the 14-node NSFnet topology (1 km length links).

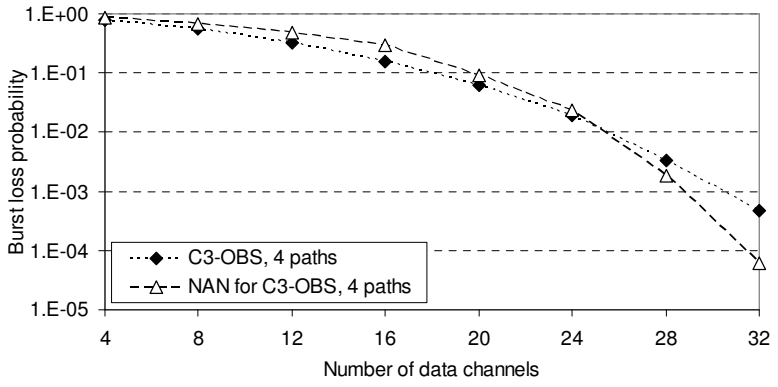


Figure 118 – Burst loss probability versus number of data channels showing C^3-OBS and C^3-OBS with NAN routing for the 14-node NSFnet topology (full scaled link lengths).

It is also interesting to observe that the point in the charts where NAN algorithm actually starts being beneficial to the network, changes from Figure 117 to Figure 118,

moving from the point where the network has 20 data channels to the point where the network has 24 data channels. The explanation to this fact is related to the better performance of C^3-OBS on networks that have long links, already presented previously as the link memory effect. So as C^3-OBS performs better with topologies with long links, there is little opportunity for NAN to improve the efficiency of the network.

6.7. Summary

The new proposal for the Common Control Channel Architecture is presented in this chapter. The common control channel concept is presented and the new C^3-OBS node functional scheme is discussed. The Local Network Model, Travel Agency Algorithm, network awareness degree concepts are presented and discussed. The new architecture scope and limitations are studied and the concept of C^3-OBS domains is proposed as a way to minimize some undesirable effects of control channel flooding. It is also discussed a proposal to use IP-PAC machines as ingress nodes for the C^3-OBS network and its integration with the C^3-OBS core nodes.

The description of the simulator designed to model the new architecture functionalities is also presented. Simulations for several topologies, both regular and irregular, both with links that were considered at full scale or with reduced fixed size are presented and the benefits from C^3-OBS are identified and associated with its specific features. We assess burst transmission delay in C^3-OBS as a mean to study the time a burst takes to travel in average in network topologies such as the EON or the NSFnet with full scale links. Finally, and complementing the data presented in section 3.4.2 we present the performance assessment of C^3-OBS using the next available neighbour complementary routing algorithm.

Chapter 7.

Final Conclusions and Future Work

This thesis presents several solutions that may be used to improve the performance and reliability of OBS networks. It is important to emphasize that OBS networks have known its share of glory and doom in the last (very) few years and very recently have again come forth to industry highlight as *startup companies* are presenting the first commercial OBS products.

Industry and Academia do not always walk hand-in-hand. Moreover, as Umberto Eco once wrote, one should never fall in love with its own Zeppelin, *i.e.*, one should never assume that this specific solution, albeit the best, is going to make it or be successful in the market. It is well known that best technologies are not always better technologies, in the sense of market relevance.

OBS is a viable technology (as opposed to OPS), interesting from OPEX and CAPEX points of view (as opposed to SONET/SDH and copper networks), fast, versatile and scalable. It has all the requirements to make a successful market technology, but telecommunications market is weary, stressed in every aspect: on one side, customers press for lower prices and larger bandwidth; on other side, regulators stress for better customer-experienced performances, larger territorial coverage and sometimes heavy taxes and fares; from another standpoint, competition is fierce and more global each day. Finally, on another side, research and equipment manufacturers deploy an enormous range of technologies, some of which never will reap the benefits of mass production. So it takes extreme caution and paramount knowledge of all the interests of all stakeholders to try to figure out what the market will best buy.

This thesis was focused on specific solutions to known problems. These solutions are bound coherently in an end-to-end OBS solution integrating an industry interest perspective and an academic approach, with or without the use of common

control channel architectures. Throughout this thesis we have addressed questions that were posed before the PhD research even started, such as “how long will a burst be” or “how long will it take to cross a network” or even “what is really the best burst assembly algorithm”, amongst others questions that surfaced in the course of this research programme, some of them still unanswered.

Without falling too much in love with this Zeppelin, there are trends that confirm this could be the right path: C^3-OBS was invented one July afternoon in 2004 and was first submitted to the European Patent Office later on that year. Its main innovation relies on knowing what is happening in the network to better manage its resources, using a strategy that relies on the management of distributed information. Recently, the need to make decisions at all levels of network elements, the implementation of intelligent agent based network management, the use of information dissemination and of distributed and cooperative routing on several types of networks and so on, seem to be in the foundation of new generation network technologies.

7.1. Final conclusions

Throughout this thesis we have presented the study of OBS networks, starting at the tributary client traffic at the ingress edge node, to the routing and switching algorithms in the core nodes, evaluating burst assembly algorithms, routing algorithms and architectures.

In Chapter 1, we have presented the motivation, the scope and the problem statement for the research. The main drive for fast and versatile optical switching is the increasing demand of bandwidth from users at a rate that is not met neither by the capacity of the switching technology nor the ability to implement full optical packet switching. Thus OBS appears as a technical compromise to deliver high speed data to users while reaping the benefits of the underling wavelength division multiplexing implemented in the optical fibres.

Chapter 2 presents the state of the art of optical burst switching networks, showing a detailed analysis on current architectural proposals that implement

independent network management or centralized management and discussing the main resource reservation protocols and schemes, with a special detail on the one way signalling and resource reservation protocols. This chapter also addresses the main approaches for contention resolution in OBS, and the new proposals for IP over WDM, IP over OBS and TCP over IP.

Chapter 3 is devoted to routing algorithms in OBS networks. It starts by detailing the main routing algorithms and strategies for OBS, including its classification as dynamic or static algorithms. We show that the Dijkstra algorithm, the *de facto* standard for route planning in networks, does not perform well in networks that contain rings with an even number of nodes, a scenario that is common in real optical network topologies. This problem led to the creation of a new routing algorithm, static and still returning the shortest path, called Extended Dijkstra routing algorithm. To evaluate the merits of this new routing algorithm we present a performance assessment including a performance of the Dijkstra algorithm using two new metrics: routing balance and routing symmetry. We show that the new Extended Dijkstra algorithm, while keeping all the advantages of the Dijkstra algorithm, is more efficient in static route planning than its predecessor. We evaluated the impact of the change of routing algorithms through simulation and observed a large performance improvement for some scenarios when the new Extended Dijkstra algorithm was used instead of the Dijkstra algorithm. We found that the use of Extended Dijkstra algorithm results in a near zero loss scenario when the initial burst loss probability was at around 1% for the Dijkstra algorithm on the eight node ring plus one central node. For the COST 239 network topology, the change in the routing algorithm results in the decrease on the burst loss probability of near an order of magnitude for the Extended Dijkstra for a network with 24 available data channels. For all the topologies considered in this chapter, Extended Dijkstra algorithm shows, in the overall, a more balanced and symmetric traffic route distribution scenario than the Dijkstra algorithm.

Chapter 4 addresses the issues of the significance of using IPv4 or IPv6 as tributary traffic and of burst assembly efficiency. We proceeded to analyse IPv6 traffic. But as IPv6 traffic is not yet as widespread and available as IPv4 traffic and we needed to assure that the results were valid with both versions of the IP protocol, we

remanufactured real previously recorded IPv4 traffic to IPv6, following two different approaches. In the first approach we merely replaced the IP header and resized the new packet according to the inferred MTU. We found that for some traces, the change in the IP protocol can result in an increase of up to near 45% more packets being transmitted in the network. In the second approach, we tried to deduce the packet generation ratio at the source machines. We found that packets were generated at around 400 micro-seconds, and that if the size of IP packets was not limited by the 1500 B boundary, a probable heritage from the underling Ethernet transport, *i.e.* if the size of the packets could be up to 64 KB, it would suffice to generate packets of 9 KB to have a decrease of up to 50% in the number of packets being transmitted through the network. As it is improbable that Ethernet restrictions will disappear, and given the fact that the traffic profile was mostly shaped by the underling Ethernet frame size restrictions, we concluded that IPv6 traffic will have a similar behaviour as IPv4, except possibly in the number of transmitted packets. In this chapter we provide an answer to one of the initial questions of this research, namely, whether it would be possible to use bursts that would size up to 4 GB, which is the size of an IPv6 Jumbogram. We found that, even for tributary traffic from high speed links, bursts as big as 1 MB could take up to almost one second to assemble and thus can only be used in traffic that is not time sensitive such as low priority email, Internet newsgroups or server backup traffic. We also assessed the performance the three main burst assembly algorithms for several burst assembly scenarios, using different thresholds for different tributary traffic inputs, and concluded that the performance of the burst assembly algorithm depends on the characteristics of input traffic. We found that for efficient burst assembly algorithms, in average the mean packet delay in a burst is of about 1 ms for burst sizes of up to 9 KB. This chapter ends with the evaluation of the significance of the burst loss, packet loss or byte loss metrics for JIT and JET OBS and we conclude that, using efficiency concerned burst assembly algorithms with real tributary traffic and despite of the very heterogeneous characteristics of the generated bursts, all of these metrics are similar and therefore equally significant. We explained this observation using the Law of Large Numbers.

The machine concept for the IP Packet Aggregator and Converter, IP-PAC, was presented in Chapter 5. The proposal of this new machine is done from a generic point

of view, not necessarily related to OBS. We defined the concept for the burst assembly machine and proposed the encapsulation of the packet aggregates inside a new IPv6 packet as a manner to improve the homogeneity of the switching resources in the network. Following the conclusions in Chapter 4, we proposed a new burst assembly algorithm, which dynamically adjusts the burst assembly thresholds. We defined the working scenarios for this machine, including the scenarios where an IP-PAC machine needs to communicate to a non-IP-PAC machine. We have evaluated the performance of the machine concept with its new burst assembly algorithm, including its contribution to the resolution of bottleneck problems. We found that the encapsulation of the bursts in an IPv6 header may result in up to 12% overhead in the number of transmitted bytes, yet, as the packets are grouped together using a single framing space, it still results in a channel occupancy decrease.

Chapter 6 presents the main contribution of this thesis, the new Common Control Channel OBS network architecture. This new architecture implements a distributed control method through the automatic dissemination of the control packets in the network, which in turn allows for a local network model of the network to mimic the expected behaviour of the network in a close future time. The degree of network awareness of the network is studied and defined and the concepts of Local Network Model, Common Control Channel and the Travel Agency Algorithm are detailed. This chapter also presents the simulator used to estimate the performance of this new architecture compared with the OBS architecture, for regular and irregular network topologies. We concluded that the C^3-OBS architecture shows a significant decrease in the burst loss probability, *e.g.* a decrease of two orders of magnitude for 32 available data channels for the 14-node NSFnet topology with full scale links. We assessed the average time needed for a burst transmission in C^3-OBS, being around 32 ms for the 14-node NSFnet with full scale links and 27 ms for the 19-node EON topology. We presented the Departure Horizon concept, were a burst may be temporarily delayed at the ingress edge node if the resources are not available. For a Departure Horizon of 5 ms, we observed that the burst loss probability drops to zero when the number of available data channels is bigger than 16 for the EON topology. The concept of C^3-OBS network domains as a mean to keep the dissemination of the control packets to a lesser geographical extent is also presented. The chapter ends with the study of the

performance of C^3-OBS networks using the Next Available Neighbour routing and the estimation of the average travel time for a burst in the full scale length network topologies in the EON and NSFnet networks. NAN routing causes burst loss to drop to zero when the number of channels is bigger than 32 data channels for the NSFnet 14-node topology.

7.2. Future work

There are issues in the area of resource reservation protocols that may benefit from further investigation. It has been shown that JIT^+ and E-JIT have better performance than other more complex protocols. This is a consequence of the simplified database structure that these protocols use and of simplified reservation tasks, *e.g.*, no void filling. But just as JIT^+ extended JIT by adding a second possibility to reservation, what will be the result if we pursue this line of reasoning a bit further? What would be the implications for an E- JIT like algorithm if it would allow 3, 4, 5 or 10 reservations? Where is the boundary for database complexity here?

Also, E-JIT proved to be more efficient when the network operates with smaller bursts, because in this case, the channel timings are used more efficiently. What would be E-JIT performance for real traffic and using efficiency concerned burst assembly algorithms? And again, is there room to improve in an E-JIT like manner the remaining protocols?

Other questions rise when we address the performance of signalling and resource reservation protocols. For example, it is widely assumed that JET and other void-filling protocols have longer setup and OXC times due to its higher inherent complexity. But exactly what is the amount of simultaneous reservations that are handled by these protocols? Or in other words, it is important to assess the size of the database associated with the operation of these more complex protocols to allow a more adequate estimate of the times that are associated with its operation.

Although several authors have addressed this issue, mainly while studying burst assembly algorithms, it would be interesting to assess the behaviour of an OBS network

with really large (although possibly not realistic) bursts. Or in another words, would it be beneficial to define an OBS path / link MTU?

Another research topic claiming attention is the study of hybrid OBS/C^3-OBS networks. Also worthy of research is the use of Hamiltonian paths and circuits in the control topology and its performance assessment from a burst loss and from a burst delay point of view.

From another standpoint, the number of *k* selectable paths for the Travel Agency Algorithm should be related to the network mean nodal degree, or at least, to the interested node nodal degree. This is another issue that is closely related to the optimization of the performance of the Travel Agency Algorithm.

On the use of NAN routing, we still need to evaluate the performance of C^3-OBS networks using cumulatively NAN routing with Departure Horizon parameters, as they are not incompatible. On OBS, we would also like to address the performance assessment of networks using other routing schemes such as FAR or Deflection Routing cumulatively with NAN.

Finally, we can apply a new Quality of Service mechanism by allowing different bursts to depart with different Departure Horizons, making this yet another area worthy of research.

Annex A

OBS simulator architecture

This annex describes with detail the architecture of the simulator build to assess the performance of the C^3-OBS architecture and of the new routing algorithms, Next Available Neighbour, the Extended Dijkstra and the Local Network Model.

Classes in the OBS Simulator are divided into two packages, both described as follows:

1. the *Network* package classes model the physical components of simulated network. These classes are (see Figure 98):

    - *Network* class – the root object for the network components, its role is mainly organizational. One instance of this class is created (per simulation) and it is used to hold parameters of the network (*e.g.* number of nodes, signalling algorithm used and resulting control packet process time, etc.), references to all other components (nodes, links, users, etc.) and statistics.
    - *Node* class – represents nodes of the network, with its characteristics, its role in the network (edge node / core node), burst assembly parameters and so on. *Node* also holds references to *links* and *users.* It also holds a number of statistics, which are updated as the simulation progresses. The *Node* class is virtual so its not instantiated. Instead, the four classes that inherit from *Node*, including *NodeJIT*, *NodeHorizon*, *NodeJET* and *NodeJITplus*, represent nodes with different signalling algorithms and are instantiated in the simulation.
    - *User* class – represents machines connected to nodes, which generate traffic. Users generate random traffic following an exponential distribution, or generate traffic based on trace files, (as described in section 4.3.1 – Real IPv4 traffic). *User* traffic generation is configurable,

i.e. it is possible to define the average delay between transmitting bursts and the average burst size, each of these following exponential distributions as well.

- *Link* class – represents the fibre connection between nodes. The characteristics of each link are the number of wavelengths used as data channels, the link length, represented by its propagation delay and the channel data rate of the link. The class also contains statistics for resource information (reservations made for the given link) and for link load. The *Link* class generalizes the *LinkJET* class, which has different structures to hold information about reservations, as the JET algorithm is the most complex of signalling algorithms.
- *ControlChannel* class – it is similar to the *Link* class as it represents the connections between nodes and specifically describes the Common Control Channel topology in C^3-OBS architecture, as the topology for the control channel does not necessarily coincide with the topology of the data channels.
- *TSHPacket* class – used only in case of real-traffic-feed from *tsh* files. Given a *tsh* file, it creates an object that contains the packet information for each record in the trace file.
- *Parameters* class – this class holds static variables which, as the class name states, are the general parameters of the simulation. Most of them are set directly in the invocation part of the simulation, accessed from the *main* function. It includes parameters such as file paths (the topology file, traffic traces files, output files and so on), counters and flags on some of the technologies used in the simulation (*e.g.* on the routing algorithm used, *i.e.* Dijkstra versus Extended Dijkstra).
- *Simul* class – the main class of the Simulator. Once a *Simul* object is created, the Simulator reads the input data from the configuration files and instantiates the necessary objects. It also creates the event list and starts processing the events. These tasks will be described in detail forward.

- *Start* class – contains a static *start* method, which is the first method to be called in the program run. With each simulation represented as an object, it is possible to run multiple simulations, with equal or different parameters using only one program call. This class is also described forward.

2. The Event List package holds classes representing different types of events, information necessary to process them and building the event list itself. These classes are (see Figure 99):

   - *List* class – extends TreeMap Java class and uses two long type numbers to create a unique time stamp for each event held in the list (first is the event occurrence time in picoseconds, second is auto incremented number, to distinguish two events scheduled for the same instant). Implementing TreeMap guarantees logarithmic access time to both extraction and addition of events.
   - *Event* class – this is a general class used to represent all types of events, responsible of the simulation process itself. An instance of the Event class is placed on the event list and represents a predefined scenario. All types of events have common characteristics that are implemented in the Event class, including the type of event and the time of occurrence.
   - *Arguments* class – abstract class, used to represent the arguments for all types of Simulator events, not included in the main arguments in the *Arguments* class of the *Network* package. Apart of the parameters specific for each event described above, different types of events hold parameters specific only for themselves. For example the arrival of a control packet is described by the number of node where the event takes place and the reference to object representing the control packet. The arrival of a C^3-OBS control packet is further extended by the information about which node was its source. For other types of events the arguments will vary and we specifically can distinguish classes like: *PacketTransmission*, *ControlPacketArrival*, *C3ControlPacketArrival*,

BurstAssembly, *BurstSending*, *BurstArrival* and *RandomBurstCreation*. All of these inherit from the *Arguments* class.

- *Burst* class – represents a burst transmitted in the network. A *Burst* consists of its control packet and a payload, which is a vector of packets created from instances of the *User* class.
- *ControlPacket* class – holds information about a single burst, which is used to make reservations along the path from source to destination. It also holds information on the size of burst, its path and number of packets. *ControlPacket* is extended by the *C3ControlPacket* class, which is contains C^3-OBS related information, *e.g.* the Local Network Model updates and parameters used for C^3-OBS features.
- *C3ControlPacketID* – serves as a unique identifier for each C^3-OBS control packet created in the simulation process, containing the number of node that issued the CP and a sequential number. The purpose of this is to distinguish between redundant CPs as presented in section 6.2 - Architecture of a C^3-OBS . The redundant CPs are discarded because the information they carry has already be reflected in the Local Network Model of the node.
- *LNMUpdate* class – used to define the updates of the Local Network Model at given node. Each update holds the following information: the reference to the link it concerns and the beginning and end of the time period in which the link is supposed to be occupied. Each time a burst is transmitted into C^3-OBS network the control packets traverses the network by the control channel, carrying the LNM updates. If the data in any of the carried LNM updates corresponds to the currently visited node, its LNM is renewed to represent the current state of the network.

The input data for the OBS Simulator consists of:

- the description of network topology – this represents the physical shape and dimensions of the network. A typical OBS network consists of users, which are connected to nodes. Following the OBS interpretation, the nodes are the aggregating and/or switching machines and are connected with each other by

optical fibre. The topology description includes the quantities and parameters of above mentioned, amount of users connected to each node, connections between nodes, link bandwidth and propagation delays;

- the routing and management algorithms used – since the Simulator is capable of simulating different algorithms, it is necessary to specify which ones are included in the simulation scenario. This means in particular Extended Dijkstra routing algorithm, Fixed Alternate Routing, C^3–OBS architecture, Travel Agency Algorithm routing, Nearest Available Neighbour routing and also the parameters for these options;
- the characteristics of the traffic produced by users – this describes the traffic produced in the users which is the base for bursts created in the simulation. Possibilities include random traffic, trace files-based or mixed generation.

The network description is kept in a topology file. Now follow some of the conventions adopted and formatting used in this file. All time values are expressed in picoseconds units (10^{-12} second) to allow a more detailed concurrent event control. Lines starting with #, with a space or blank lines will be treated as a comment and discarded while parsing the file. Nodes in the Simulator are numbered in ascending order, following the configuration file, starting from `1.` with the following structure:

- The word `network:` followed by five lines representing general network parameters:
  - Optical cross-connect time (T_{OXC}) – an integer value representing the time it takes for the cross-connect mechanism to switch. Assumed to be equal for each node in the network.
  - Time to process control packet (T_{Setup}) – an integer value representing the time necessary to process a control packet, depending from the signalling algorithm used. Also assumed to be the same for each node.
  - Signalling algorithm – an integer value in the range of 1 to 4. The following algorithms and corresponding integer values have been implemented in the Simulator: 1 – JIT, 2 – Horizon, 3 – JET, 4 – JIT^+.

- Number of nodes in the network (n) – an integer value.
- Number of traffic classes – an integer value.

- n sets of parameters, starting with the word `node:` and representing each node in the network
  - The word `label:` followed by a text value in a new line, representing the description of the node. This parameter is optional and if not desired, should be omitted along with the word `label:`.
  - The word `coordinates`: followed by two integer values in a new line, representing the position of the node when displaying the topology in a graphic. This parameter is also optional, similarly to label of the node.
  - The name of the folder holding traffic trace files – a text value representing a path to a folder with *tsh* trace files for the current node.
  - Number of users – an integer value describing the number of users connected to given node. This value also contributes to the representation of the intensity of traffic generated by the given node. It is assumed that all users generate the same amount of traffic and thus the amount of generated traffic is proportional to the number of users.
  - Burst assembly time – an integer parameter for the burst aggregation scheme. The assembly time represents a cycle after which IP packets should be aggregated into a data burst and injected into the network.
  - Burst assembly size – an integer value, second parameter for the burst aggregation scheme. This value represents (in bytes) the maximum allowed size of a burst. These two values represent the thresholds for the HA algorithm presented in section 4.2.3 – Hybrid Assembly.
  - Description of links – a set of lines, one describing each link connected with the given node. They are described with following

integer parameters separated by a single space: number of node that the link leads to, number of data channels per link, propagation delay and bandwidth per single wavelength (in MBits per second).

After links description, if a word `node:` is encountered again, a new node description is assumed. This continues until the file encounters the word `end`. After parsing the file, a network model is built, as described forward. Table 13 shows a sample configuration file for a four node ring network.

Table 13 – Sample network configuration file for the 4 node ring network.

```
# This is network topology description file
# Each line starting with a hash or a space will be treated as a
comment,
# blank lines will be discarded

network:
# T oxc
10000000000
# T setup
12500000
# signalling algorithm 1:JIT 2:Horizon 3:JET 4:JIT+
1
# nodes count
4
# traffic classes
4

node:
# number 1
coordinates:
10 10
AMP
# users
2
# burst assembly time
34300000000
# burst assembly size
650000
links:
# number of node, number of wavelengths, delay, throughput per
wavelength (Mbit per second)
2 1 5000000 1024
4 1 5000000 1024

node:
# number 2
coordinates:
20 10
FRG
# users
2
# burst assembly time
34300000000
# burst assembly size
650000
links:
```

```
# number of node, number of wavelengths, delay, throughput per
wavelength (Mbit per second)
1 1 5000000 1024
3 1 5000000 1024

node:
# number 3
coordinates:
20 20
MRA
# users
2
# burst assembly time
34300000000
# burst assembly size
650000
links:
# number of node, number of wavelengths, delay, throughput per
wavelength (Mbit per second)
2 1 5000000 1024
4 1 5000000 1024

node:
# number 4
coordinates:
10 20
FRG
# users
64
# burst assembly time
34300000000
# burst assembly size
650000
links:
# number of node, number of wavelengths, delay, throughput per
wavelength (Mbit per second)
1 1 5000000 1024
3 1 5000000 1024
End
```

The general simulation parameters are encapsulated in the parameters class, as public static variables. These serve to flag various simulation parameters and to set values for numerical parameters of the simulation.

In detail they include:

- *UserAverageIdle* – a double type value, expressing the average time (in milliseconds) between burst issued by a single user. This parameter is used with the random traffic generation scenario. Note that the average time corresponds to a single user, not to a node. Hence, when set for example to 1 ms, a node with 64 users will issue in average 64 bursts each millisecond.
- *userAverageBurstLenght* – a double type value, expressing the average length of each burst issued. The length is understood as a time in

milliseconds necessary for a burst to travel one hop in the network. This value and the UserAverageIdle serve as average values for the exponential distributions which describes the time gap between two bursts and the length of the burst.

- *topologyFile* – a string value, representing the path to the network description file.
- *controlChannelFile* - a string value, representing the path to the C^3-OBS control channel description file.
- *C3Architecture* – a Boolean type value, flagging the usage of the C^3-OBS architecture.
- *useTravellAgency* - a Boolean type value, when set to true the Travel Agency algorithm will be used in the simulation.
- *numberOfPathsForTravelAgency* – an integer value, representing the number of different paths to use with the Travel Agency routing algorithm.
- *pathSelectionScheme* – can be set to two integer values representing two different path choosing system for the Travel Agency algorithm. The value 0 corresponds to the constant value of TRAVEL_AGENCY_SMALLEST_DEPARTURE_TIME and uses the path that offers the shortest departure time for the burst. The value 0 corresponds to the constant value of The TRAVEL_AGENCY_SMALLEST_ARRI-VAL_TIME and instead of using the departure time, calculates the arrival time for each burst (departure time + the propagation from ingress node to the egress node) and uses the path that offers the shorter possible one.
- *longestAllowedReservation* – the C^3-OBS scheduling mechanism parameter, describes (in picoseconds) how long can be the offset between burst aggregation and burst transmission. Set to 0 means that immediately after the burst assembly the burst should be injected into the network, if the resources are available. When set to a value bigger than 0, means that it is possible to wait before burst transmitting, when no resources are available. This value is described as Departure Horizon in section 6.2 – Architecture of a C^3-OBS .

- *NANminimalPathPart* – the Nearest Available Neighbor parameter in a double type value, describing the minimum portion of its paths the burst has to successfully travel before it will be allowed to perform NAN routing.
- *NANnumberOfNANs* – another NAN parameter, of an integer type. Describes how many times one burst can perform NAN routing.
- *NANarchitectureEnabled* – a Boolean type value, when set to true the simulation will use the NAN routing technology.
- *C3WavelengthRoutingEnabled* – a Boolean type value, when set to true in case of a Concurrent Reservation scenario the burst will try to re-route itself to another available wavelength on the same link.
- *C3BurstReInjectionEnabled* – a Boolean type value, when set to true the bursts encountering a Concurrent Reservation scenario will be dropped and re-injected into the network.
- *randomDestinations* – a Boolean type value, when set to true allows to use traffic that is half trace file based and half random traffic based. The timestamps and sizes for all packets will be read from the trace files but the destinations will be randomly chosen from the available nodes (excluding the node which performs the burst transmission).
- *randomBurstCreation* – a Boolean type value, when set to true the simulation is be performed using a random traffic generation scenario.
- *NewDijkstraEnabled* – a Boolean Value, when set to true the routing is be performed using the Extended Dijkstra Shortest Path algorithm. When set to false, the original Dijkstra Shortest Path algorithm is used.
- *showDrawing* – a Boolean type value, when set to true with the end of simulation a graph with the topology and simulation results is written to a PNG file.
- *Logging* – when set to true enables various logging features, like logging of each event taking place in the simulation, the size of each burst created, the time between creation of any two bursts, etc. The files holding results include: ConsoleOutput.txt, NewMetrics.txt, a series of files Out***n***.txt, where ***n*** is the number of simulation.

- *setWavelengths* – an integer value, used to set an equal number of available data channels for each of the links in the network.
- *setWalengthConverters* – similar to the above parameter, but used to set an equal number of available data converters in each node; a value smaller than 0 means that full wavelength conversion is available.
- *setUsers* – similar to the two previous parameters, serves to set an equal number of users for each node.
- *eventList*, *process* – two Boolean type parameters, when set to false the first one prevents the creation of event list, the second one prevents the processing of the event list. When set to true, the simulation performs normally. Used for debug and test of the simulator.
- *simsNumber* – an integer value, describing how many simulation should be performed for each of the scenarios. The final result is the arithmetic average of each of the simulations results.
- *simTime* – a double type value, describing in seconds the length of each simulation scenario. This value corresponds to the simulated network functioning time, not the real simulation time.

In Table 14 one can see a sample output resulting from a short simulation run using the configuration for the data depicted in Table 13.

Table 14 – Sample simulation results for a short simulation run.

```
==================================
C3: true
topology: New Research//Ring4.txt
control channel: New Research//Ring4 cc1.txt
paths: 2
Extended Dijkstra: false

Wavelengths:      4
Burst loss: =14298/20535
Concurrent reservations:      678
Control packets processed: 26548
Average burst transmission time: 23.057525643 ms
Standard Deviation: 15.237007101 ms

Wavelengths:      8
Burst loss: =8608/20387
Concurrent reservations:      2101
Control packets processed: 50206
Average burst transmission time: 24.806394284 ms
Standard Deviation: 17.596196224 ms
```

```
Wavelengths:      12
Burst loss: =3768/20502
Concurrent reservations:      3026
Control packets processed: 71330
Average burst transmission time: 24.948893241 ms
Standard Deviation: 17.391337575 ms

Wavelengths:      16
Burst loss: =991/20443
Concurrent reservations:      2460
Control packets processed: 82800
Average burst transmission time: 24.676576105 ms
Standard Deviation: 17.030488347 ms

Wavelengths:      20
Burst loss: =61/20165
Concurrent reservations:      1171
Control packets processed: 85497
Average burst transmission time: 23.886221124 ms
Standard Deviation: 16.104363693 ms

Wavelengths:      24
Burst loss: =5/20883
Concurrent reservations:      512
Control packets processed: 88702
Average burst transmission time: 23.488787514 ms
Standard Deviation: 15.228059324 ms

Wavelengths:      28
Burst loss: =0/20559
Concurrent reservations:      123
Control packets processed: 87432
Average burst transmission time: 23.41268696 ms
Standard Deviation: 15.072876113 ms

Wavelengths:      32
Burst loss: =0/20434
Concurrent reservations:      37
Control packets processed: 86755
Average burst transmission time: 23.512920737 ms
Standard Deviation: 14.907592930 ms
```

References

[1] "The GÉANT 2 network". URL: http://www.geant2.net, last access in 18 September 2007.

[2] K. M. Sivalingam and S. Subramaniam, *Optical WDM Networks Principle and Practice*, 3rd ed. Kluwer Academic Publishers, 2000.

[3] Cisco Systems Inc. (2006), "Cisco XR 12000 and 12000 Series [Cisco 12000 Series Routers]", Cisco Systems. Inc. URL: http://www.cisco.com/en/US/products/hw/routers/ps167/products_qanda_item0900aecd8027c915.shtml, last access in 15 December 2007.

[4] P. E. Ross (2006), "5 Commandments - The rules engineers live by aren't always set in stone", IEEE Spectrum Online. URL: http://www.spectrum.ieee.org/careers/careerstemplate.jsp?ArticleId=n120403, last access in 14-05-2007.

[5] L. G. Roberts, "Beyond Moore's law: Internet growth trends", in *IEEE Computer*, vol. 33, No. 1, pp. 117-119, January 2000.

[6] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of Internet traffic in 1998-2003", in *Proceedings of Winter International Synposium on Information and Communication Technologies (2004)*, Cancun, Mexico, 2004, pp. 1-6.

[7] A. M. Odlyzko, "Internet traffic growth: Sources and implications", in *Proceedings of Optical Transmission Systems and Equipment for WDM Networking II*, B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato (Eds.), vol. 5247, 7–11 September 2003, pp. 1-15.

[8] A. M. Odlyzko, S. Hong, and A. Pakanati (2007), "Minnesota Internet Traffic Studies (MINTS)", University of Minnesota. URL: http://www.dtc.umn.edu/mints, last access in 30 December 2007.

[9] J. Postel, "Internet Protocol IPv4 Specification", RFC 791, Internet Engineering Task Force (IETF), 1981.

[10] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, Internet Engineering Task Force (IETF), 1998.

[11] J. J. C. P. Rodrigues, "Performance Assessment of One-Way Resource Reservation Protocols in IP over Optical Burst Switched Networks", PhD thesis, University of Beira Interior, Covilhã, Portugal, 2006.

[12] D. Borman, S. Deering, and R. Hinden, "IPv6 Jumbograms", RFC 2675, Internet Engineering Task Force (IETF), 1999.

[13] N. M. Garcia, P. Lenkiewicz, M. M. Freire, and P. P. Monteiro, "On the Performance of Shortest Path Routing Algorithms for Static Source Routed Networks – an Extension to the Dijkstra Algorithm", in *Proceedings of The Second International Conference on Systems and Networks Communications (ICSNC 2007)*, IEEE Computer Society Press, ISBN: 0-7695-2938-0, 6 pages, Cap Esterel, French Riviera, France, 25-31 August 2007.

[14] N. M. Garcia, P. P. Monteiro, M. M. Freire, J. R. Santos, and P. Lenkiewicz, "Next Available Neighbour (NAN) Routing for traffic aware networks", submitted, internal reference number 2006E18954PT, 29-08-2006.

[15] N. M. Garcia, P. Lenkiewicz, M. M. Freire, P. P. Monteiro, and J. R. Santos, "Next Available Neighbour Routing for Optical Burst Switching Networks", submitted to publication in a journal.

[16] N. M. Garcia, M. M. Freire, and P. P. Monteiro, "The Ethernet Frame Payload Size and its Impact on IPv4 and IPv6 Traffic ", in *Proceedings of The International Conference on Information Networking (ICOIN 2008)*, 5 pages, Busan, Korea, 23-25 January 2008.

[17] N. M. Garcia, P. P. Monteiro, and M. M. Freire, "The Effect of the *de facto* 1500 Byte Packet Size Limit on the IPv6 Traffic Profile - a New Argument for a Bigger Ethernet Frame", submitted to publication in a journal.

[18] N. M. Garcia, M. Hajduczenia, M. M. Freire, P. P. Monteiro, and H. Silva, "Method for Aggregating a Plurality of Data Packets into a Unified Transport Data Packet and Machine for Performing said Method ", european patent, filed in the European Patent Office, patent number EP 06.007.960.5, 18-04-2006.

[19] N. M. Garcia, P. P. Monteiro, and M. M. Freire, "Assessment of Burst Assembly Algorithms Using Real IPv4 Data Traces", in *Proceedings of The 2nd International Conference on Distributed Frameworks for Multimedia Applications (DFMA'2006)*, IEEE Press, New York, USA, ISBN: 1-4244-0409-6, Pulau Pinang, Malaysia, 14-17 May 2006, pp. 173-178.

[20] N. M. Garcia, P. P. Monteiro, and M. M. Freire, "Burst Assembly with Real IPv4 Data - Performance Assessment of Three Assembly Algorithms", in *Next Generation Teletraffic and Wired/Wireless Advanced Networking 2006*, Yevgeni Koucheryavy, Jarmo Harju, Villy B. Iversen (Eds.), *Lecture Notes in Computer Science*, LNCS 2003, Springer-Verlag, Berlin Heidelberg, May 2006, ISBN-10: 3-540-34429-2, pp. 223-234 (*Proceedings of the 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2006)*, St. Petersburg, Russia, 29 May - 3 June 2006).

[21] N. M. Garcia, P. Lenkiewicz, P. P. Monteiro, and M. M. Freire, "Issues on Performance Assessment of Optical Burst Switched Networks: Burst Loss Versus Packet Loss Metrics", in *Networking 2006 - Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems*, Fernando Boavida, Thomas Plagemann, Burkhard Stiller, Cédric Westphal, Edmundo Monteiro (Eds.), *Lecture Notes in Computer Science*, LNCS 3976, Springer-Verlag, Berlin Heidelberg, May 2006, ISBN-10: 3-540-34192-7, pp. 778-786 (*Proceedings of the 5th International IFIP-TC6 Networking Conference (NETWORKING 2006)*, Coimbra, Portugal, 15-19 May 2006).

[22] N. M. Garcia, M. Hajduczenia, P. P. Monteiro, M. M. Freire, and H. Silva, "The IP Burst Switching Networks Concept and its Performance Assessment", submitted to publication in a journal.

[23] M. M. Freire, N. M. Garcia, P. P. Monteiro, and J. M. R. R. Santos, "Method for the Transmission of Data Packets by Means of an Optical Burst Switching Network and Network Nodes for an Optical Burst Switching Network", International Patent WO 2006/072406-A1, 2006.

[24] N. M. Garcia, P. P. Monteiro, M. M. Freire, and J. R. Santos, "A New Architecture for Optical Burst Switched Networks Based on a Common Control Channel", in *Proceedings of Fifth International Conference on Networking (ICN 2006)*, P. Lorenz, P. Dini, D. Magoni, and A. Mellouk (Eds.), IEEE Computer Society Press, Los Alamitos, California, USA, ISBN: 0-7695-2522-0, 6 pages, Morne, Mauritius, 23-28 April 2006.

[25] N. M. Garcia, P. P. Monteiro, M. M. Freire, J. R. Santos, and P. Lenkiewicz, "A New Architectural Approach for Optical Burst Switching Networks Based on a Common Control Channel", *Elsevier Optical Switching and Networking - A Computer Networks Journal*, vol. 9, No. 4, 1573-4277, pp. 173-188, 2007.

[26] N. M. Garcia, P. Lenkiewicz, M. M. Freire, P. P. Monteiro, and J. R. Santos, "A New Architecture with Distributed Control for Optical Burst Switching Networks Based on a Common Control Channel", *submitted for publication in a journal*.

[27] N. M. Garcia, P. P. Monteiro, M. M. Freire, and J. R. Santos, "Frontier Node Architecture for the Implementation of Network Domains in Common Control Channel Optical Burst Switched Networks (C³-OBS)", submitted to publication in a journal.

[28] M. Yoo, M. Jeong, and C. Qiao, "A High Speed Protocol for Bursty Traffic in Optical Networks", in *SPIE's All-Optical Communications Systems*, vol. 3230, pp. 79-90, 1997.

[29] M. Yoo and C. Qiao, "A High Speed Protocol for Bursty Traffic in Optical Networks", in *Proceedings of IEEE/LEOS Summer Topical Meeting*, Montreal, Canada, August 11-15, 1997.

[30] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet", *Journal of High Speed Networks*, vol. 8, No. 1, pp. 69-84, January 1999.

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

[32] E. F. Haselton, "A PCM frame switching concept leading to burst switching network architecture", in *IEEE Communications Magazine*, vol. 21, No. 6, pp. 13-19, September 1983.

[33] S. R. Amstutz, "Burst Switching - An Introduction", in *IEEE Communications Magazine*, vol. 21, No. 8, pp. 36-42, November 1983.

[34] R. Rajaduray, S. Ovadia, and D. J. Blumenthal, "Analysis of an Edge Router for Span-Constrained Optical Burst Switched (OBS) Networks", *Journal of Lightwave Technology*, vol. 22, No. 11, pp. 2693-2705.

[35] A. Zapata and P. Bayvel, "Impact of burst aggregation schemes on delay in optical burst switched networks", in *Proceedings of IEEE/LEOS 2003*, IEEE, Tucson, Arizona, USA, October 26-30, 2003.

[36] B. Kantarci, S. Oktug, and T. Atmaca, "Analyzing the Effects of Burst Assembly in Optical Burst Switching under Self-Similar Traffic", in *Proceedings of Advanced Industrial Conference on Telecommunications*, vol. 1, IEEE, Lisbon, Portugal, July 17-20, 2005, pp. 109-114.

[37] G. Hu, K. Dolzer, and C. M. Gauger, "Does burst assembly really reduce the self-similarity?" in *Proceedings of Optical Fiber Communications Conference (OFC 2003)*, vol. 1, Atlanta, Georgia, USA, 23-28 March 2003, pp. 124-126.

[38] J. Luo, Q. Zeng, H. Chi, Z. Zhang, and H. Zhao, "The Impacts of Burst Assembly on the Traffic Properties in Optical Burst Switching Networks", in *Proceedings of International Conference on Communication Technology (ICCT'03)*, vol. 1, 9-11 April 2003, pp. 521-524.

[39] J. Choi, J. Choi, and M. Kang, "Dimensioning Burst Assembly Process in Optical Burst Switching Networks", in *EICE Trans. Commun.*, vol. E88-B, No. 10, pp. 3855-3863, October 2005.

[40] A. Tanenbaum, *Computer Networks*, New York, USA: Prentice Hall, 1999.

[41] G. Guillemin and P. Boyer, "ATM block transfer capabilities: The special case of ABT/DT", in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '96)*, London, United Kingdom, 1996, pp. 762-766.

[42] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optican WAN's", in *IEEE Transactions on Communications*, vol. 40, No. 7, pp. 1171-1182, July 1992.

[43] PhysOrg.Com (2005), "First time 1,000 channel WDM transmission demonstration in an installed optical fiber". URL: http://www.physorg.com/news3316.html, last access in 13 July 2007.

[44] C. Kan, H. Balt, S. Michel, and D. Verchère, "Network-element view information model for an optical burst core switch", in *Proceedings of The International Society for Optical Engineering - SPIE Asia-Pacific Optical Wireless Communications (APOC 2001)*, vol. 4584, Beijing, China, 12-16 November 2001, pp. 115-125.

[45] C. Kan, H. Balt, S. Michel, and D. Verchère, "Information Model of an Optical Burst Edge Switch", in *Proceedings of IEEE International Conference on Communications (ICC 2002)*, vol. 5, New York, USA, 28 April - 2 May 2002 pp. 2717-2721.

[46] Alcatel (2005), "Alcatel Products by Sub.Category: Optical Cross Connects", Alcatel. URL: http://www.alcatel.com/products/productsbysubfamily.jhtml?subCategory=Optical%20Cross%20Connects&category=Optical%20Transmission&pageNumber=1, last access in 25 September 2007.

[47] Ciena (2005), "Optical Cross-Connect: The Replacement for Separate ADM, MSPP and Digital Cross-Connect Systems", Ciena. URL: http://www.ciena.com/products/productsapps_3909.htm, last access in 25 October 2007.

[48] Matisse Networks (2006), "Matisse Networks Optical Burst Switching". URL: http://www.matissenetworks.com/, last access in 13 December 2007.

[49] L. Xu, "Performance Analysis of Optical Burst Switched Networks", PhD thesis, North Carolina State University, Raleigh, North Carolina, USA, 2002.

[50] C. Qiao and J. Staley, "Labeled Analog Burst Switching For Any High-Layer-Over-Signal Channel Integration", United States Patent Office, USA, patent number US 2003/0067919 A1, 10 April 2003.

[51] MEMX California (2005), "MEMS Optical Cross Connect". URL: http://www.memx.com/cross%20connect.htm, last access in 25 September 2007.

[52] Texas Instruments Inc. (2005), "Block Diagram - Optical Networking - Optical Cross Connect (OXC) and Optical Add/Drop Multiplexer (OADM)", Texas Instruments Inc. URL: http://focus.ti.com/docs/apps/catalog/resources/blockdiagram.jhtml?bdId=851, last access in 25 September 2007.

[53] J. Kim, J. Cho, M. Jain, D. Gutierrez, C. Su, R. Rabbat, and T. Hamada, "Demonstration of 2.5 Gbps Optical Burst Switched WDM Rings Network", in *Proceedings of Optical Fiber Communication Conference and Exposition (OFC 2006) and the National Fiber Optic Engineers Conference (NFOEC 2006)*, Anaheim, California, USA.

[54] Y. Sun, T. Hashiguchi, V. Q. Min, X. Wang, H. Morikawa, and T. Aoyama, "Design and Implementation of an Optical Burst-Switched Network Testbed", in *IEEE Optical Communications*, vol. 43, No. 11, pp. S48-S55, November 2005.

[55] J. Teng, "A Study of Optical Burst Switched networks with the Jumpstart Just in Time Signaling Protocol", PhD thesis, North Carolina State University, Raleigh, NC, USA, 2004.

[56] I. Baldine, H. Perros, G. Rouskas, and D. Stevenson, "JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks", in *Proceedings of Networking 2002*, vol. LNCS 2345, Springer-Verlag, pp. 1081-1086.

[57] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic, "SLOB: a Switch with Large Optical Buffers for Packet Switching", in *IEEE Journal of Lightwave Technology*, vol. 16, pp. 1725.

[58] V. Vokkarane, G. P. V. Thodime, V. U. B. Challagulla, and J. P. Jue, "Channel Scheduling Algorithms using Burst Segmentation and FDLs for Optical Burst-Switched Networks", in *Proceedings of IEEE International Conference on Communications (ICC 2003)*, Anchorage, Alaska, USA, 11-15 May 2003.

[59] V. Vokkarane and J. P. Jue, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks", in *Proceedings of Optical Fiber Communication Conference and Exhibit (OFC 2002)*, Anaheim, California, USA, 17-22 March 2002.

[60] S. Sheeshia and C. Qiao, "Burst Grooming in Optical Burst-Switched Networks", in *Proceedings of IEEE/SPIE First Workshop on Traffic Grooming in WDM Networks*, IEEE/SPIE (Eds.), San Jose, California, USA, 29 October 2004, pp. 153-166.

[61] K. Zhu and B. Mukherjee, "Traffic grooming in WDM optical mesh networks", in *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 122-133, January 2002.

[62] M. Düser and P. Bayvel, "Analysis of a Dynamically Wavelength-Routed Optical Burst Switched Architecture", in *Journal of Lightwave Technology*, vol. 20, No. 4, pp. 574-585, April 2002.

[63] A. Zapata and P. Bayvel, "Dynamic Wavelength-Routed Optical Burst-Switched Networks: Scalability Analysis and Comparison with Static Wavelength-Routed Optical Networks", in *Proceedings of Optical Fiber Communication Conference (OFC 2003)*, vol. 1, Atlanta, Georgia, USA, pp. 212-213.

[64] H. Kong and C. Philips, "Prebooking Reservation Mechanism for Next-Generation Optical Networks", *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, No. 4, pp. 645-652, July/August 2006.

[65] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration", in *IEEE Communications Magazine*, vol. 38, No. 9, pp. 104-114, September 2000.

[66] E. Mannie ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, Internet Engineering Task Force (IETF), 2004.

[67] K. Long, Z. Yi, X. Yang, and H. Liu, "Generalized MPLS (GMPLS) architectures extensions for optical Burst Switch network ", IETF Draft, Internet Engineering Task Force (IETF), 2005.

[68] V. M. Vokkarane, K. Haridoss, and J. P. Jue, "Threshold-Based Burst Assembly Policies for QoS Support on Optical Burst-Switched Networks", in *Proceedings of SPIE Optical Networking and Communications Conference (OPTICOMM 2002)*, Boston, Massachussets, USA, 29 July - 1 August 2002, pp. 125-136.

[69] A. Ge and F. Callegati, "On Optical Burst Switching and Self-Similar Traffic", *IEEE Communications Letters*, vol. 4, No. 3, pp. 98-100, March 2000.

[70] X. Yu, Y. Chen, and C. Qiao, "Performance Evaluation of Optical Burst Switching with Assembled Burst Traffic Input", in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2002)*, vol. 3, 0-7803-7632-3, Taipei, 11-17 November 2002, pp. 2318-2322.

[71] M. C. Yuang, J. Shil, and P. L. Tien, "QoS Burstification for Optical Burst Switched WDM Networks", in *Proceedings of Optical Fiber Communication Conference (OFC 2002)*, Anaheim, California, USA, 17-22 March 2002.

[72] H. L. Vu and M. Zuckerman, "Blocking Probability for Priority Classes in Optical Burst Switching Networks", in *IEEE Communication Letters*, vol. 6, No. 5, pp. 214-216, May 2002.

[73] C. Qiao and J. Staley, "Method to Route and Re-Route Data in OBS/LOBS and other Burst Switched Networks", United States Patent Office, USA, patent number US 2003/0206521-A1, 6 November 2003.

[74] V. M. Vokkarane, Q. Zhang, J. P. Jue, and B. Chen, "Generalized Burst Assembly and Scheduling Techniques for QoS Support in Optical Burst Switched Networks", in *Proceedings of IEEE International Conference on Communications (ICC 2002)*, New York, USA, 28 April - 2 May 2002.

[75] K. Dolzer, "Assured Horizon – An efficient framework for service differentiation in optical burst switched networks", in *Proceedings of SPIE Optical Networking and Communications Conference (OptiComm 2002)*, Boston, Massachusetts, USA, 30-31 July 2002.

[76] T. Tachibana, T. Ajima, and S. Kasahara, "Round-Robin Burst Assemby with Constant Transmission Scheduling for Optical Burst Switching Networks", in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2003)*, vol. 5, IEEE, San Francisco, California, USA, 1 - 5 December 2003, pp. 2772-2776.

[77] K. Long, R. S. Tucker, and C. Wang, "A New Framework and Burst Assembly for IP DiffServ over Optical Burst Switching Networks", in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2003)*, vol. 5, IEEE, San Francisco, California, USA, 1 - 5 December 2003, pp. 3159-3164.

[78] M. d. V. Rodrigo and J. Götz, "An Analytical Study of Optical Burst Switching Aggregation Strategies", in *Proceedings of First Annual International Conference on Broadband Networks (BROADNETS 2004)*, San Jose, California, USA, 25-29 October 2004.

[79] D. Q. Liu and M. T. Liu, "Priority-based Burst Scheduling Scheme and Modeling in Optical Burst-Switched WDM Networks", in *Proceedings of International Conference on Telecommunications ICT*, Beijing, China, 23-26 June 2002.

[80] D. Q. Liu and M. T. Liu, "Burst Scheduling for Differentiated Services in Optical Burst Switching WDM Networks", *International Journal of Communication Systems*, vol. 17, No. 2, pp. 127-140, March 2004.

[81] I. Widjaja, "Performance Analysis of Burst Admission Control Protocols", in *IEE Proceding of Communications*, vol. 142, pp. 7-14, February 1995.

[82] R. Karanam, V. Vokkarane, and J. Jue, "Intermediate Node Initiated (INI) Signalling: A Hybrid Reservation Technique for Optical Burst-Switched Networks", in *Proceedings of Optical Fiber Communications Conference (OFC 2003)*, vol. 1, Atlanta, Georgia, USA, 23-28 March 2003, pp. 213-215.

[83] H. Li, K. H. Ling, I. L.-J. Thng, and M. W. L. Tan, "Dual control packets in optical burst switching networks", in *Journal of Optical Networking*, vol. 3, No. 11, pp. 787-801, November 2004.

[84] J. Teng and G. N. Rouskas, "A Detailed Analysis and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks", *Photonic Network Communications*, vol. 9, No. 3, pp. 311-335, May 2005.

[85] A. Detti and M. Listanti, "Application of Tell & Go and Tell & Wait Reservation Strategies in a Optical Burst Switching Network: a Performance Comparison", in *Proceedings of IEEE 8th International Conference on Telecommunication (ICT 2001)*, vol. 2, IEEE, Bucharest, Romania, 4-7 June 2001, pp. 540-548.

[86] S.-Y. Oh and M. Kang, "Control Packet Structure and method for Generating a Data Burst in Optical Burst Switching Networks", United States Patent Office, USA, patent number US 2003/0099243A1, 29 May 2003.

[87] I. Bryskin and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", RFC 4397, Internet Engineering Task Force (IETF), 2006.

[88] L. Berger et al., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, Internet Engineering Task Force (IETF), 2003.

[89] J. S. Turner, "Terabit burst switching", *Journal of High Speed Networks*, vol. 8, No. 1, pp. 3-16, January 1999.

[90] J. Y. Wei and R. I. McFarland, "Just-in-Time signaling for WDM optical burst switching networks", *Journal of Lightwave Technology*, vol. 18, No. 12, pp. 2019-2037, December 2000.

[91] I. Baldine, G. Rouskas, H. Perros, and D. Stevenson, "JumpStart - A Just-In-Time Signaling Architecture for WDM Burst-Switched Networks", *IEEE Communications Magazine*, vol. 40, No. 2, pp. 82-89, February 2002.

[92] J. J. C. P. Rodrigues, M. Freire, N. M. Garcia, and P. Monteiro, "Enhanced Just-in-Time: A New Resource Reservation Protocol for Optical Burst Switching Networks", in *Proceedings of 12th IEEE International Symposium on Computers and Communications (ISCC 2007)*, IEEE Computer Society Press, Los Alamitos, California, USA, ISBN: 1-4244-1520-9, Aveiro, Portugal, 1-4 July 2007, pp. 121-126.

[93] B. Zhou, M. A. Bassiouni, and G. Li, "Improving Fairness in optical-burst-switching networks", in *Journal of Optical Networking*, vol. 3, No. 4, pp. 214-228, April 2004.

[94] M. Yoo and C. Qiao, "QoS performance of optical burst switching in IP over WDM networks", *IEEE Journal of Selected Areas in Communications*, vol. 18, No. 10, pp. 2026-2071, October 2000.

[95] Q. Zhang, V. M. Vokkarane, B. Chen, and J. P. Jue, "Early Drop and Wavelength Grouping Schemes for Providing Absolute QoS Differentiation in Optical Burst-Switched Networks", in *Proceedings of IEEE Global Communication Conference (GLOBECOM 2003)*, vol. 5, San Francisco, California, USA, pp. 2628-2632.

[96] C. M. Gauger, H. Butcha, E. Patzak, and J. Saniter, "Performance meets technology - an integrated evaluation of OBS nodes with FDL buffers ", in *Proceedings of 1st International Workshop on Optical Burst Switching (WOBS)*, Dallas, Texas, USA.

[97] T. Battestilli and H. Perros, "Optical Burst Switching: A Survey", North Carolina State University, NCSU Computer Science Technical Report, TR-2002-10, July 2002.

[98] C. M. Gauger, K. Dolzer, and M. Scharf, "Reservation strategies for FDL buffers in OBS networks ", University of Stuttgart, Stuttgart, Germany, (Report), IND, University of Stuttgart, 2001, 2001.

[99] J. Li and C. Qiao, "Proactive Burst Contention Avoidance Scheduling Algorithms for Labeled Optical Burst Switching Networks", United States Patent Office, USA, patent number US 2004/0063461 A1, 1 April 2004.

[100] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[101] B. Golden, "Shortest-Path Algorithms: A Comparison", *Operations Research*, vol. 24, No. 6, pp. 1164-1169, November - December 1976.

[102] A. V. Goldberg and R. E. Tarjan, "Expected Performance of Dijkstra's Shortest Path Algorithm", NEC Research Institute, Princeton, New Jersey, USA, Technical Report num. 96-062, June 1996.

[103] Dawn Networking Research Labs (2002), "The network simulator ns-2". URL: http://www.isi.edu/nsnam/ns/ last access in 10 December 2007.

[104] N. M. Bhide and K. M. Sivalingam, "Design of OWns: Optical Wavelength Division Multiplexing (WDM) Network Simulator", in *Proceedings of First SPIE Optical Networking Workshop*, Dallas, Texas, USA, Jan 2000.

[105] J. J. P. C. Rodrigues, N. M. Garcia, M. M. Freire, and P. Lorenz, "Object-Oriented Modeling and Simulation of Optical Burst Switching Networks", in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2004) - Conference Workshops, 10th IEEE Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD 2004)*, Dallas, Texas, USA, November 29th - December 3rd, 2004, pp. 288-292.

[106] R. Shenai, S. Gowda, and K. Sivalingam (2005), "OIRC OBS-ns Simulator". URL: http://wine.icu.ac.kr/~obsns/index.php, last access in 19 November 2007.

[107] N. Nagatsu and Y. Hamazumi, "Optical path accommodation design applicable to large scale networks", in *IEICE Trans. Commun.*, vol. E78-B, No. 4, pp. 597-607, 1995.

[108] N. Nagatsu, S. Okamoto, and K.-I. Sato, "Optical Path Cross/Connect System Scale Evaluation Using Path Accomodation Design for Restricted Wavelength Multiplexing", in *IEEE Journal on Selected Areas in Communications*, vol. 14, No. 5, pp. 893-901, June 1996.

[109] J. Teng and G. Rouskas, "On wavelength assignment in optical burst switched networks", in *Proceedings of First International Conference on Broadband Networks BroadNets 2004*, San Jose, California, USA, 25-29 Oct. 2004, pp. 24-33.

[110] L. Li and A. K. Somani, "Dynamic wavelength routing techniques and their performance analyses", in *Optical WDM Networks - Principles and Practice*, K. M. Sivalingam and S. Subramaniam, Eds., The Nederlands: Kluver Academic Publishers, 2000, pp. 247-275.

[111] Y. Zhu, G. Rouskas, and H. Perros, "A comparison of allocation policies in Wavelength-Routed networks ", in *Photonic Network Communications* vol. 2, No. 3, pp. 265-293, August 2000.

[112] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient Channel Scheduling Algorithms in Optical Burst Switched Networks", in *Proceedings of IEEE Infocom 2003*, San Francisco, California, USA, 1-3 April 2003.

[113] K. H. Liu, *IP Over WDM*, West Sussex: John Wiley & Sons, 2002.

[114] R. Batchellor and O. Gerstel, "Cost Effective Architectures for Core Transport Networks", in *Proceedings of Optical Fiber Communication Conference and Exposition (OFC 2006) and the National Fiber Optic Engineers Conference (NFOEC 2006)*, Anaheim, California, USA.

[115] F. Farahmand, J. Jue, V. Vokkarane, J. J. P. C. Rodrigues, and M. M. Freire, "A Layered Architecture for Supporting Optical Burst Switching ", in *Telecommunications 2005*, Petre Dini, Pascal Lorenz, Mário Freire, Pierre Rolin, Pawel Szulakiewicz, and Alexandra Cristea (Eds.), A Layered Architecture for Supporting Optical Burst Switching IEEE Computer Society Press, Los Alamitos CA, July 2005, ISBN: 0-7695-2388-9, pp. 213-218 (*Proceedings of the Advanced Industrial Conference on Telecommunications / Service Assurance with Partial and Intermittent Resources Conference / E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE 2005)*, Lisbon, Portugal, 17-20 July 2005).

[116] International Organization for Standardization (ISO), "Open Systems Interconnection Basic Model", ISO 7498, 1984.

[117] J. Postel, "Transmission Control Protocol (TCP)", RFC 793, Internet Engineering Task Force (IETF), 1981.

[118] N. M. Garcia, M. Freire, and P. Monteiro, "Measuring and Profiling IP Traffic", in *Proceedings of 4th European Conference on Universal Multiservice Networks (ECUMN 2007)*, IEEE Computer Society Press, Los Alamitos, California, USA, ISBN: 0-7695-2768-X, Toulouse, France, 14-16 February 2007, pp. 283-291.

[119] V. Paxson (2005), "A Comparative Analysis of TCP Tahoe, Reno, New-Reno, SACK and Vegas". URL: http://inst.eecs.berkeley.edu/~ee122/fa05/projects/Project2/SACKRENEVEGAS.pdf, last access in 25 September 2007.

[120] S. Gowda, R. K. Shenai, K. M. Sivalingam, and H. C. Cankaya, "Performance Evaluation of TCP over Optical Burst-Switched (OBS) WDM Networks", in *Proceedings of IEEE International Conference on Communications (ICC 2003)*, vol. 2, Anchorage, Alaska, USA, 11-15 May 2003, pp. 1433-1437.

[121] X. Yu, C. Qiao, and Y. Liu, "TCP Implementations and False Time Out Detection in OBS Networks", in *Proceedings of 23rd Conference of the IEEE Communications Society (IEEE INFOCOM 2004)*, Hong-Kong, China, 7-11 March 2004.

[122] Cisco Systems Inc. (2007), "Cisco Press - 1587200546 - CCNP Practical Studies: Routing", Cisco Press. URL: http://safari.ciscopress.com/1587200546/ch02, last access in 30 December 2007.

[123] D. J. Blumenthal, P. R. Prucnal, and J. R. Sauer, "Photonic Packet Switches: Architectures and Experimental Implementations", in *Proceedings of the IEEE*, vol. 82, No. 11, pp. 1650-1667, November 1994.

[124] A. Concepcion (1991), "Algorithma 2006 - Dijkstra's Shortest Path", California State University. URL: http://algodev.csci.csusb.edu:8080/Prototype17/pages/dijkstraGenInfo.html, last access in 25 October 2007.

[125] J. Moy, "OSPF version 2", RFC 2328, Internet Engineering Task Force (IETF), 1998.

[126] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6", RFC 2740, Internet Engineering Task Force (IETF), 1999.

[127] IEEE Project 802 Working Group 802.1, "IEEE Spanning-Tree Protocol", 802.1D, IEEE, 1998.

[128] A. Mokhtar and M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks", in *IEEE/ACM Transactions on Networking*, vol. 6, No. 2, pp. 197-206, April 1998.

[129] "The GÉANT network". URL: http://www.geant.net, last access in 1 August 2007.

[130] M. J. O'Mahony, "Results from the COST 239 project. Ultra-high capacity optical transmission networks", in *Proceedings of 22nd European Conference on Optical Communications (ECOC'96)*, vol. 2, Oslo, Norway, 15-19 September 1996, pp. 11-18.

[131] D. Thaler and C. Hops, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, Internet Engineering Task Force (IETF), 2000.

[132] I. d. Miguel, M. Düser, and P. Bayvel, "Traffic Load Bounds for Optical Burst-Switched Networks with Dynamic Wavelength Allocation", in *Proceedings of 5th Working-Conference on Optical Network Design and Modelling (ONDM 2001)*, Vienna, Austria.

[133] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Assessment of Optical Burst Switching Ring and Chordal Ring Networks", in *Kluwer Telecommunications Systems*, vol. 27, 2004, pp. 133-149.

[134] J. C. S. Castro, J. Pedro, and P. P. Monteiro, "Burst Loss Reduction in OBS Networks by Minimizing Network Congestion", in *Proceedings of Conftele 2005*, Tomar, Portugal, April 2005.

[135] J. Teng and G. Rouskas, "Traffic engineering approach to path selection in optical burst switching networks", *Journal of Optical Networking*, vol. 4, No. 11, pp. 759-777, November 2005.

[136] P. Baran, "On Distributed Communications Networks", in *IEEE Transactions on Communications Systems*, vol. 12, No. 1, pp. 1-9, March 1964.

[137] X. Wang, H. Morikawa, and T. Aoyama, "Burst optical deflection routing protocol for wavelength routing WDM networks", in *Proceedings of SPIE/IEEE Opticom 2000*, Dallas, Texas, USA, October 2000.

[138] C.-F. Hsu, T.-L. Liu, and N.-F. Huang, "Performance Analysis of Deflection Routing in Optical Burst Switched Networks", in *Proceedings of IEEE Infocom 2002*, New York, New York, USA.

[139] M. Ueda, T. Tachibana, and S. Kasahara, "Last-hop preemptive scheme based on the number of hops for optical burst switching networks", *Journal of Optical Networking*, vol. 4, No. 10, pp. 640-660, October 2005.

[140] N. M. Garcia, M. Pereira, M. M. Freire, P. P. Monteiro, and P. Lenkiewicz, "Performance of Optical Burst Switched Networks for Grid Applications", in *Proceedings of Third International Conference on Networking and Services (ICNS 2007), International Workshop on GRID over Optical Burst Switching Networks (GOBS 2007)*, IEEE Computer Society Press, Los Alamitos, California, USA, ISBN: 0-7695-2858-9, Athens, Greece, 19-25 July 2007, pp. 120-126.

[141] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks", in *Communications of the ACM*, vol. 19, No. 5, pp. 395-404, July 1976, available at http://www.acm.org/classics/apr96.

[142] K.-Y. Siu and R. Jain, "A Brief Overview of ATM: Protocol Layers, LAN Emulation, and Traffic Management", in *ACM Computer Communications Review*, vol. 25, No. 2, pp. 6.

[143] J. Reynolds, "Assigned numbers", RFC 3232, Internet Engineering Task Force (IETF), 2002.

[144] X. Mountrouidou and H. Perros, "Characterization of Burst Aggregation Process in Optical Burst Switching", in *Proceedings of Networking 2006*, F. Boavida et al. (Eds.), vol. 1, Springer, Coimbra, Portugal, 15-19 May 2006, pp. 752-764.

[145] National Laboratory for Applied Network Research (2005), "NLANR Sites". URL: http://pma.nlanr.net/Sites/, last access in 13 September 2007.

[146] National Laboratory for Applied Network Research (2005), "NLANR PMA: Special Traces Archive". URL: http://pma.nlanr.net/Special/, last access in 13 September 2007.

[147] National Laboratory for Applied Network Research (2005), ".tsh file format (Time Stamped Header)", NLANR. URL: http://pma.nlanr.net/Traces/tsh.format.html, last access in 13 September 2005.

[148] National Laboratory for Applied Network Research (2005), "IPv4 hashing function source code (.tsh file format)", NLANR. URL: ftp://pma.nlanr.net/pub/dagtools-0.9.6.tar.gz, last access in 13 September 2007.

[149] Hewlett Packard Inc., "IEther-00 (iether) Gigabit Ethernet Driver for HP-UX 11i v 2 of September 2004 Release Notes", Hewlett Packard Inc., Product Specification.

[150] *Determining Your Service Provider's MTU Settings*, GE Security, 2004.

[151] T. J. Hacker and B. D. Athey, "The End-to-End Performance Effects of Parallel TCP Sockets on a Lossy Wide Area Network", in *Proceedings of International Parallel and Distributed Processing Symposium*, Fort Lauderlade, Florida, USA, April 15-19, 2002.

[152] *Configuring a GRE Tunnel over IPSec with OSPF*, Cisco Systems Inc., 2005.

[153] Cisco Systems Inc., "Configuring PIX to PIX Dynamic−to−Static IPSec with NAT and Cisco VPN Client", Cisco Systems Inc., Technical Manual, Nov 08, 2002.

[154] J. McCann, S. Deering, and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, Internet Engineering Task Force (IETF), 1996.

[155] M. Mathis "Raising the Internet MTU". URL: http://www.psc.edu/~mathis/MTU/index.html, last access in 13 July 2007.

[156] Apparent Networks, "Maximum Transmission Unit: Hidden Restrictions on High Bandwith Networks", Apparent Networks, White Paper, December 2001.

[157] X. Mountrouidou, "Bimodal Scheduling for OBS Networks and Characterization of Burst Aggregation Algorithms", PhD thesis, North Carolina State University, Raleigh, North Carolina, USA, 2006.

[158] S. R. Amstutz, "Burst Switching - An Update", in *IEEE Communications Magazine*, vol. 27, No. 9, pp. 50-57, September 1989.

[159] G. Almes, S. Kalidindi, and M. Zekauskas, "One Way Delay Metrics for IP Performance Metrics (IPPM)", RFC 2679, Internet Engineering Task Force (IETF), 1999.

[160] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, Internet Engineering Task Force (IETF), 2002.

[161] IP Performance Metrics IETF workgroup IETF. URL: http://www.ietf.org/html.charters/ippm-charter.html, last access in 15 June 2005.

[162] S. Malik and U. Killat, "Impact of Burst Aggregation Time on Performance in Optical Burst Switching Networks", in *Proceedings of Optical Fibre Communications Conference (OFC 2004)*, vol. 2, IEEE, Los Angeles, California, USA, February 23-27, 2004, pp. 2.

[163] X. Yu, Y. Chen, and C. Qiao, "A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks", in *Proceedings of Optical Networking and Communications*, vol. 4874, SPIE, Boston, Massachusetts, USA, July 2002, pp. 149-159.

[164] W. T. Willinger and R. M. S. Sherman, "Self-similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the source level", *IEEE / ACM Transactions on Networking*, No. 5, pp. 71-86.

[165] K. Park, "How does TCP generate Pseudo-self-similarity?" in *Proceedings of 1997 Winter Simulation Conference*, Atlanta, Georgia, USA, 7-10 December 1997, pp. 215-223.

[166] M. S. Borella, S. Uludag, G. B. Brewster, and I. Sidhu, "Self-similarity of Internet Packet Delay", in *Proceedings of IEEE International Conference on Communications (ICC'97)*, Montreal, Quebec, Canada, 08-12 June 1997, pp. 513-517.

[167] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (extended version)", *IEEE Transactions on Networking*, vol. 2, pp. 1-15, 1994.02.

[168] D. G. Waddington and F. Chang, "Realizing the Transition to IPv6", *IEEE Communications Magazine*, vol. 40, No. 6, pp. 138-148, June 2002.

[169] H. Ning, "IPv6 Test-bed Networks and R&D in China", in *Proceedings of International Symposium on Applications and the Internet Workshops (SAINTW'04)*, Tokyo, Japan, 26-30 January 2004.

[170] J. Bound, "IPv6 Deployment Next Steps & Focus (Infrastructure!!!)", presented to IPv6 US Summit, December 8-11, 2003, Arlington, Virginia, USA.

[171] Cisco Systems Inc. (2002), "IPv6 Deployment Strategies", Cisco Systems Inc. URL: http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/ipv6_sol/ipv6dswp.pdf, last access in 15-09-2007.

[172] T.-Y. Wu, H.-C. Chao, T.-G. Tsuei, and E. Lee, "A Measurement Study of Network Efficiency for TWAREN IPv6 Backbone", *International Journal of Network Management*, vol. 15, No. 6, pp. 411-419, November 2005.

[173] J.-M. Uzé, "Enabling IPv6 Services in ISP Networks", presented to International Workshop on IPv6 Testing Certification and Market Acceptance, 22 September 2003, Brussels, Belgium.

[174] J. Palet, "Addressing the Digital Divide with IPv6-enabled Broadband Power Line Communications", ISOC - Internet Society, Report, 5 May 2003.

[175] J. Palet, "IPv6 Overall Status", IPv6 Task Force, IPv6 Steering Commitee, Report, 14 February 2003.

[176] P. Schlütter (2000), "Aggregation of IP Flows", Siemens AG. URL: http://mr-ip.icm.siemens.de/mr/traffic/TR/flow-agg.pdf, last access in 15-01-2006.

[177] A. Conta, S. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", IETF Draft on RFC 2463, Internet Engineering Task Force, 2004.

[178] T. Ferrari, "End-To-End performance analysis with Traffic Aggregation", *Computer Networks*, vol. 34, No. 6, pp. 905-914, 2000.

[179] J. A. Cobb, "Preserving Quality of Service Guarantees in Spite of Flow Aggregation", *IEEE/ACM Transactions on Networking*, vol. 10, No. 1, pp. 43-53, February 2002.

[180] A. Detti and M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks", in *Proceedings of IEEE Infocom'02*, New York, USA, June 23-27, 2002.

[181] E. Dotaro and A. Jorgdan, "Optical Backbone Topology Design under Traffic Aggregation and Flexibility Constraints", in *Proceedings of Optical Society of America Optical Fiber Communication Conference and Exhibit (OFC 2001)*, vol. 2, Anaheim, California, USA, 17-22 March 2001, pp. TuG6-1- TuG6-3.

[182] M. Düser and P. Bayvel, "Burst Aggregation Control and Scalability of Wavelength Routed Optical Burst-Switched (WR-OBS) Networks", in *Proceedings of European Conference on Optical Communications (ECOC'02)*, Copenhagen, Denmark, September 9-12, 2002.

[183] A. Sridharan, S. Bhattacharyya, C. Dyot, R. Guérin, J. Jetcheva, and N. Taft, "On The Impact of Aggregation on The Performance of Traffic Aware Routing", in *Proceedings of International Teletraffic Congress*, Salvador da Bahia, Brazil, December 2001.

[184] A. J. Van Der Schaft, M. Dalsmo, and B. M. Maschke, "Mathematical structures in the network representation of energy-conserving physical systems", in *Memorandum- University of Twente Faculty of Applied Mathematics*, vol. 1333, pp. all.

[185] S. Mittal (2004), "Implementation of K-shortest Path Dijkstra Algorithm used in All-optical Data Communication Networks", University of Arizona, Tucson, Arizona, USA, Project Report Spring 2004. URL: http://www.u.arizona.edu/~saurabh/SIE/kDijkstra546.pdf, last access in 20-07-2007.

[186] D. Eppstein, "Finding the *k* shortest paths", *Annual Symposium on Foundations of Computer Science*, vol. 35, pp. 154, 1994.

[187] D. Q. Liu and M. T. Liu, "Priority-based Burst Scheduling Scheme and Modeling in Optical Burst-Switched WDM Networks", in *Proceedings of ICT*, 2002.

[188] Yoo, C. Qiao, and S. Dixit, "Optical Burst Switching for Service Differentiation in the Next generation Optical Internet", *IEEE Communications Magazine*, February 2001.

[189] F. Farahmand and J. Jue, "Look-ahead Window Contention Resolution in Optical Burst Switched Networks", in *Proceedings of IEEE Workshop on High Performance Switching and Routing HPSR*, Torino, Italy, 24-27 June 2003, pp. 147-151.

[190] K. Gokyu, K.-i. Baba, and M. Murata, "On Path Accommodation Methods for Optical Compression TDM Ring", in *Proceedings of Wokshop on Optical Networks* Dallas, Texas, USA, January 2000.

[191] P. Bartford and L. Landweber, "Bench-style Network Research in an Internet Instance Laboratory", *ACM SIGCOMM Computer Communications Review*, vol. 33, No. 3, pp. 21, Julho 2003.

[192] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, New York: Plenum Press, 1992.

[193] H. Perros, *Computer Simulation Techniques: The definitive introduction!*: Computer Science Department, North Carolina State University, Raleigh, North Carolina, USA, 2003, [Online]. Available: URL: http://www.csc.ncsu.edu/faculty/perros/simulation.pdf, last access 25 October 2007.

[194] J. Teng and G. N. Rouskas, "A Comparison of the JIT, JET, and Horizon Wavelength Reservation Schemes on a Single OBS Node", in *Proceedings of International Workshop on Optical Burst Switching (WOBS 2003)*, Dallas, Texas, USA, 16 October 2003.

[195] C. Alexopoulos and A. F. Seila, "Implementing the Batch Means Method in Simulation Experiments", in *Proceedings of IEEE Winter Simulation Conference*, Coronado, CA, USA, 8-11 December 1996, pp. 214-221.

[196] Sun Microsystems Inc. (2003), "Class Math". URL: http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Math.html/ last access in 16-07-2004.

[197] Sun Microsystems Inc. (2003), "Class Random". URL: http://java.sun.com/j2se/1.4.2/docs/api/java/util/Random.html, last access in 16-07-2004.

[198] T. S. Schriber and R. W. Andrews, "Interactive Analysis of Simulation Output by the Method of Batch Means", in *Proceedings of IEEE 11th Winter Simulation Conference*, IEEE Press, San Diego, California, USA, 3-5 December 1979, pp. 513-526.

[199] M. Listanti, V. Eramo, and R. Sabella, "Architectural and Technological Issues for Future Optical Internet Networks", in *IEEE Communications Magazine*, vol. 39, No. 9, pp. 82-93, September 2000.